Seminar on Machine Teaching

Saarland University – Winter Semester 2019

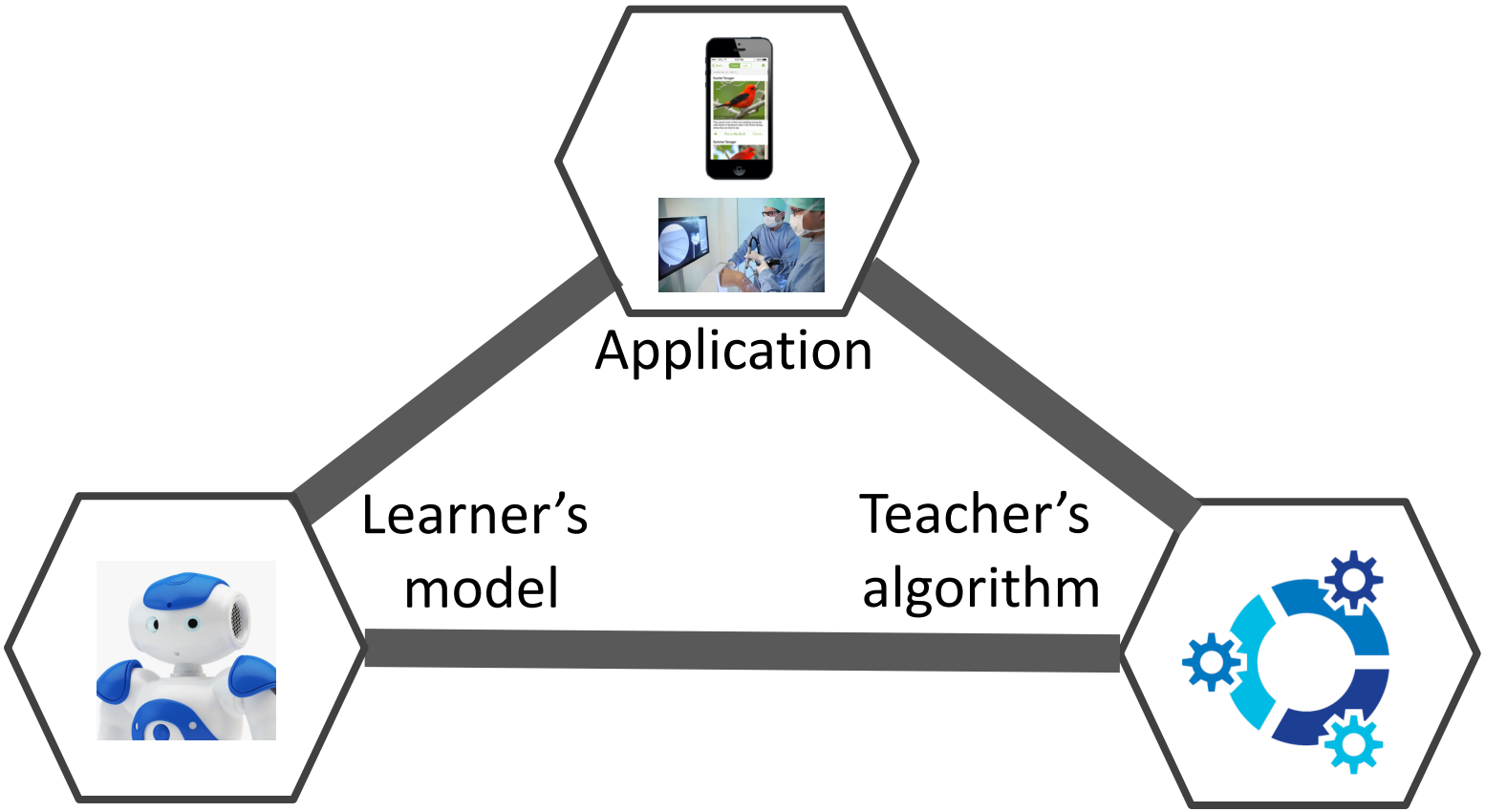
Adish Singla

5th Nov 2019

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

MAX-PLANCK-GESELLSCHAFT

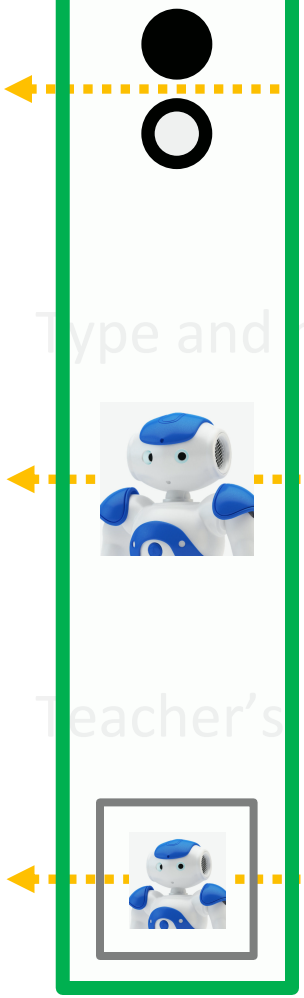
Machine Teaching: Key Components

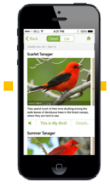


Machine Teaching: Problem Space

#1

- Type and complexity of task





duolingo

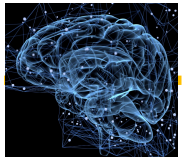
memrise

GROW YOUR MIND

coursera

edX

- Type and model of learning agent



- Teacher's knowledge and observability

Complexity Measures: TD

Notion of teaching complexity: Teaching dimension TD

- Introduced by [Goldman, Kearns '91]
- Analysis setting
  - randomized version space learner
  - worst-case analysis
  - finite size hypothesis class
  - exact teaching

Formal definition of TD

- Length of optimal teaching sequence for h^* is $|TS(h^*; \mathcal{H}, \mathcal{X})|$
- Teaching dimension is defined as

$$TD(\mathcal{H}, \mathcal{X}) := \max_{h^* \in \mathcal{H}} |TS(h^*; \mathcal{H}, \mathcal{X})|$$

Complexity Measures: TD

Examples for computing TD

	x_1	x_2	x_3	x_4	x_5	$\lvert TS(h^*)\rvert$
h_1	1	1	1	1	1	1
h_2	0	1	1	1	1	2
h_3	0	0	1	1	1	2
h_4	0	0	0	1	1	2
h_5	0	0	0	0	1	2

$$TD(\mathcal{H}, \mathcal{X}) = 2$$

	x_1	x_2	x_3	x_4	$\lvert TS(h^*)\rvert$
h_1	1	0	0	0	1
h_2	0	1	0	0	1
h_3	0	0	1	0	1
h_4	0	0	0	1	1
h_5	0	0	0	0	4

$$TD(\mathcal{H}, \mathcal{X}) = 4$$

Complexity Measures: VCD

Notion of learning complexity: VCD

- Introduced by [Vapnik, Chervonenkis '71]
- Sample complexity bounds for learning grow as $\Theta\big(VCD(\mathcal{H}, \mathcal{X})\big)$

