Knowledge Tracing Adaptive Testing

Fabrice Popineau

(With help from Yolaine Bourda, Jill-Jênn Vie and Benoît Choffin)
CentraleSupelec – LRI

Who am I?

Fabrice POPINEAU

Web site: https://fabrice.popineau.net/

Professor @ CentraleSupelec

Affiliated to LRI (UMR 8623) in the LAHDAK team

Briefly:

- Worked in the industry for a short time (CapGemini Innovation, 2 years)
- Assistant professor and professor @ Supélec, Metz Campus for 20 years (almost)
  - In charge of the 3rd year option about Artificial Intelligence (6 years)
- Moved to Supélec, Gif/Yvette campus in 2010, CS department
- Ecole Centrale Paris and Supélec merger in 2015/01
- National expert for France/AFNOR at ISO JTC1/SC42 "Artificial Intelligence"

Why am I here?

Because I have been kindly invited by Anne Boyer and Armelle Brun ☺

Working on AI use in education for about 20 years

Co-supervised several thesis about the use of AI in education with Yolaine Bourda:

- Cédric Jacquiot (2006) (with Chantal Reynaud co-supervision)
  - *Modélisation logique et générique des systèmes d'hypermédias adaptatifs.*
- Georges Dubus (2013)
  - *Transformation de programmes logiques : application à la personnalisation et à la personnification d'agents.*
- Hiba Hajri (2018)
  - *Linked Education for Personalization.*
- Jill-Jênn Vie (2016) (with Eric Bruillard co-supervision)
  - *Construction et analyse de tests adaptatifs dans un cadre de crowdsourcing – Applications aux MOOC.*
- Benoît Choffin (due 2020)
  - *Personalized learning plans on an online training platform for sustainable learning.*

Knowledge Tracing

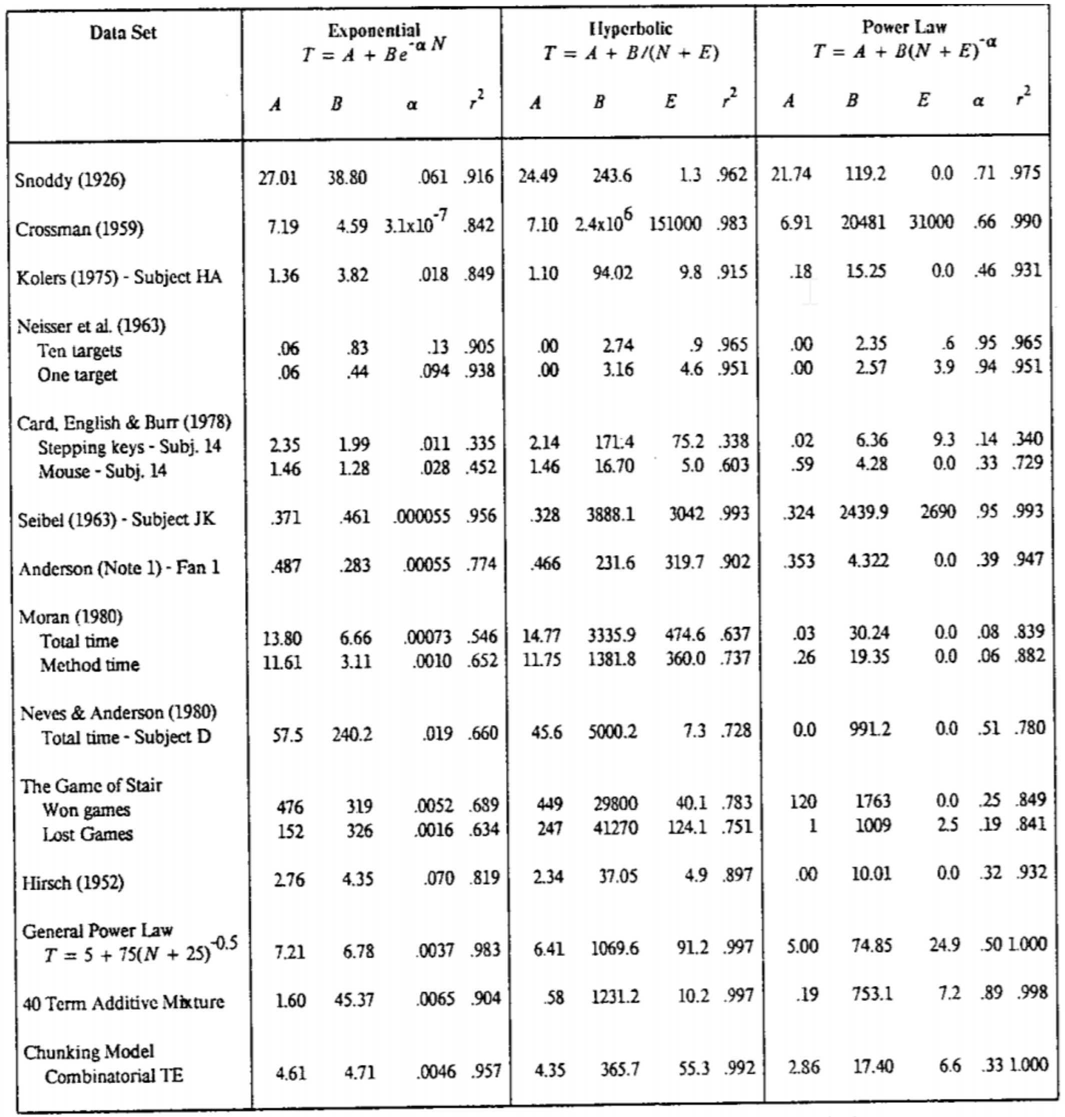
Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge
Corbett, A.T. & Anderson, J.R. User Model User-Adap Inter (1994) 4: 253.
https://doi.org/10.1007/BF01099821

Some recent survey:

Radek Pelánek, Bayesian Knowledge Tracing, Logistic Models, and Beyond: An Overview of Learner Modeling Techniques

Power law

- Newell, A., & Rosenbloom, P. S. (1981). *Mechanisms of skill acquisition and the law of practice.* Cognitive skills and their acquisition, 1, 1-55.
- Fit empirical curves optimally to 3 classes of functions
- Heathcote, Brown, and Mewhort (2000) *The power law repealed: the case for an exponential law of practice.* give alternative explanation:
  - Each student's practice is better fit by an exponential curve
  - Aggregation of them fit a power law curve
- (Debate possibly not closed)



Data Set	Exponential $T = A + Be^{-\alpha N}$				Hyperbolic $T = A + B/(N + E)$				Power Law $T = A + B(N + E)^{-\alpha}$				
	A	B	α	r^2	A	B	E	r^2	A	B	E	α	r^2
Snoddy (1926)	27.01	38.80	.061	.916	24.49	243.6	1.3	.962	21.74	119.2	0.0	.71	.975
Crossman (1959)	7.19	4.59	$3.1\text{x}10^{-7}$	.842	7.10	$2.4\text{x}10^{6}$	151000	.983	6.91	20481	31000	.66	.990
Kolers (1975) - Subject HA	1.36	3.82	.018	.849	1.10	94.02	9.8	.915	.18	15.25	0.0	.46	.931
Neisser et al. (1963)													
Ten targets	.06	.83	.13	.905	.00	2.74	.9	.965	.00	2.35	.6	.95	.965
One target	.06	.44	.094	.938	.00	3.16	4.6	.951	.00	2.57	3.9	.94	.951
Card, English & Burr (1978)													
Stepping keys - Subj. 14	2.35	1.99	.011	.335	2.14	171.4	75.2	.338	.02	6.36	9.3	.14	.340
Mouse - Subj. 14	1.46	1.28	.028	.452	1.46	16.70	5.0	.603	.59	4.28	0.0	.33	.729
Seibel (1963) - Subject JK	.371	.461	.000055	.956	.328	3888.1	3042	.993	.324	2439.9	2690	.95	.993
Anderson (Note 1) - Fan 1	.487	.283	.00055	.774	.466	231.6	319.7	.902	.353	4.322	0.0	.39	.947
Moran (1980)													
Total time	13.80	6.66	.00073	.546	14.77	3335.9	474.6	.637	.03	30.24	0.0	.08	.839
Method time	11.61	3.11	.0010	.652	11.75	1381.8	360.0	.737	.26	19.35	0.0	.06	.882
Neves & Anderson (1980) Total time - Subject D	57.5	240.2	.019	.660	45.6	5000.2	7.3	.728	0.0	991.2	0.0	.51	.780
The Game of Stair													
Won games	476	319	.0052	.689	449	29800	40.1	.783	120	1763	0.0	.25	.849
Lost Games	152	326	.0016	.634	247	41270	124.1	.751	1	1009	2.5	.19	.841
Hirsch (1952)	2.76	4.35	.070	.819	2.34	37.05	4.9	.897	.00	10.01	0.0	.32	.932
General Power Law $T = 5 + 75(N + 25)^{-0.5}$	7.21	6.78	.0037	.983	6.41	1069.6	91.2	.997	5.00	74.85	24.9	.50	1.000
40 Term Additive Mixture	1.60	45.37	.0065	.904	.58	1231.2	10.2	.997	.19	753.1	7.2	.89	.998
Chunking Model Combinatorial TE	4.61	4.71	.0046	.957	4.35	365.7	55.3	.992	2.86	17.40	6.6	.33	1.000

Table 2: The General Learning Curves: Parameters from Optimal Fits in the *Log* Transformation Spaces