Formal definition of ``shattering" and VCD

- Consider a set $\mathrm{C} \subseteq \mathcal{X}$ of size d given by $\mathrm{C} = \{c_1, c_2, \ldots, c_d\}$
  - Pattern of h on C is given by $P(h, C) \coloneqq \big(h(c_1), h(c_2), \ldots, h(c_d)\big)$
  - Unique patterns of $\mathcal{H}$ on C are given by $P(\mathcal{H}, C) \coloneqq \cup_{h \in \mathcal{H}} \{P(h, C)\}$
- $\mathcal{H}$ shatters C if $|P(\mathcal{H}, C)| = 2^d$
- $\mathcal{H}$ has VCD of d, i.e., $VCD(\mathcal{H}, \mathcal{X}) = d$ when
  1. $\exists\, \mathrm{C} \subseteq \mathcal{X}$ of size d such that $\mathcal{H}$ shatters C
  2. $\nexists\, \mathrm{C} \subseteq \mathcal{X}$ of size $d + 1$ such that $\mathcal{H}$ shatters C

Complexity Measures: VCD

Examples for computing VCD

	x_1	x_2	x_3	x_4
h_1	1	0	0	0
h_2	0	1	0	0
h_3	0	0	1	0
h_4	0	0	0	1
h_5	0	0	0	0

$$VCD(\mathcal{H}, \mathcal{X}) = 1$$

	x_1	x_2	x_3	x_4
h_1	1	0	0	0
h_2	0	1	0	0
h_3	0	0	1	0
h_4	0	0	0	1
h_5	0	0	0	0
h_6	0	0	1	1

$$VCD(\mathcal{H}, \mathcal{X}) = 2$$

Complexity Measures: TD vs. VCD

A fundamental question: TD vs. VCD?

- $TD(\mathcal{H}, \mathcal{X})$ is $O\big(VCD(\mathcal{H}, \mathcal{X})\big)$?
- There exist problems with
  - $TD(\mathcal{H}, \mathcal{X}) \ll O\big(VCD(\mathcal{H}, \mathcal{X})\big)$
  - $TD(\mathcal{H}, \mathcal{X}) \gg O\big(VCD(\mathcal{H}, \mathcal{X})\big)$

	x_1	x_2	x_3	x_4
h_1	1	0	0	0
h_2	0	1	0	0
h_3	0	0	1	0
h_4	0	0	0	1
h_5	0	0	0	0

$$TD(\mathcal{H}, \mathcal{X}) = 4$$

$$VCD(\mathcal{H}, \mathcal{X}) = 1$$

Improved Notions of TD: RTD

Teaching an "adversarial" learner: Classic TD

- Simple classes can be difficult to teach

Teaching a "cooperative" learner: Recursive TD (RTD)

- Introduced by [Zilles et al. @ COLT'08]
- $RTD(\mathcal{H}, \mathcal{X})$ is $O\big(VCD(\mathcal{H}, \mathcal{X})\big)$? [Simon, Zilles @ COLT'15]
- An active area of research
  - $O\big(d\ 2^d \log\log|\mathcal{H}|\big)$ [Moran et al. @ FOCS'15]
  - $O\big(d\ 2^d\big)$ [Chen et al. @ NIPS' 16]
  - $O(d^2)$ [Hu et al. @ COLT' 17]

  where d denotes $VCD(\mathcal{H}, \mathcal{X})$

Improved Notions of TD: Unified View

Key limitations of existing models and measures

- A number of complexity measures: Classic-TD, RTD, PBTD, NCTD, …
- Batch teaching
  - Order of examples does not matter
  - Learner's feedback does not matter
- Teaching complexity not linear in VCD

Our recent work

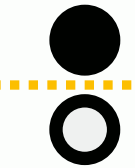
- ``Preference-Based Batch and Sequential Teaching: Towards a Unified View of Models" [NeurIPS'19]
- ``Understanding the Role of Adaptivity in Machine Teaching: The Case of Version Space Learners" [NeurIPS'18]

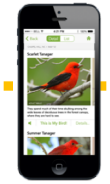
Machine Teaching: Problem Space

- Type and complexity of task
- Type and model of learning agent
- Teacher's knowledge and observability

#1

#2, #3



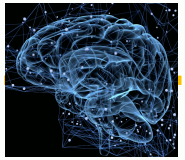


duolingo

memrise
GROW YOUR MIND

coursera

edX



Cognitive Model of Skill Acquisition

Cognitive tutors

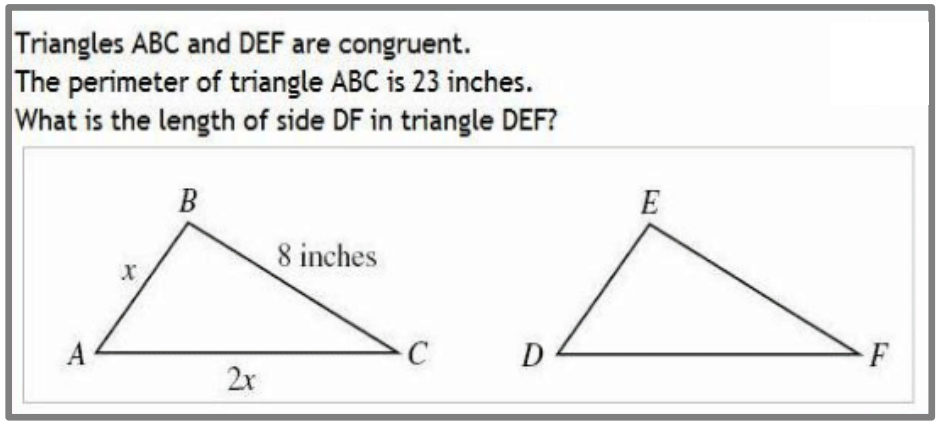
- Used by millions of students for K-12 education
  - https://www.carnegielearning.com/
  - https://new.assistments.org/

Bayesian Knowledge Tracing (BKT)

- Introduced by [Corbett, Anderson '95]
- Knowledge Components (KC)
  - A learning task is associated with a set of skills
  - Practicing a skill leads to mastery of that skill