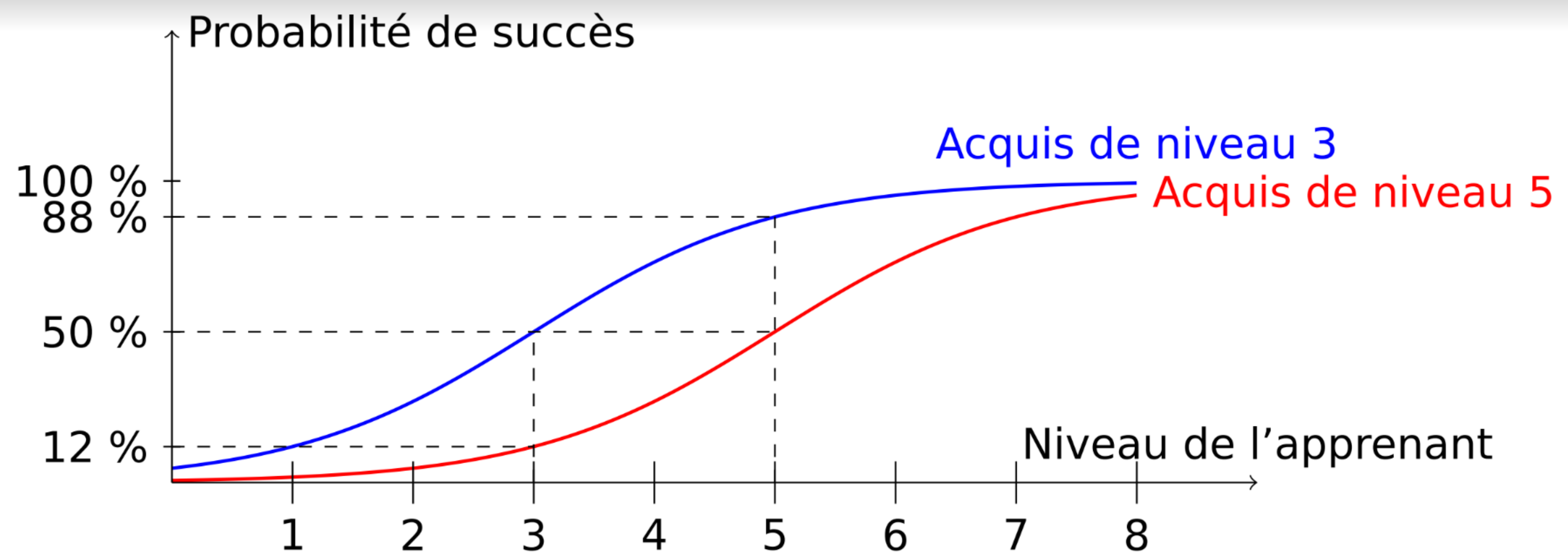
Item Response Theory

- A first simple, yet reliable model: the **Rasch model**
  - $R_{ij} \in \{0,1\}$ outcome of user *i* over item *j* (right/wrong)
  - Θ_i ability of user *i*
  - d_j difficulty of item *j*
  - Sigmoid $\sigma : x \rightarrow 1/(1 + \exp(-x))$

$$\Pr(R_{ij} = 1) = \sigma(\theta_i - d_j).$$

- Training
  - Learn Θ_i and d_j for historical data (maximizing log-likelihood)

Item Response Theory



IRT 1PL and 2PL, 3PL

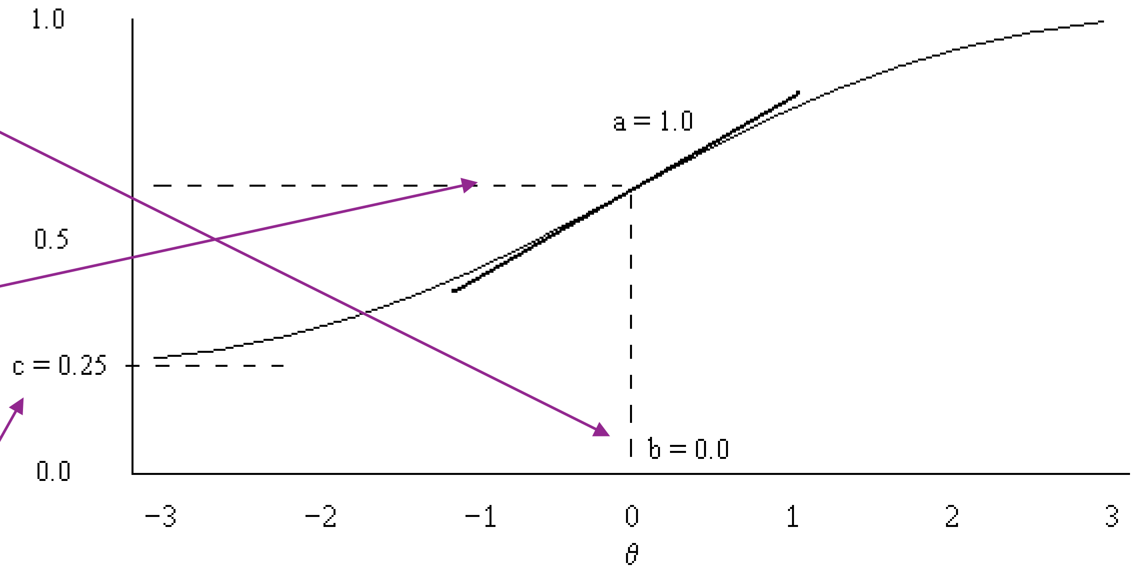
$$\Pr(R_{ij} = 1) = \sigma(\theta_i - d_j).$$

Difficulty

$$\Pr(R_{ij} = 1) = \sigma(a_j(\theta_i - d_j)).$$

Discrimination

a_j = 1 gives 1PL



$$\Pr\left(R_{ij} = 1\right) = c_j + (1 - c_j)\sigma(a_j(\theta_i - d_j)).$$

Pseudo-guessing

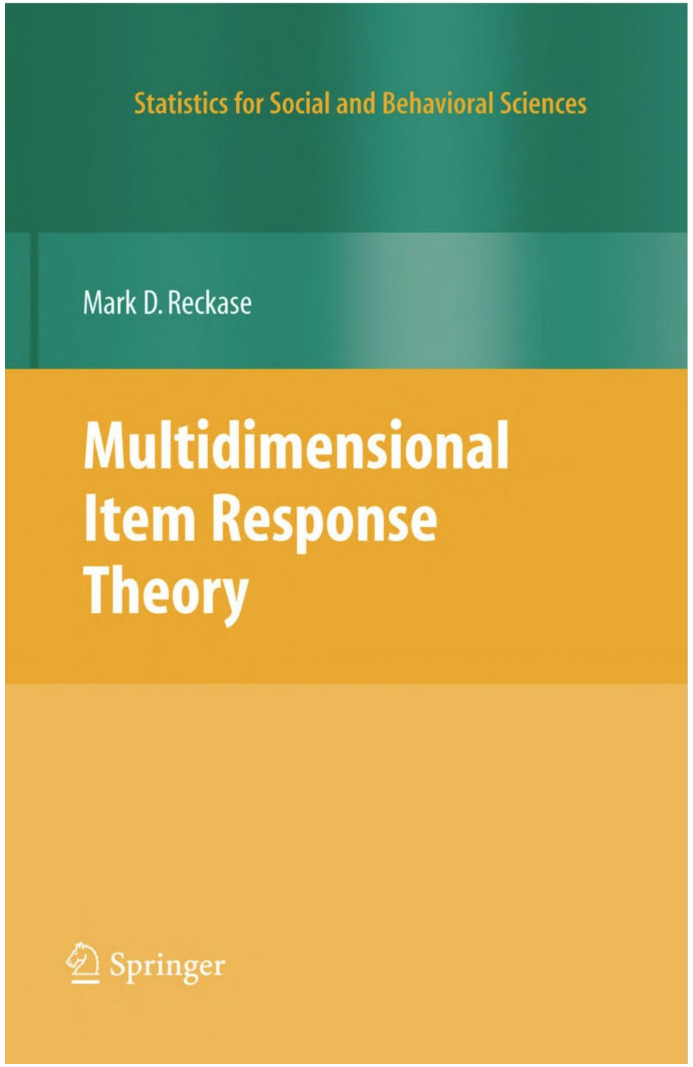
c_j = 0 gives 2PL

MIRT

- Extension of IRT to multiple skills

$$Pr(D_{ij} = 1) = \Phi(\boldsymbol{\theta_i} \cdot \boldsymbol{d_j} + \delta_j)$$

- $\boldsymbol{\theta_i}$ et $\boldsymbol{d_j}$ are vectors of dimension d
- δ_j is an easiness parameter (yep !)
- Collapses to unidimensional Rasch model when d=1
- Hard to calibrate

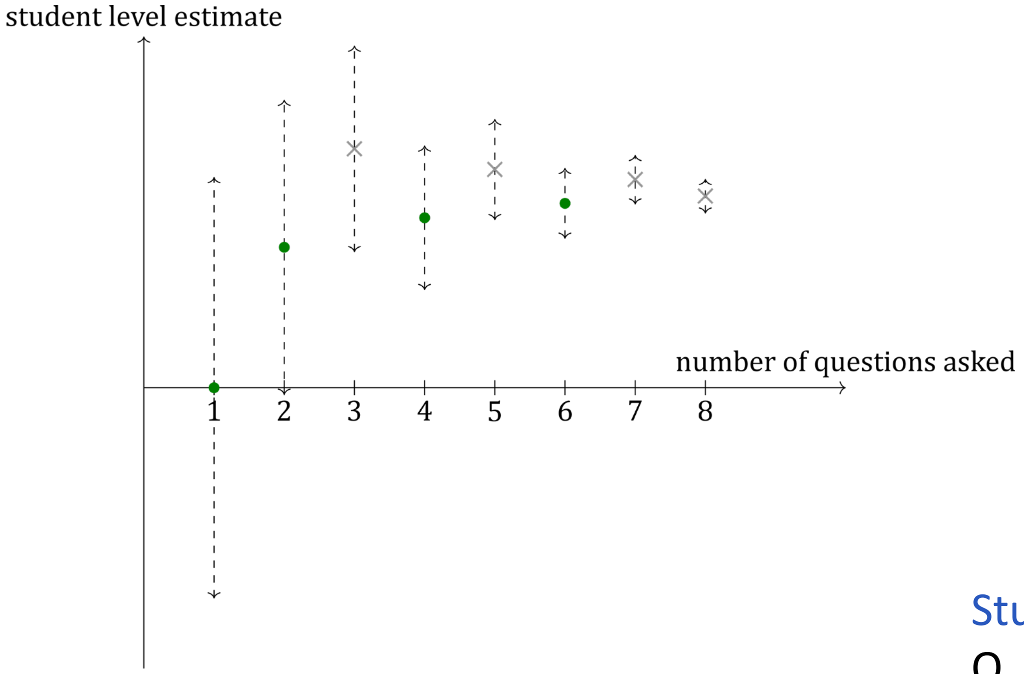

Statistics for Social and Behavioral Sciences
Mark D. Reckase
Multidimensional Item Response Theory
Springer