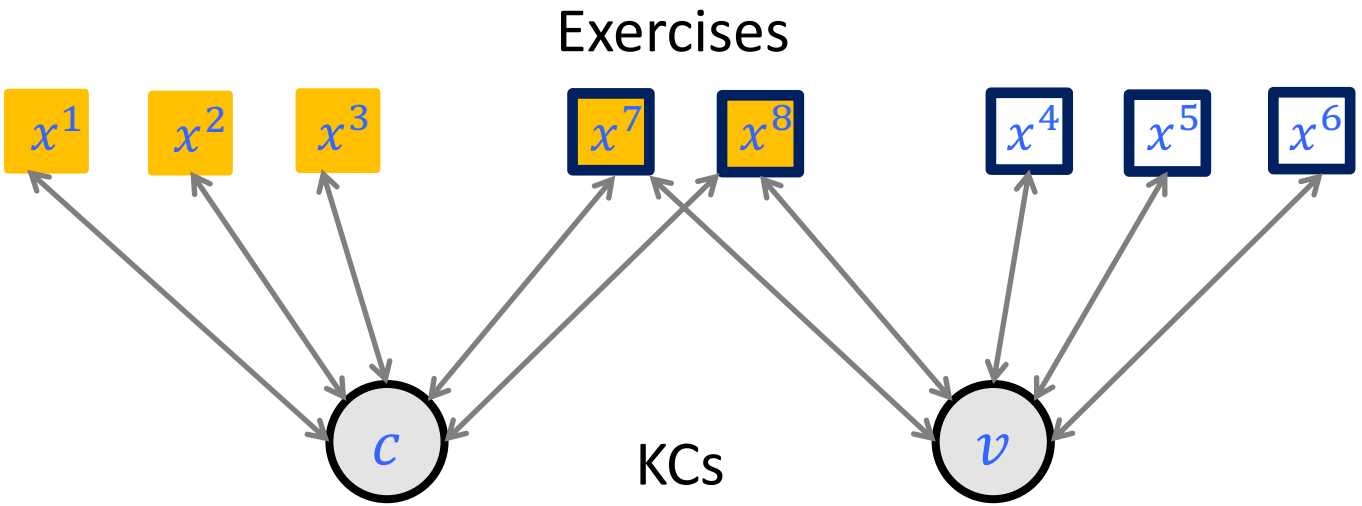
Task: Geometry and Algebra

Knowledge components (KCs) and exercises



$k = c$: Congruent triangles

$k = v$: One variable equations



Teaching Interaction under BKT

- Each KC k is associated with a knowledge state h^k
  - $h^k = 1$ represents that the skill has been mastered
  - $h^k = 0$ otherwise

Interaction at time $t = 1, 2, \ldots T$

- Denote the value of h^k at the end of time t as h_t^k
- Initialize h_0^k for all KCs
- At time t:
  - Teacher provides exercise x_t associated with KC k
  - Learner responds $y_t \in \{0, 1\}$ with knowledge h_{t-1}^k
  - Learner updates knowledge from h_{t-1}^k to h_t^k

BKT Learner Model

Learner's initial knowledge (one parameter per KC)

- Probability of *mastery before teaching* $P^k_{\text{init}} := P(h^k_0 = 1)$

Learner's response (two parameters per KC)

- Conditional probability of *guessing* $P^k_{\text{guess}} := P\left(y_t = 1 \mid h^k_{t-1} = 0\right)$
- Conditional probability of *slipping* $P^k_{\text{slip}} := P\left(y_t = 0 \mid h^k_{t-1} = 1\right)$

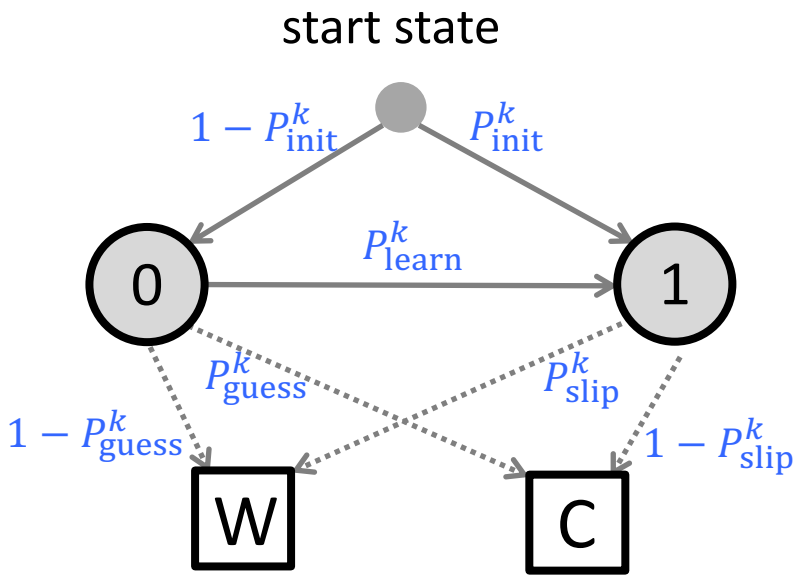
Learner's update (one parameter per KC)

- Conditional probability of *learning* $P^k_{\text{learn}} := P\left(h^k_t = 1 \mid h^k_{t-1} = 0\right)$