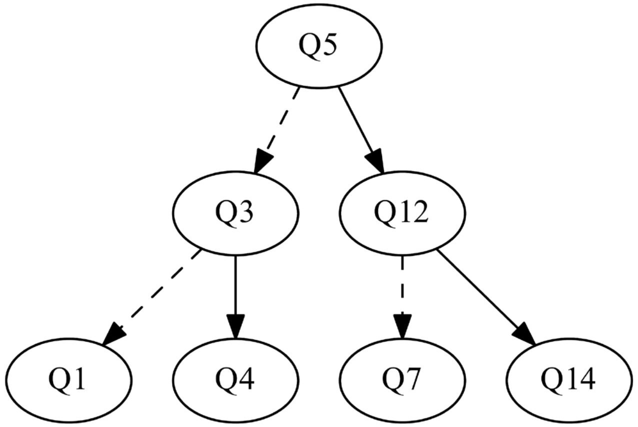
Adaptive Testing

Adaptive testing

- Choose the next question to ask to the student knowing its past answers since the beginning of the test: shorten the length of the assessment
  - Linden, Wim J. van der and Cees A. W. Glas (2010). *Elements of adaptive testing.* Springer (cit. on p. 26).
- Reasons: efficiency, avoid boredom
- Summative or formative assessment
- 2 criteria:
  - Choice of the next question
  - Termination
- Difficulties:
  - Slip: due to inattention, the student can fail on a question on which he should have succeeded
  - Guess: the student can answer correctly by luck
- We need robust methods, e.g. statistical
- These tests are heavily used: more than 238000 GMAT tests over the 2012-2013 period

Adaptive testing





Hypothesis: the student level does not change between questions

Student 1

Q. 5 asked…

Correct !

Q. 12 asked…

Incorrect !

Q. 7 asked

Incorrect.

Student 1 level is 6.

Student 2

Q. 5 asked

Incorrect.

Q. 3 asked

Correct !

Q. 4 asked

Incorrect.

Student 2 level is 3.

Item Response Theory

- A first simple, yet reliable model: the **Rasch model**
  - $R_{ij} \in \{0,1\}$ outcome of user *i* over item *j* (right/wrong)
  - Θ_i ability of user *i*
  - d_j difficulty of item *j*
  - Sigmoid $\sigma : x \rightarrow 1/(1 + \exp(-x))$

$$\Pr(R_{ij} = 1) = \sigma(\theta_i - d_j).$$

- Training
  - Learn Θ_i and d_j for historical data (maximizing log-likelihood)
  - For a new examinee *i*: keep d_j, learn Θ_i
  - Initialize $\hat{\theta}(0) = 0$
  - For each time t = 0,...,T − 1:
    - Ask question of difficulty d_j closest to student ability $\hat{\theta}(t)$ (proba closest to 1/2)
    - Refine student ability $\hat{\theta}(t+1)$ (maximum likelihood estimate)

Adaptive testing

- Process
- With IRT 1PL

$$\Pr\left(R_{ij} = 1\right) = p_{ij} = \sigma\left(\theta_i - d_j\right)$$

- How to choose the next question?
  - Observation: past answers
  - Want to estimate $\hat{\theta}(t+1)$
  - Fisher information

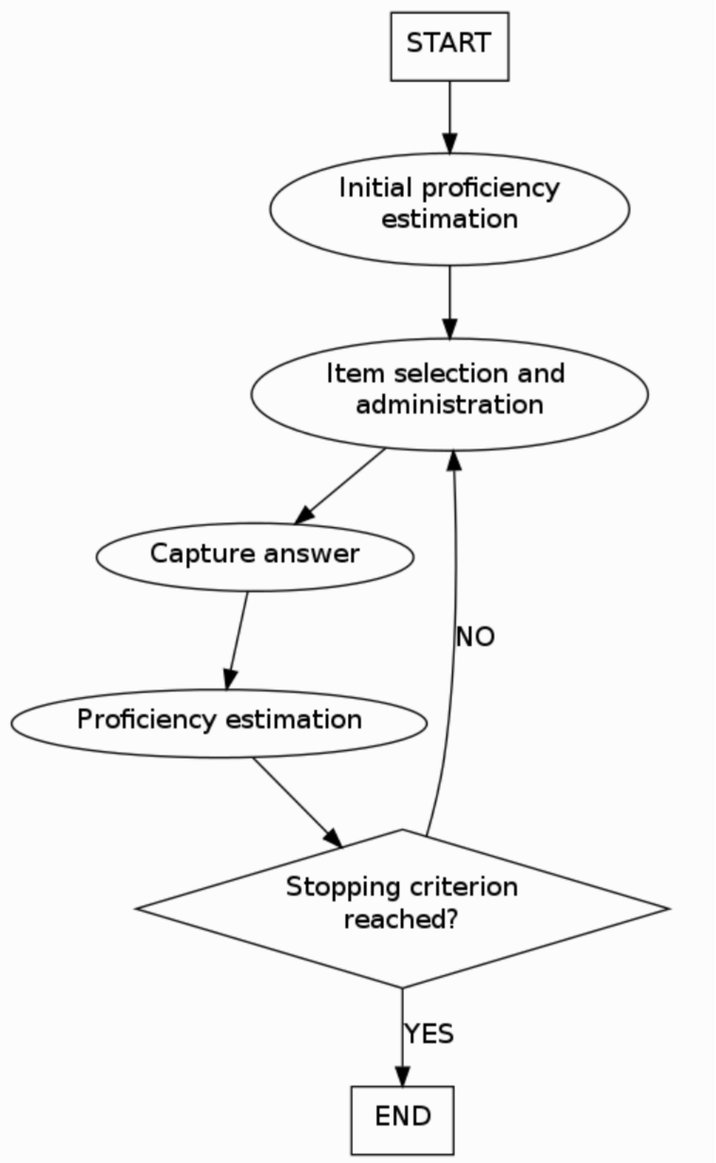
$$I_j(\theta) = E_{X_j}\left[\left(\frac{\partial}{\partial\theta}\log f\left(X_j,\theta,d_j\right)\right)^2 \mid \theta\right]$$

Probability function

User level

Item difficulty

Random variable about student success/failure: 1 if success, 0 if failure



START
Initial proficiency estimation
Item selection and administration
Capture answer
Proficiency estimation
NO
Stopping criterion reached?
YES
END

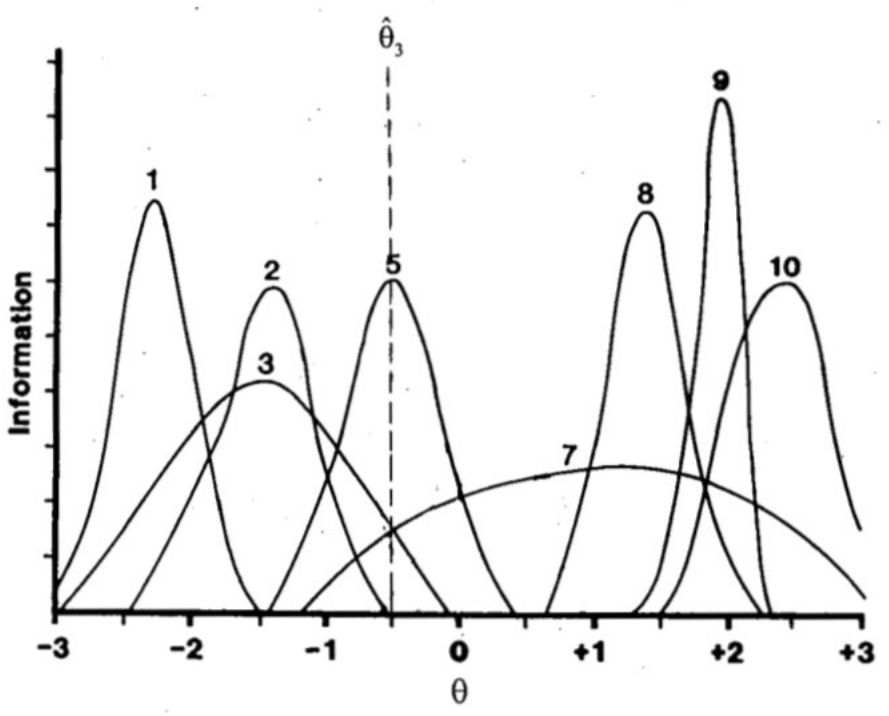
Adaptive Testing

- Fisher information for IRT 1PL

$$I_j(\theta) = p_{ij}(1 - p_{ij})$$

- Fisher information for IRT 2PL

$$I_j(\theta) = a_j^2 p_{ij}(1 - p_{ij})$$



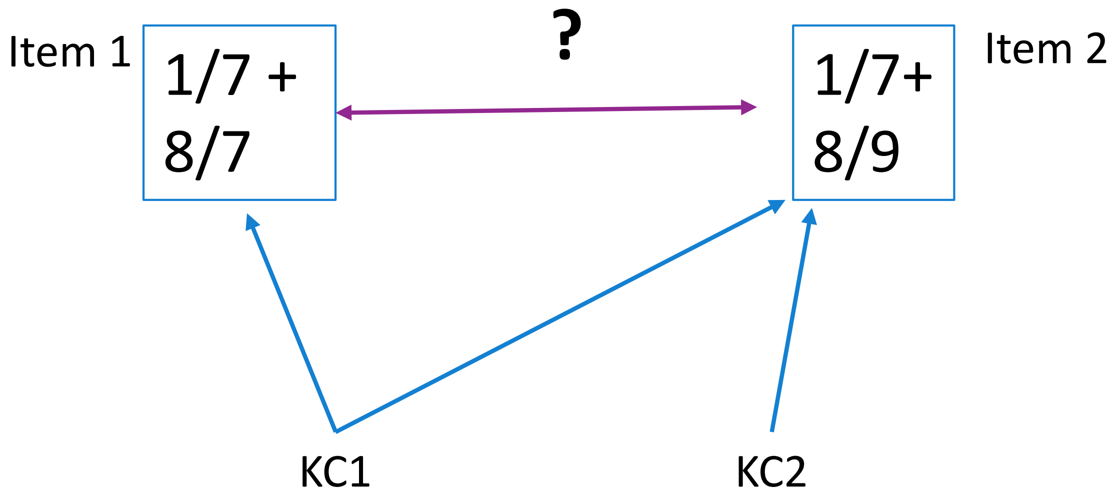
(a) -

(c) After administration of two items

[Computerized Adaptive Testing for Effective and Efficient Measurement in Counseling and Education](#) (Weiss, 2004)

Cognitive Diagnosis

- IRT does not help in diagnosing the difficulties experienced by the student
- Model hypothesis: the success on questions/tasks relies on the mastery of knowledge components (KC)



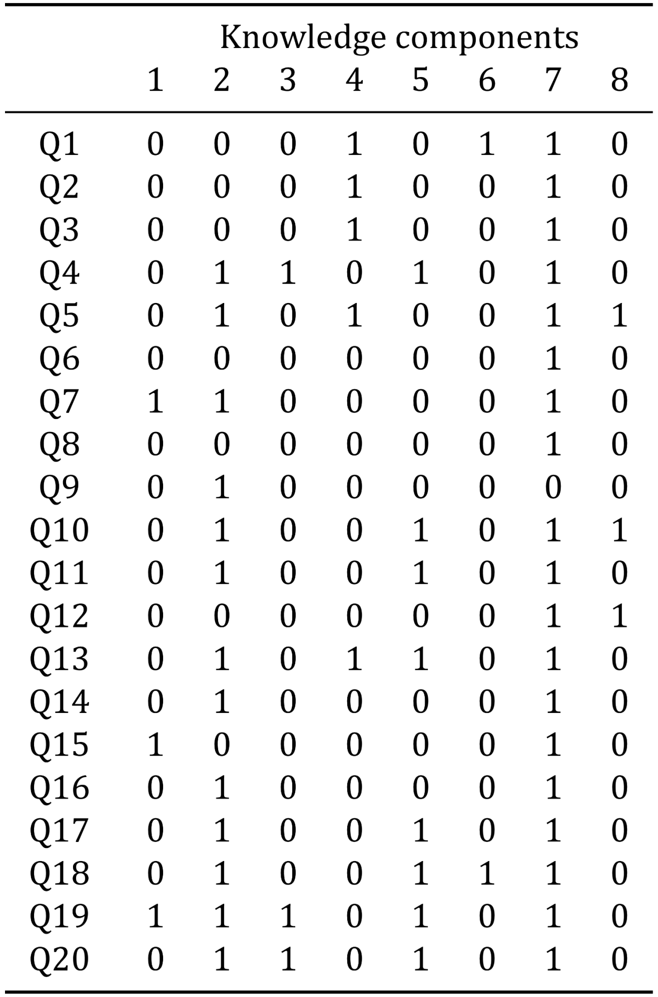
Q-Matrix

- (Tatsuoka, 1984)
- Establish a skills vs items connexion:

	Skills			
Items	Add	Sub	Mul	Div
a/b	0	0	0	1
a*b+c	1	0	1	0
a*b-c	0	1	1	0
(a+b)*c	1	0	1	0

- How to setup a Q-matrix?
  - Design it by hand (ouch!)
  - Learn it -> interpretability problem, difficult

Knowledge Components



	Knowledge components							
	1	2	3	4	5	6	7	8
Q1	0	0	0	1	0	1	1	0
Q2	0	0	0	1	0	0	1	0
Q3	0	0	0	1	0	0	1	0
Q4	0	1	1	0	1	0	1	0
Q5	0	1	0	1	0	0	1	1
Q6	0	0	0	0	0	0	1	0
Q7	1	1	0	0	0	0	1	0
Q8	0	0	0	0	0	0	1	0
Q9	0	1	0	0	0	0	0	0
Q10	0	1	0	0	1	0	1	1
Q11	0	1	0	0	1	0	1	0
Q12	0	0	0	0	0	0	1	1
Q13	0	1	0	1	1	0	1	0
Q14	0	1	0	0	0	0	1	0
Q15	1	0	0	0	0	0	1	0
Q16	0	1	0	0	0	0	1	0
Q17	0	1	0	0	1	0	1	0
Q18	0	1	0	0	1	1	1	0
Q19	1	1	1	0	1	0	1	0
Q20	0	1	1	0	1	0	1	0

Description of the 8 knowledge components :

1. convert an integer into a fraction
2. separate an integer from a fraction
3. simplify befor subtract
4. put to the same denominator
5. substract a fraction from an integer
6. handle the carry when subtracting numerators
7. subtracting numerators
8. reduce fractions to irreducible form

TABLE 1 – Q-matrix example for a 20 questions test about subtracting fractions.

DINA model

- DINA = *Deterministic-Input, Noisy-And-Gate*
- When Cognitive Diagnosis Meets Computerized Adaptive Testing: CD-CAT (Cheng, 2009)
- Latent state of the students wrt to KC: $c = (c_1, \dots, c_K)$
- Probability distribution over the latent states $\pi_i(c)$
- c_i = 1 means the student masters the KC
- Items are augmented with *slip* and *guess* parameters
- Initially: $\pi_{i,0}(c_k) = \frac{1}{|C|} = \frac{1}{2^K}$
- At step *t*, update the distribution: $\pi_{i,t+1}(c) = k_i(c)\pi_{i,t}(c)/Z$

Normalization

- With: $k_i(c) = \begin{cases} 1 - s_i & \text{If } c \text{ allows to answer and } r_i = 1 \text{ (correct answer)} \\ s_i & \text{If } c \text{ allows to answer and } r_i = 0 \text{ (wrong answer)} \\ g_i & \text{If } c \text{ does not allow to answer and } r_i = 1 \text{ (correct answer)} \\ 1 - g_i & \text{If } c \text{ does not allow to answer and } r_i = 0 \text{ (wrong answer)} \end{cases}$

DINA Model

- Choice of the next question based on entropy:

$$H(X) = -\sum_{c\in C} \pi(c)\log_2(\pi(c)).$$

- $H(\pi_t)$: uncertainty about the latent state
- $H(\pi_{t+1}|v)$: uncertainty after update, with the answer *v* information
- Ask the question *i* with the lowest value of:

$$Pr(r_i = 1)H(\pi_{t+1}|r_i = 1) + Pr(r_i = 0)H(\pi_{t+1}|r_i = 0)$$

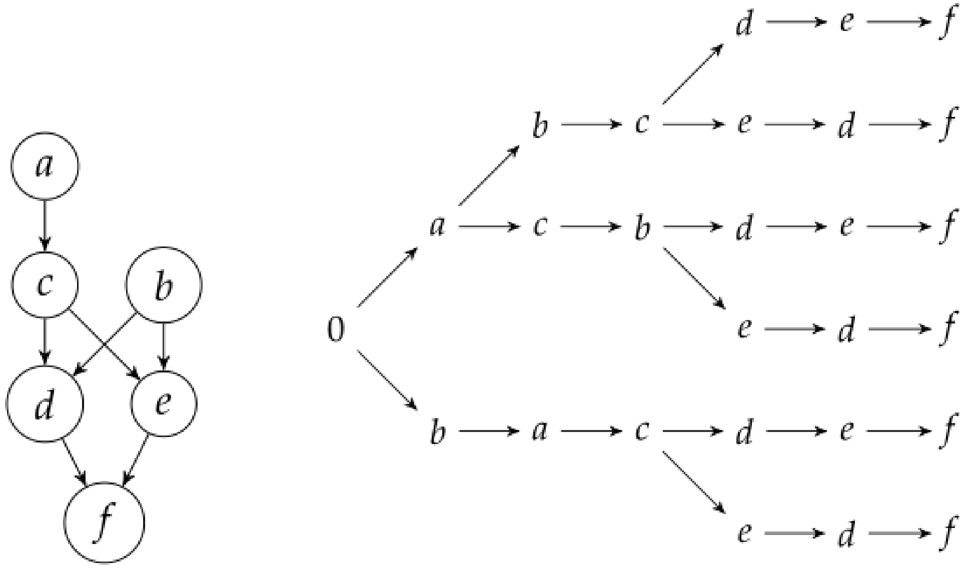
- This quantity is the mean entropy after student answer:

$$\Pr(r_i = 1) = (1 - s_i)\cdot \sum_{c|c \rhd i} \pi_t(c) + g_i \cdot \sum_{c|c \ntriangleright i} \pi_t(c)$$

- Here, $c \rhd i$ denotes the fact that KC in latent state c are sufficient to answer question *i* (and the converse for $c \ntriangleright i$)

DINA Model

- Successfully used for adaptive testing
- Big problem: 2^K latent states, untractable for high values of K
- Possible circumvention:
  - hierarchical representation of KC
  - [The assessment of knowledge, in theory and in practice](#) (Falmagne et al., 2006)
  - ALEKS belongs to McGraw-Hill Education (big business!)
  - The dependencies relations between KC help diminishing the size of the latent state space



Adaptive Testing and Collaborative Filtering

- Similarity
  - Seek to learn who is the user
  - Seek to provide the « right » item to the user
    - CAT : seek to estimate the ability of the user
    - CF: seek to satisfy the user (buy, buy, buy)
- Cold-start
  - User cold-start: how to handle a user we don’t know anything about?
  - Item cold-start: how to handle a new question never answered?
- More on this topic later

Hands On

IRT and CAT

Colab Notebook 01-Introduction to IRT.ipynb

Knowledge Tracing

(some slides taken from Baker et al. 2012

Learnlab (Pittsburgh Science of Learning Center) -- Educational Data Mining)

Bayesian Knowledge Tracing

- Goal: For each knowledge component (KC), infer the student’s knowledge state from performance.