BKT Learner Model: HMM Representation

Hidden Markov Model (HMM) for a single KC k

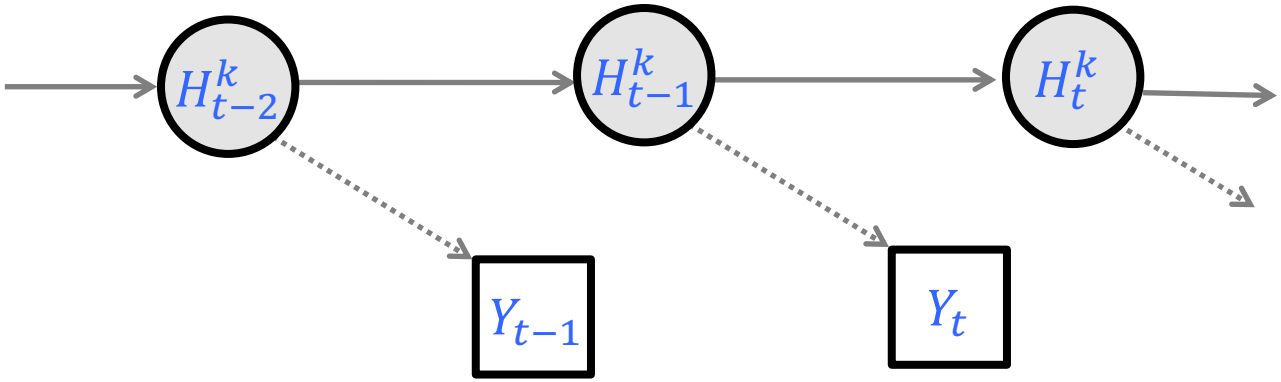
h^k knowledge states (unobserved)



Responses: **W**rong or **C**orrect (observed)

BKT Learner Model: DBN Representation

Dynamic Bayesian Network (DBN) for a single KC k



$P(H^k_0 = 1)$

P^k_{init}

$P(Y_t = 1 \mid H^k_{t-1})$

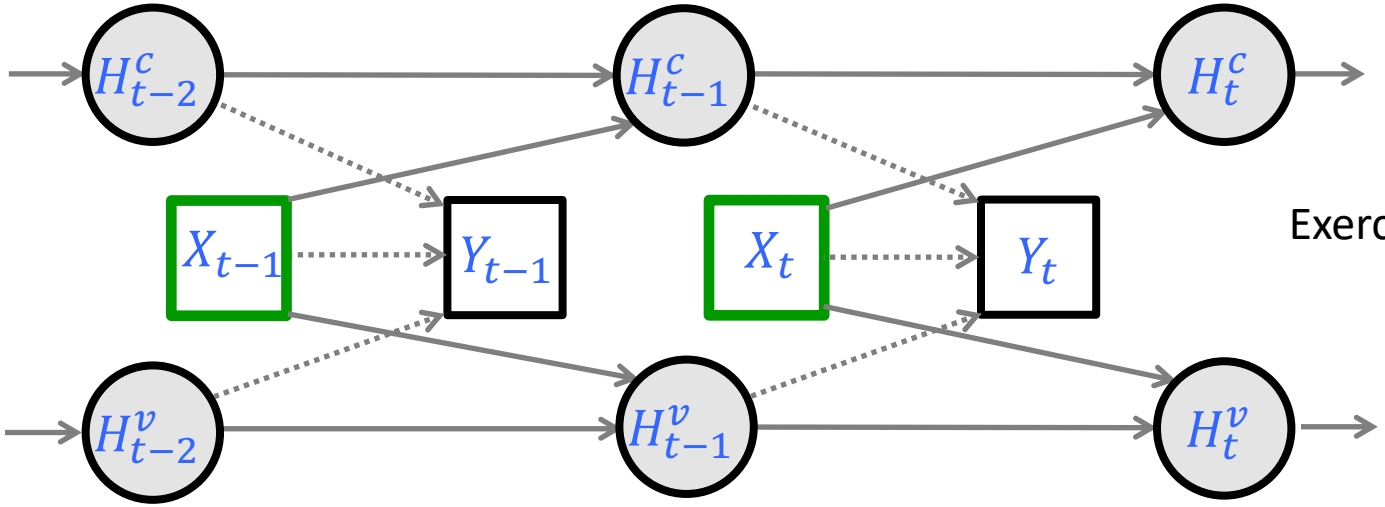
$H^k_{t-1} = 0$	P^k_{guess}
$H^k_{t-1} = 1$	$1 - P^k_{\text{slip}}$

$P(H^k_t = 1 \mid H^k_{t-1})$

$H^k_{t-1} = 0$	P^k_{learn}
$H^k_{t-1} = 1$	1

BKT Learner Model: DBN Representation

Dynamic Bayesian Network for two independent KCs $\{c, v\}$



Exercise X is chosen by teacher and takes value $\{c, v\}$

$P(Y_t = 1 \mid H^c_{t-1}, H^v_{t-1}, X_t)$

	$X_t = c$	$X_t = v$
$H^c_{t-1} = 0, H^v_{t-1} = 0$	P^c_{guess}	P^v_{guess}
$H^c_{t-1} = 1, H^v_{t-1} = 0$	$1 - P^c_{\text{slip}}$	P^v_{guess}
$H^c_{t-1} = 0, H^v_{t-1} = 1$	P^c_{guess}	$1 - P^v_{\text{slip}}$
$H^c_{t-1} = 1, H^v_{t-1} = 1$	$1 - P^c_{\text{slip}}$	$1 - P^v_{\text{slip}}$