- Suppose a student has six opportunities to apply a KC and makes the following sequence of correct (1) and incorrect (0) responses. Has the student has learned the rule?

0 0 1 0 1 1

Model Learning Assumptions

- Two-state learning model
  - Each skill is either learned or unlearned

- In problem-solving, the student can learn a skill at each opportunity to apply the skill

- A student does not forget a skill, once he or she knows it

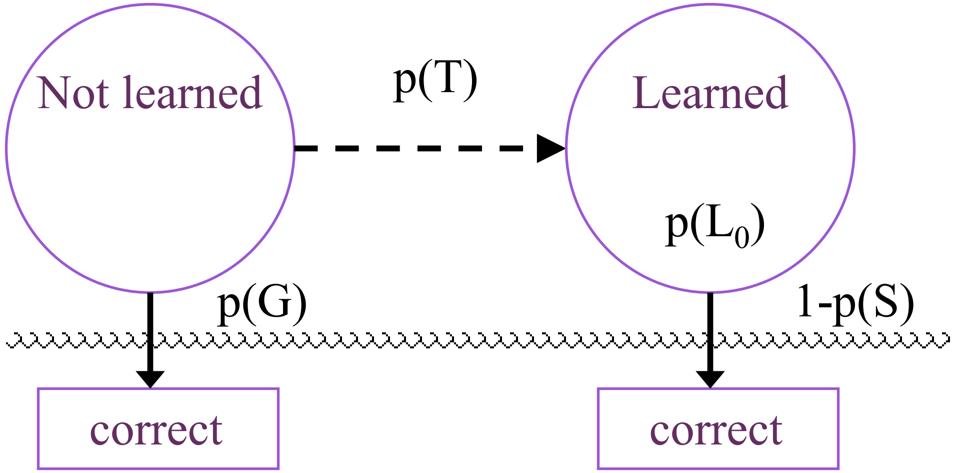
- Only one skill per action

Model Performance Assumptions

- If the student knows a skill, there is still some chance the student will slip and make a mistake.

- If the student does not know a skill, there is still some chance the student will guess correctly.

Corbett and Anderson's Model



Two Learning Parameters

p(L_0) Probability the skill is already known before the first opportunity to use the skill in problem solving.

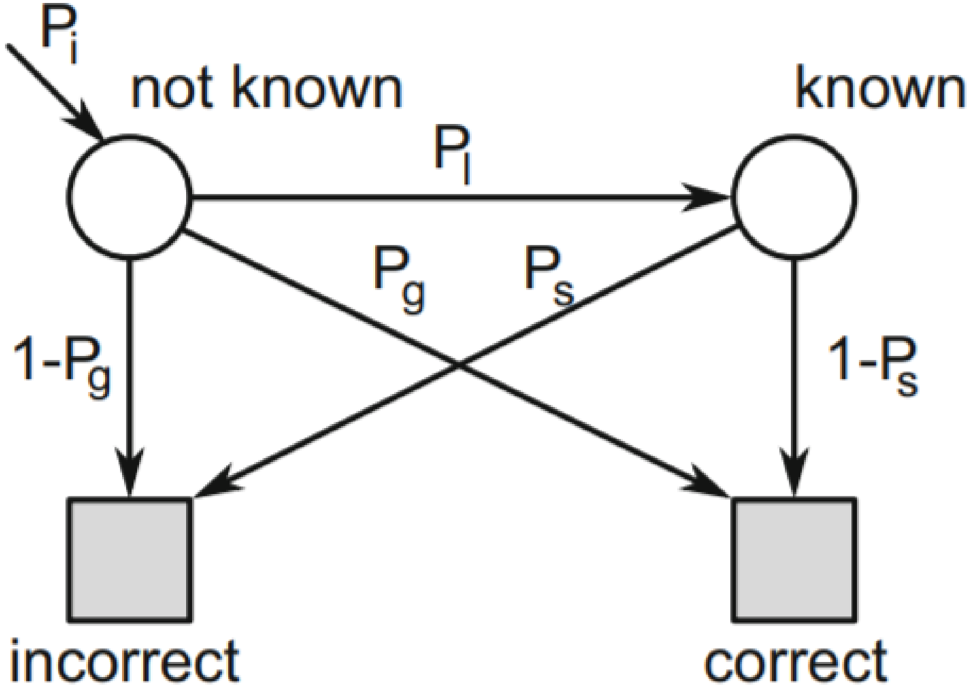
p(T) Probability the skill will be learned at each opportunity to use the skill.

Two Performance Parameters

p(G) Probability the student will guess correctly if the skill is not known.

p(S) Probability the student will slip (make a mistake) if the skill is known.

Formulas



update equation:

if $c = 1$: $\theta' := \frac{\theta(1-P_s)}{\theta(1-P_s)+(1-\theta)P_g}$

if $c = 0$: $\theta' := \frac{\theta P_s}{\theta P_s+(1-\theta)(1-P_g)}$

$\theta := \theta' + (1 - \theta')P_l$

prediction equation:

$P_{correct} = P_g \cdot \theta + (1 - P_s) \cdot (1 - \theta)$

Whenever the student has an opportunity to use a skill, the probability that the student knows the skill is updated using formulas derived from Bayes' Theorem

Knowledge Tracing

- How do we know if a knowledge tracing model is any good?
- Our primary goal is to predict ***knowledge***
- But knowledge is a latent trait
- But we can check those knowledge predictions by checking how well the model predicts ***performance***

Fitting a Knowledge-Tracing Model

- In principle, any set of four parameters can be used by knowledge-tracing

- But parameters that predict student performance better are preferred

Knowledge Tracing

- So, we pick the knowledge tracing parameters that best predict performance

- Defined as whether a student’s action will be correct or wrong at a given time

- Effectively a classifier

Recent Advances

- Recently, there has been work towards contextualizing the guess and slip parameters
  (Baker, Corbett, & Aleven, 2008a, 2008b)

- The intuition:
  Do we really think the chance that an incorrect response was a slip is equal when
  - Student has never gotten action right; spends 78 seconds thinking; answers; gets it wrong
  - Student has gotten action right 3 times in a row; spends 1.2 seconds thinking; answers; gets it wrong

Recent Advances

- In this work, P(G) and P(S) are determined by a model that looks at time, previous history, the type of action, etc.

- Significantly improves predictive power of method
  - Probability of distinguishing correct from incorrect increases by about 15% of potential gain
    - To 71%, so still room for improvement

Uses

- Outside of EDM, can be used to drive tutorial decisions

- Within educational data mining, there are several things you can do with these models

Uses of Knowledge Tracing

- If you want to understand a student's strategic/meta-cognitive choices, it is helpful to know whether the student knew the skill
  - Help-Seeking and Metacognition (Aleven et al, 2004, 2008)
- Gaming the system means something different if a student already knows the step, versus if the student doesn't know it
  - Gaming the System (Baker et al, 2004)
- A student who doesn't know a skill should ask for help; a student who does, shouldn't
  - Off-Task Behavior (Baker, 2007)
- KT can be interpreted to learn about skills

Often key components in models of other constructs

Skills from the Algebra Tutor

skill	L0	T
AddSubtractTypeinSkillIsolatepositiveIso	0.01	0.01
ApplyExponentExpandExponentsevalradicalE	0.333	0.497
CalculateEliminateParensTypeinSkillElimi	0.979	0.001
CalculatenegativecoefficientTypeinSkillM	0.953	0.001
Changingaxisbounds	0.01	0.01
Changingaxisintervals	0.01	0.01
ChooseGraphicala	0.001	0.306
combineliketermssp	0.943	0.001

Which skills could probably be removed from the tutor?

skill	L0	T
AddSubtractTypeinSkillIsolatepositiveIso	0.01	0.01
ApplyExponentExpandExponentsevalradicalE	0.333	0.497
CalculateEliminateParensTypeinSkillElimi	0.979	0.001
CalculatenegativecoefficientTypeinSkillM	0.953	0.001
Changingaxisbounds	0.01	0.01
Changingaxisintervals	0.01	0.01
ChooseGraphicala	0.001	0.306
combineliketermssp	0.943	0.001

Which skills could use better instruction?

skill	L0	T
AddSubtractTypeinSkillIsolatepositiveIso	0.01	0.01
ApplyExponentExpandExponentsevalradicalE	0.333	0.497
CalculateEliminateParensTypeinSkillElimi	0.979	0.001
CalculatenegativecoefficientTypeinSkillM	0.953	0.001
Changingaxisbounds	0.01	0.01
Changingaxisintervals	0.01	0.01
ChooseGraphicala	0.001	0.306
combineliketermssp	0.943	0.001

Bayesian Knowledge Tracing

- Introduced in 1995 (Corbett & Anderson)
- Four parameter simplification of ACT-R theory of skill acquisition (Anderson 1993)
- Computations based on a variation of Bayesian calculations proposed in 1972 (Atkinson)
- Formalized as equivalent to a Dynamic Bayesian Network (Rye, 2004) “Student modeling based on belief networks”

However:

- BKT cannot model the fact that a question might require several KCs

Additive Factors Model

How can we apply learning curves to model a student's learning in an intelligent tutoring system?

- There may be individual differences in students.

  (θ_i : Ability of student *i*)

- Students learn different skills at different rates.

  (β_k : learning rate of skill *k*)

- Different problems may share some of the same skills.

  (Q matrix: maps problems to skills)

Additive Factors Model

- $p_{ij,T}$: Probability that student *i* answers question *j* correctly at opportunity *T*.

$$p_{ij,T+1} = \sigma\left(\theta_i + \sum_{k \in KC(j)} (\beta_k + \gamma_k t_{ik})\right)$$

β_k: easiness of KC k

γ_k: learning rate of KC k

t_{ik}: number of opportunities for user i to practice KC k

- Review computation and application of the Additive Factor Model AFM (Durand et al. 2017)

Performance Factors Analysis

- Performance Factors Analysis – A New Alternative to Knowledge Tracing (Pavlik, Cen and Koedinger 2009)
- builds on AFM and uses past outcomes of practice instead of simple encounter counts:

$$\mathbb{P}(Y_{i,j}=1)=\sigma\left(\sum_{k\in KC(j)}\beta_k+\gamma_k c_{i,k}+\rho_k f_{i,k}\right)$$

number of correct answers of student i on KC k prior to this attempt

number of wrong answers of student i on KC k prior to this attempt

- θ_i has been removed from the model since it is not usually be estimated ahead of time in adaptive situations

Hands On

https://github.com/jilljenn/ktm/blob/master/doc/tuto.pdf

Knowledge Tracing:
A Step Further

Limitations of Students Models

We need to be able to infer skill memory strength and dynamics,

however in the student modeling literature:

- some models leverage item-skills relationships
- some others incorporate forgetting

But none does both!

Our Contribution

We take a model-based approach for this task.

1. Traditional adaptive spacing algorithms can be extended to review and practice skills (not only flashcards).
2. We developed a new student learning and forgetting model that leverages item-skill relationships: **DAS3H**.
   - DAS3H outperforms 4 SOTA student models on 3 datasets.
   - Incorporating skill info + forgetting effect improves over models that consider one or the other.
   - Using precise temporal information on past skill practice + assuming different learning/forgetting curves for different skills improves performance.

DASH

- Stands for item **D**ifficulty, student **A**bility, and **S**tudent **H**istory
  Improving Students’ Long-Term Knowledge Retention Through Personalized Review
  (Lindsey et al. 2014)
- Bridges the gap between *Factor Analysis models* and *memory models*

$$\mathbb{P}\left(Y_{s,j,t} = 1\right) = \sigma(\alpha_s - \delta_j + h_\theta(\mathrm{t}_{s,j,1:l}, \mathrm{y}_{s,j,1:l-1}))$$

- where:
  - $Y_{i,j,t}$ binary correctness of student i answering item j at time t
  - σ logistic function
  - α_i ability of student i
  - δ_j difficulty of item j
  - h_θ summarizes the effect of the l-1 previous attempts of i on j at times $\mathrm{t}_{i,j,1:l-1}$ + the binary outcomes $\mathrm{y}_{i,j,1:l-1}$

DASH

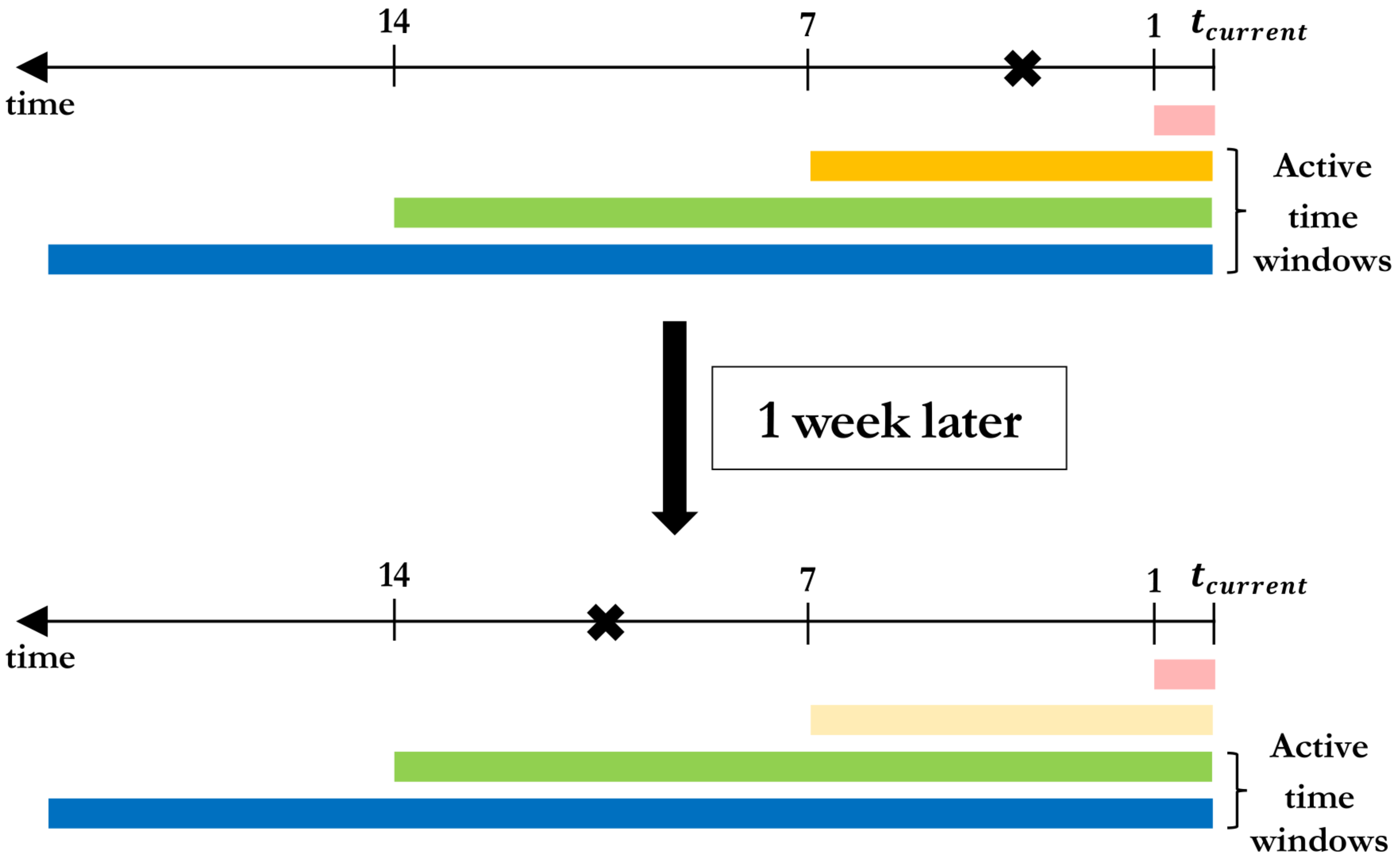
- (Lindsey et al. 2014) choose:

$$h_\theta(\mathrm{t}_{i,j,1:l}, \mathrm{y}_{i,j,1:l-1}) = \sum_{w=0}^{W-1} \theta_{2w+1} \log(1 + c_{i,j,w}) - \theta_{2w+2} \log(1 + a_{i,j,w})$$

- where:
  - w indexes a set of expanding time windows
  - $c_{i,j,w}$ number of correct answers of i on j in time window w
  - $a_{i,j,w}$ number of attempts of i on j in time window w
  - θ is *learned* by DASH.

DASH

- Assuming that the set of time windows is {1, 7, 14, +∞}:



DASH

- accounts for both learning and forgetting processes
- induces diminishing returns of practice inside a time window (log-counts)
- has a time module h_θ inspired by ACT-R (Anderson, Matessa,and Lebiere 1997) and MCM (Pashler, Cepeda, Lindsey, Vul, and Mozer 2009)
- outperforms a hierarchical Bayesian IRT on Lindsey et al. experimental data (vocabulary learning)
- was successfully used to adaptively personalize item review in a real-world cognitive psychology experiment.

However, DASH

- does not handle multiple skill item tagging → useful to account
- for knowledge transfer from one item to another
- assumes that memory decays at the same rate for every KC.

DAS3H

We extend DASH in 3 ways:

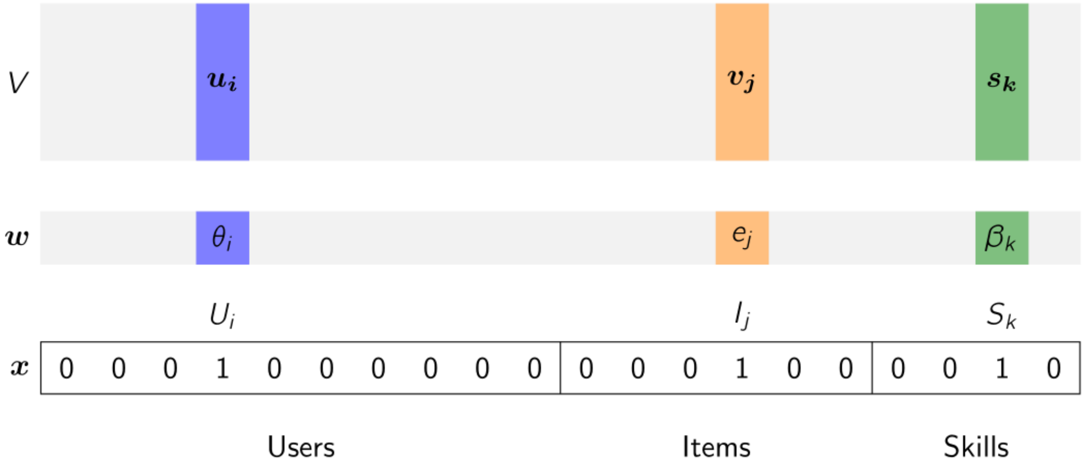
1. Extension to handle multiple skills tagging: new temporal module h_θ that also takes the multiple skills into account
   - Influence of the temporal distribution of past attempts and outcomes can differ from one skill to another
2. Estimation of easiness parameters for each item j and skill k
3. Use of KTMs (Vie and Kashima 2019) instead of mere logistic regression for multidimensional feature embeddings and pairwise interactions.