$P(H^c_t = 1 \mid H^c_{t-1}, X_t)$

$H^c_{t-1} = 0, X_t = c$	P^c_{learn}
$H^c_{t-1} = 1, X_t = c$	1
$H^c_{t-1} = 0, X_t = v$	0
$H^c_{t-1} = 1, X_t = v$	1

BKT Teacher

Prediction and inference for a single KC k

- Learner's responses at the end of time t: $D_t := \{y_1, y_2, \dots, y_t\}$
- Predicting learner's response: $P\left(Y_t^k = 1 \mid D_{t-1}\right)$
- Inferring learner's knowledge: $P\left(H_t^k = 1 \mid D_t\right)$ denoted as θ_t^k

Incremental computations

- Initial $\theta_0^k = P_{\text{init}}^k$ is known
- Compute $P\left(Y_t^k = 1 \mid D_{t-1}\right)$ from θ_{t-1}^k
- Compute θ_t^k from θ_{t-1}^k and y_t

BKT Teacher

Predicting learner's response

$$P\left(Y_t^k = 1 \mid D_{t-1}\right) = \left(1 - P_{\text{slip}}^k\right) \cdot \theta_{t-1}^k + P_{\text{guess}}^k \cdot \left(1 - \theta_{t-1}^k\right)$$

Derivation:

$$P\left(Y_t^k = 1 \mid D_{t-1}\right) = P\left(Y_t^k = 1, H_{t-1}^k = 1 \mid D_{t-1}\right) + P\left(Y_t^k = 1, H_{t-1}^k = 0 \mid D_{t-1}\right)$$

$$= P\left(Y_t^k = 1 \mid H_{t-1}^k = 1, D_{t-1}\right) \cdot P\left(H_{t-1}^k = 1 \mid D_{t-1}\right) + P\left(Y_t^k = 1 \mid H_{t-1}^k = 0, D_{t-1}\right) \cdot P\left(H_{t-1}^k = 0 \mid D_{t-1}\right)$$

$$= P\left(Y_t^k = 1 \mid H_{t-1}^k = 1\right) \cdot P\left(H_{t-1}^k = 1 \mid D_{t-1}\right) + P\left(Y_t^k = 1 \mid H_{t-1}^k = 0\right) \cdot P\left(H_{t-1}^k = 0 \mid D_{t-1}\right)$$

$$= \left(1 - P_{\text{slip}}^k\right) \cdot \theta_{t-1}^k + P_{\text{guess}}^k \cdot \left(1 - \theta_{t-1}^k\right)$$

BKT Teacher

Inferring learner's knowledge

$$P\left(H_t^k = 1 \mid D_t\right) = \hat{\theta}_{t-1}^k + P_{\text{learn}}^k \cdot \left(1 - \hat{\theta}_{t-1}^k\right)$$

where $\hat{\theta}_{t-1}^k$ is an intermediate quantify computed from θ_{t-1}^k and y_t

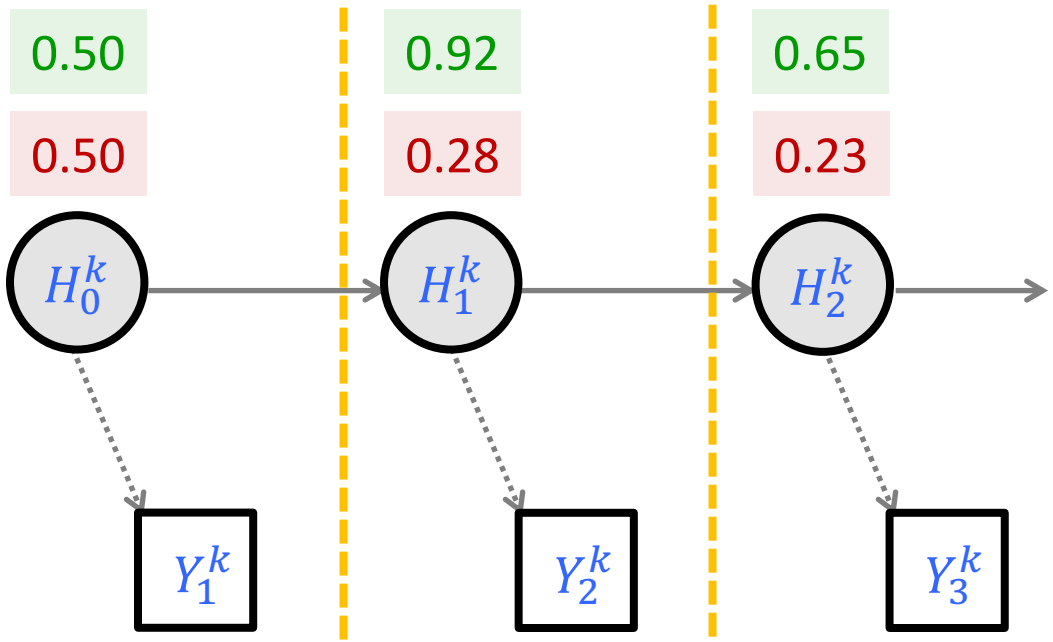
Computing $\hat{\theta}_{t-1}^k$ by applying Bayes rule