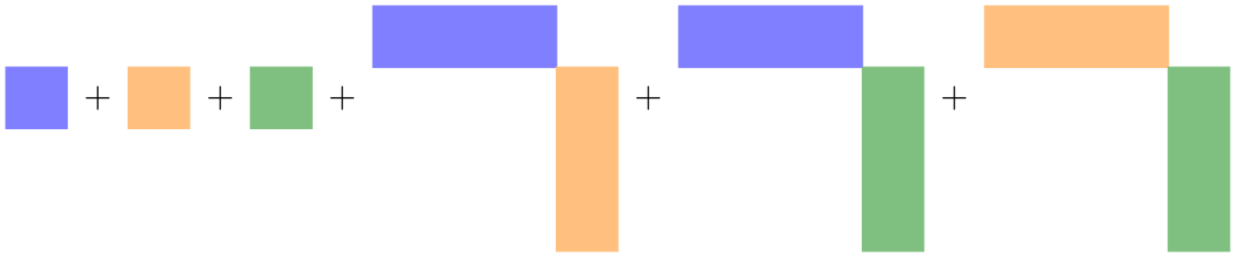
(Vie and Kashima 2019). Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing. In: Proceedings of the 33th AAAI Conference on Artificial Intelligence, to appear.

DAS3H

CentraleSupélec

Just pick features (e.g. user, item, skill) and you get a student model.
Each feature k is modeled by a bias w_k and an embedding v_k





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

DAS3H

→ DAS3H = item **D**ifficulty, student **A**bility, **S**kill and **S**tudent **S**kill practice **H**istory

For an embedding dimension of $d = 0$, DAS3H is:

$$\mathbb{P}(Y_{s,j,t} = 1) = \sigma(\alpha_s - \delta_j + \underbrace{\sum_{k \in KC(j)} \beta_k}_{\text{skill easiness biases}} + h_\theta(\mathrm{t}_{s,j,1:l}, \mathrm{y}_{s,j,1:l-1})).$$

We choose:

$$\begin{aligned} h_\theta(\mathrm{t}_{s,j,1:l}, \mathrm{y}_{s,j,1:l-1}) = \sum_{k \in KC(j)} \sum_{w=0}^{W-1} & \theta_{k,2w+1} \log(1 + c_{s,k,w}) \\ & - \theta_{k,2w+2} \log(1 + a_{s,k,w}). \end{aligned}$$

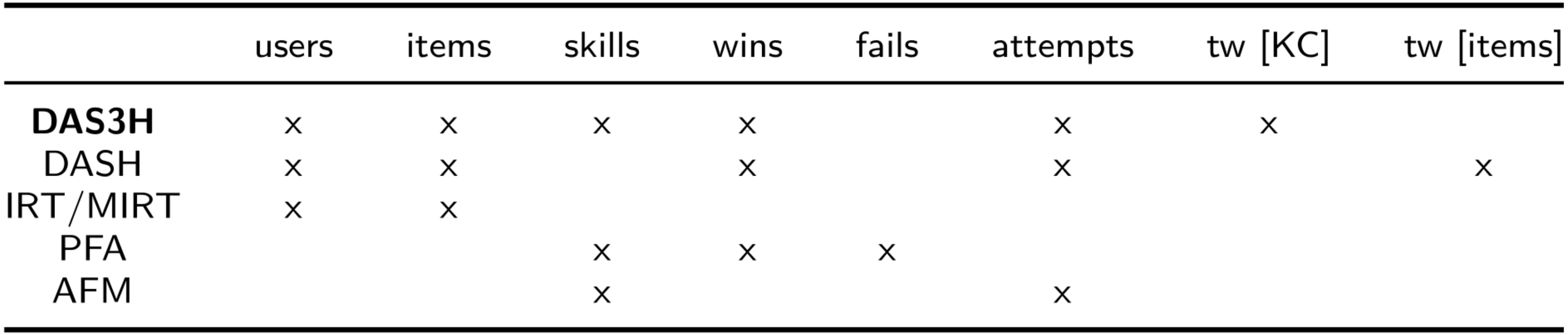
→ Now, h_θ can be seen as a sum of *skill* memory strengths!

Comparison With Other Models

5 contenders:

- DAS3H
- DASH (Lindsey, Shroyer, Pashler, and Mozer 2014)
- IRT/MIRT (Linden and Hambleton 2013)
- PFA (Pavlik, Cen, and K. R. Koedinger 2009)
- AFM (Cen, K. Koedinger, and Junker 2006)

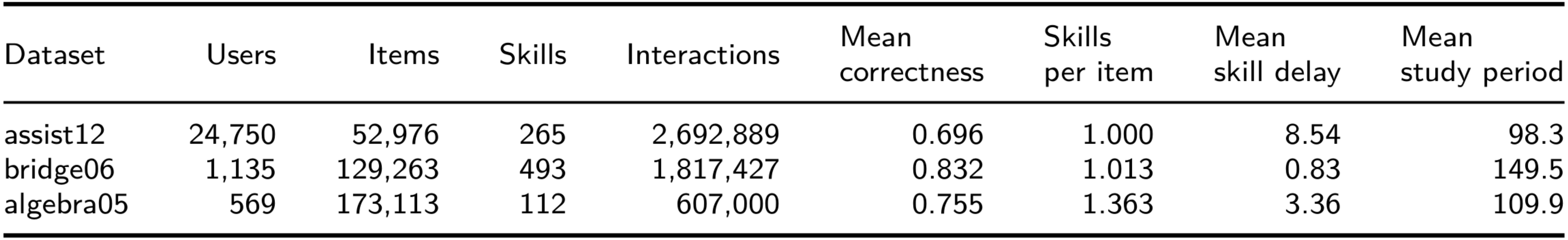
Every model was cast within the KTM framework → 3 embedding dimensions (0, 5 & 20) + sparse feature encoding.



	users	items	skills	wins	fails	attempts	tw [KC]	tw [items]
DAS3H	×	×	×	×		×	×	
DASH	×	×		×		×		×
IRT/MIRT	×	×						
PFA			×	×	×			
AFM			×			×		

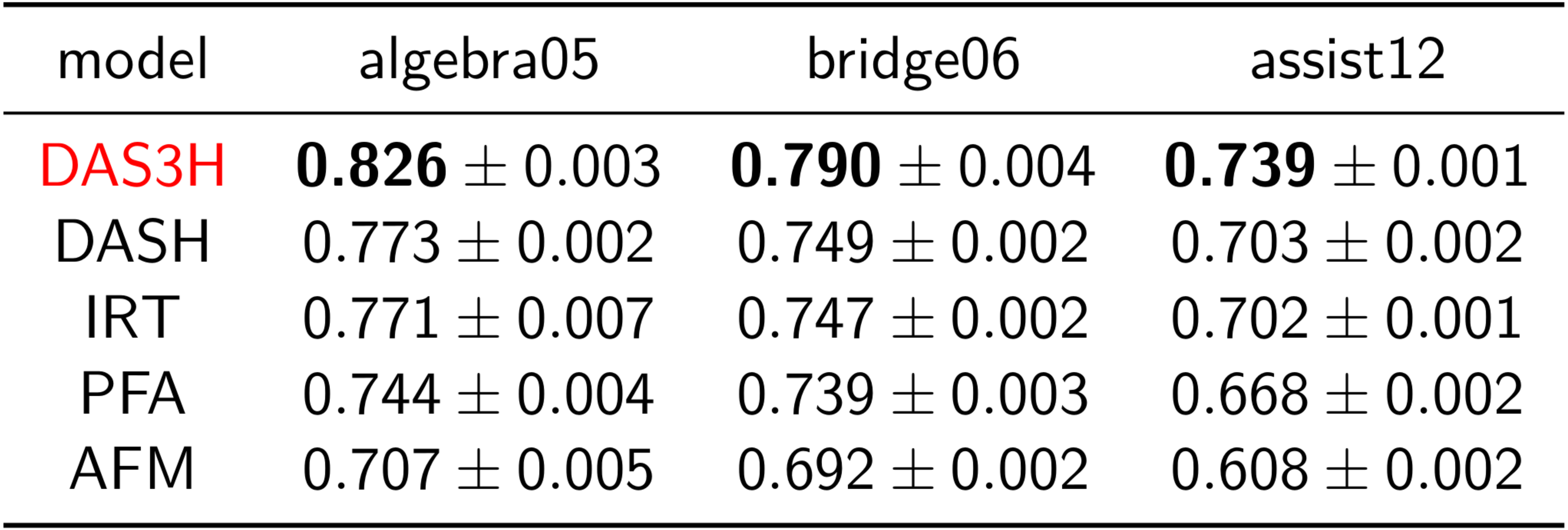
Datasets

- 3 datasets: ASSISTments 2012-2013, Bridge to Algebra 2006-2007 & Algebra I 2005-2006 (KDD Cup 2010)
  - Data consists of logs of student-item interactions on 2 ITS
  - Selected because they contain both timestamps and items with multiple skills → rare species in the EDM datasets fauna
- Preprocessing scheme: removed users with < 10 interactions, interactions with NaN skills, duplicates