- For $y_t = 1$, $\hat{\theta}_{t-1}^k := \frac{\left(1-P_{\text{slip}}^k\right)\cdot\theta_{t-1}^k}{\left(1-P_{\text{slip}}^k\right)\cdot\theta_{t-1}^k + P_{\text{guess}}^k\cdot\left(1-\theta_{t-1}^k\right)}$
- For $y_t = 0$, $\hat{\theta}_{t-1}^k := \frac{P_{\text{slip}}^k\cdot\theta_{t-1}^k}{P_{\text{slip}}^k\cdot\theta_{t-1}^k + \left(1-P_{\text{guess}}^k\right)\cdot\left(1-\theta_{t-1}^k\right)}$

BKT Teacher

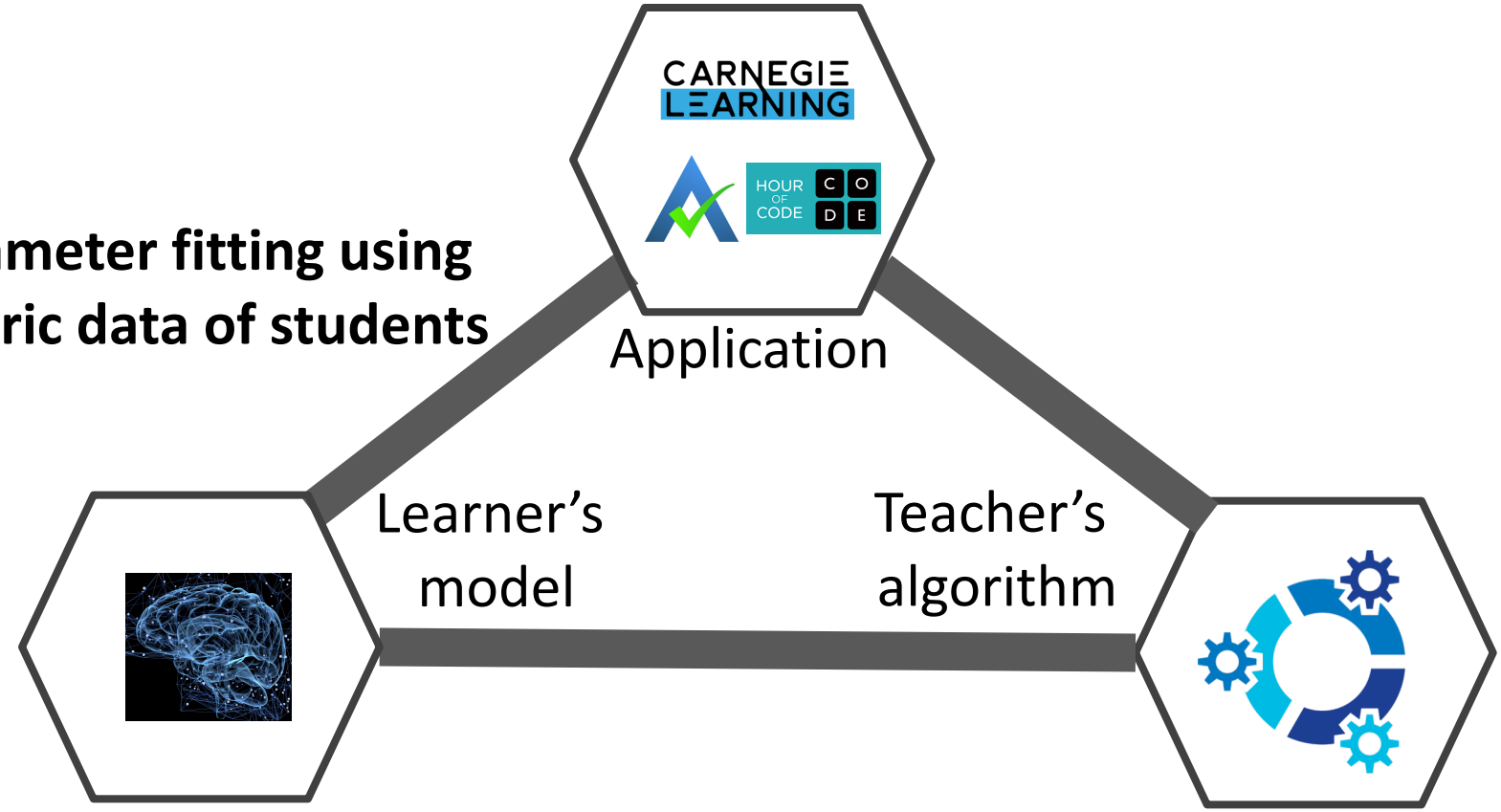
An example of prediction and inference

- Parameters: $P^k_{\text{init}} = 0.5$, $P^k_{\text{learn}} = 0.2$, $P^k_{\text{guess}} = 0.1$, $P^k_{\text{slip}} = 0.1$