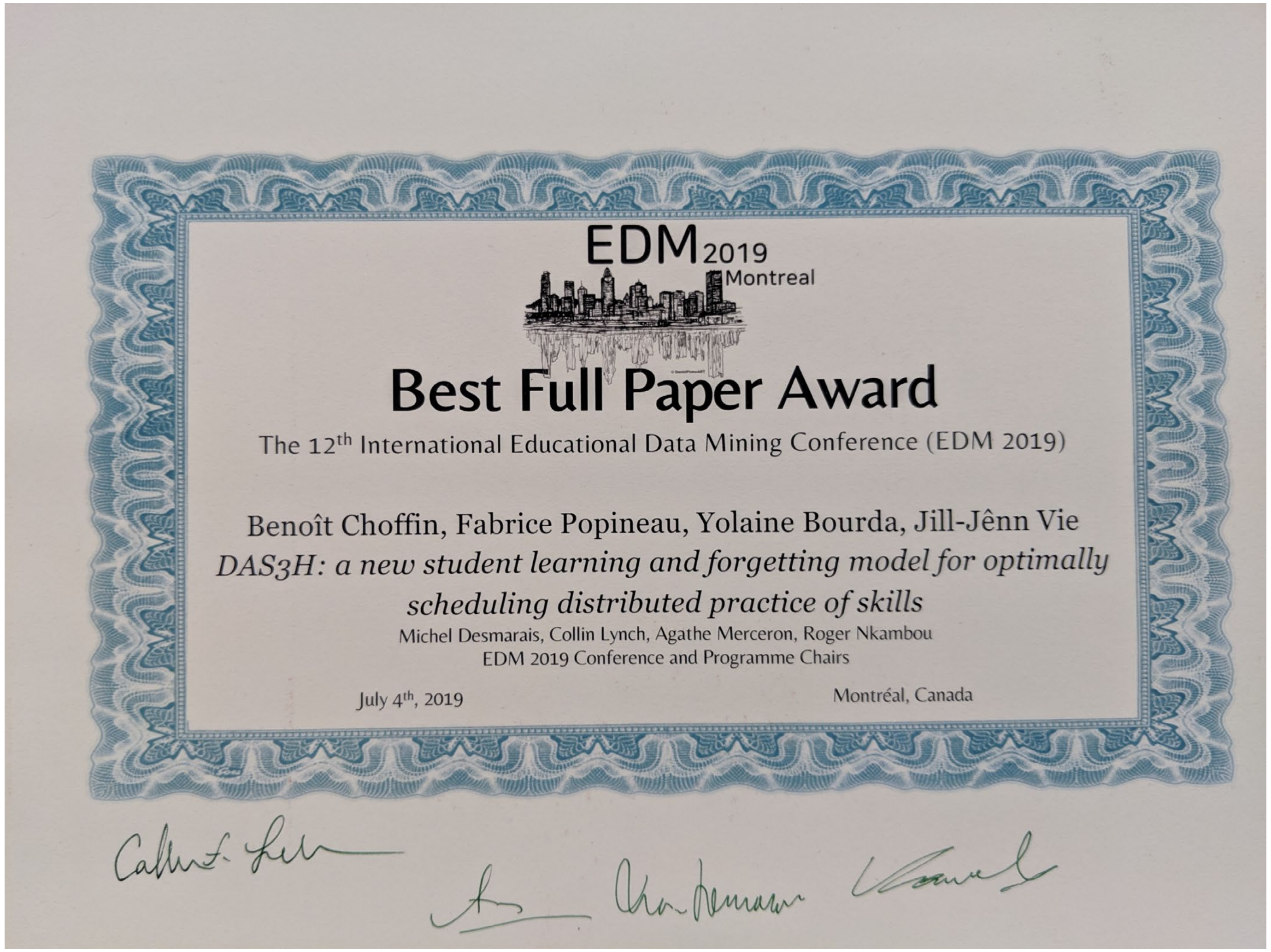
Dataset	Users	Items	Skills	Interactions	Mean correctness	Skills per item	Mean skill delay	Mean study period
assist12	24,750	52,976	265	2,692,889	0.696	1.000	8.54	98.3
bridge06	1,135	129,263	493	1,817,427	0.832	1.013	0.83	149.5
algebra05	569	173,113	112	607,000	0.755	1.363	3.36	109.9

Main Results



model	algebra05	bridge06	assist12
DAS3H	**0.826** ± 0.003	**0.790** ± 0.004	**0.739** ± 0.001
DASH	0.773 ± 0.002	0.749 ± 0.002	0.703 ± 0.002
IRT	0.771 ± 0.007	0.747 ± 0.002	0.702 ± 0.001
PFA	0.744 ± 0.004	0.739 ± 0.003	0.668 ± 0.002
AFM	0.707 ± 0.005	0.692 ± 0.002	0.608 ± 0.002

DAS3H



EDM2019 Montreal

Best Full Paper Award

The 12th International Educational Data Mining Conference (EDM 2019)

Benoît Choffin, Fabrice Popineau, Yolaine Bourda, Jill-Jênn Vie

DAS3H: a new student learning and forgetting model for optimally scheduling distributed practice of skills

Michel Desmarais, Collin Lynch, Agathe Merceron, Roger Nkambou
EDM 2019 Conference and Programme Chairs

July 4th, 2019

Montréal, Canada

Deep Learning

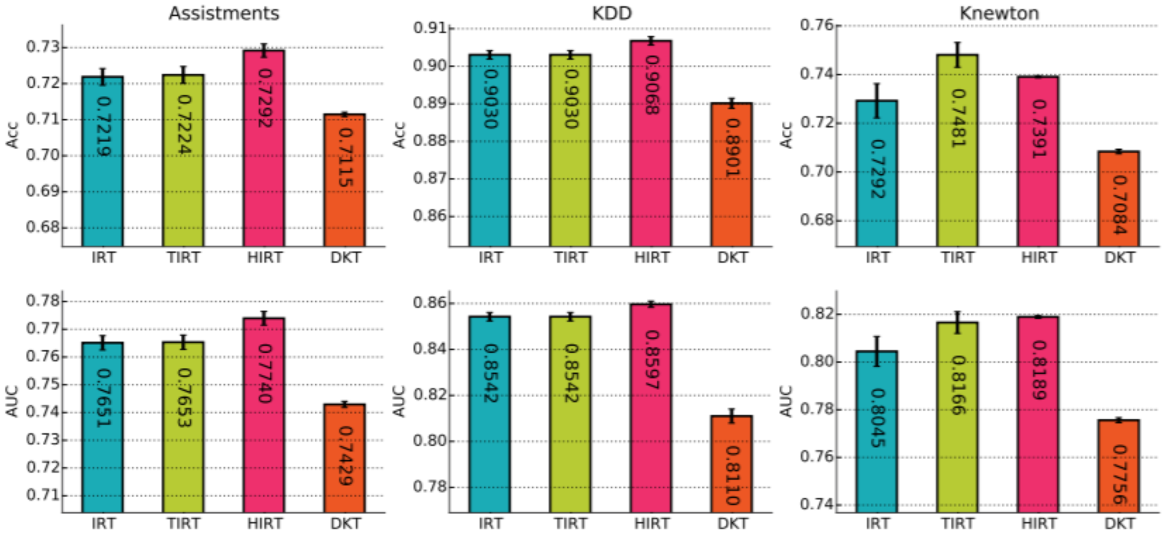
Deep Models

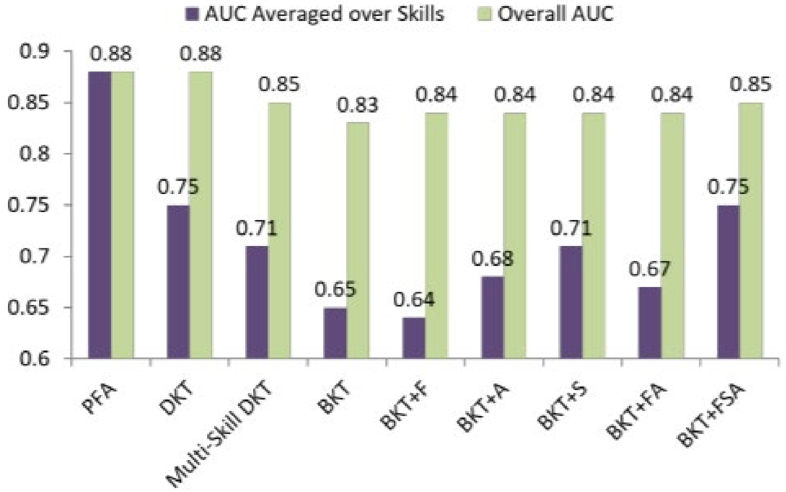
- And what about “deep-learning” ? (we are in 2019 after all!)
- Full line of papers about DKT, aka: Deep Knowledge Tracing
- What do these papers advocate?
  - Feature engineering
  - Better performance

Deep Models

- But wait … there also this line of papers:

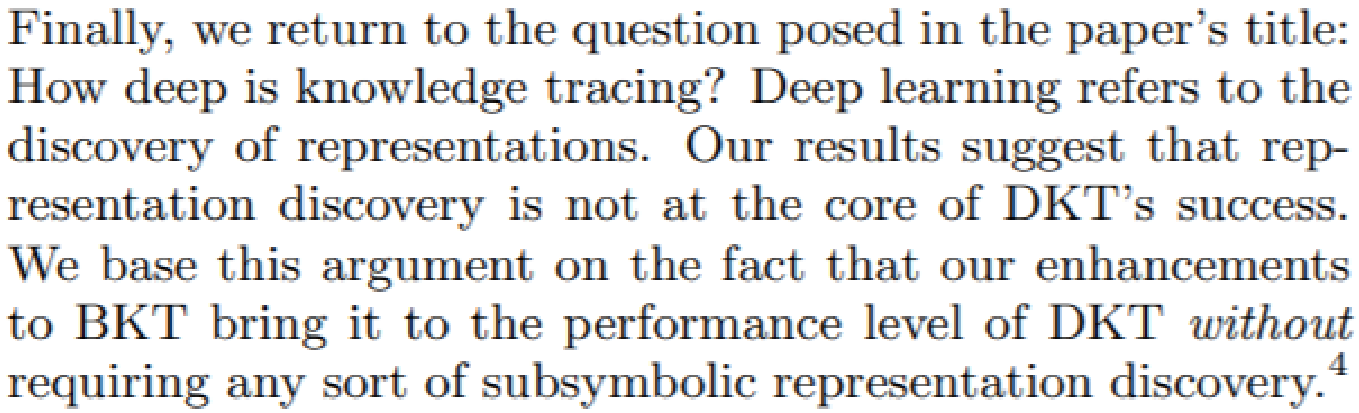
- Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation (Wilson et al. 2016)
- How deep is knowledge tracing? (Khadjah et al. 2016)
- Few hundred parameters outperform few hundred thousands? (Lalwani and Agraval 2017)





Deep Models

- So what? Have I been lied with the promises of Deep-Learning?



Finally, we return to the question posed in the paper's title: How deep is knowledge tracing? Deep learning refers to the discovery of representations. Our results suggest that representation discovery is not at the core of DKT's success. We base this argument on the fact that our enhancements to BKT bring it to the performance level of DKT *without* requiring any sort of subsymbolic representation discovery.[4]

(Kadhjah et al. 2016)

Conclusion

Hope you enjoyed it!

And that you will take the bite!