Teaching Process using BKT

Parameter fitting using historic data of students



- Datasets publicly available
- Parameter fitting by standard techniques

BKT: Two Main Research Themes

Improving learner model

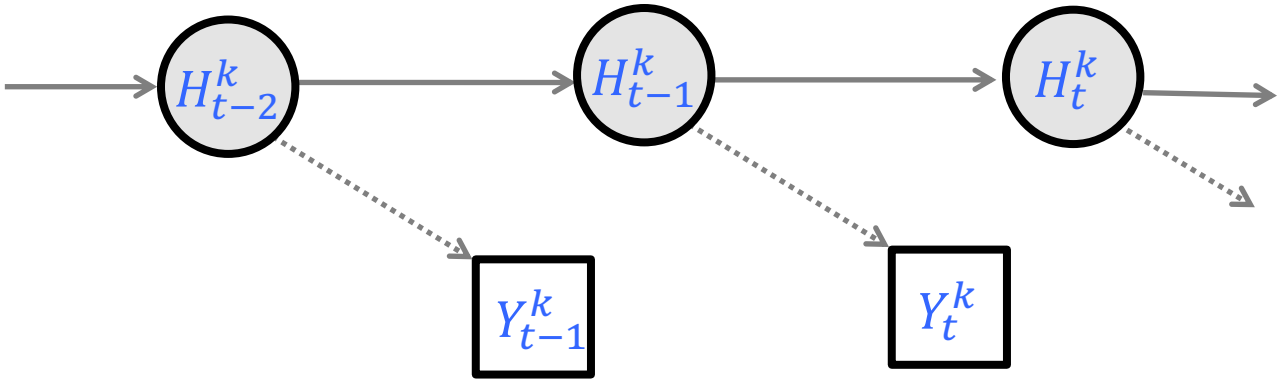
- Forgetting
- Individualization per student
- Skill discovery
  - exercises to skills mapping
  - Inter-skill similarity and prerequisite structure

Designing teaching policies

- When to stop teaching a skill?
- Optimizing the curriculum via planning in DBN

Improved Learner Models for BKT

DBN for a single KC k with forgetting



$P(H^k_0 = 1)$

P^k_{init}

$P(Y_t = 1 \mid H^k_{t-1})$

$H^k_{t-1} = 0$	P^k_{guess}
$H^k_{t-1} = 1$	$1 - P^k_{\text{slip}}$

$P(H^k_t = 1 \mid H^k_{t-1})$

$H^k_{t-1} = 0$	P^k_{learn}
$H^k_{t-1} = 1$	$1 - P^k_{\text{forget}}$

Improved Learner Models for BKT

Comparing different models [Khajah, Lindsey, Mozer @ EDM'16]

- **BKT**: Standard model
  - **BKT_1:** One model for all skills
  - **BKT_2:** Multiple models, one per skill
- **BKT-F**: With forgetting
- **BKT-I**: Individualization per student
- **BKT-S**: Skill discovery as part of BKT
- **BKT-FIS**: Above three extensions combined

Improved Learner Models for BKT

Comparing different models [Khajah, Lindsey, Mozer @ EDM'16]

- Dataset from 
  - # students: 15,900
  - # skills: 124 (with multiple exercises per skill)
  - # student-exercise attempts: 0.5 million
- Cross-validation by splitting data based on student ids
- Performance metric: AUC (ranging from 0.5 to 1)

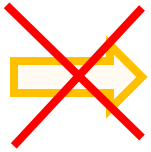
BKT_1	BKT_2	BKT-F	BKT-I	BKT-S	BKT-FIS	Deep BKT
0.67	0.73	**0.83**	0.785	0.76	0.825	**0.86**

Deep Knowledge Tracing [Piech et al. @ NIPS'15]

Designing Teaching Policies

Much less research on designing teaching policies

- The most popular way of using BKT for teaching is
  - STOP teaching skill k when $P\left(H_t^k = 1 \mid D_t\right) \geq 0.95$
- Planning techniques
  - Faster teaching via POMDP Planning [Rafferty et al. @ CogSci'16]
- "When to stop" instructional policies with guarantees
  - When to stop? Towards Universal Instructional Policies [Käser, Klingler, Gross @ LAK'16]
  - From Predictive Models to Instructional Policies [Rollinson, Brunskill @ LAK'15]

Better predictive models  Better instructional policies

Cognitive Models of Skill Acquisition

Summary of BKT

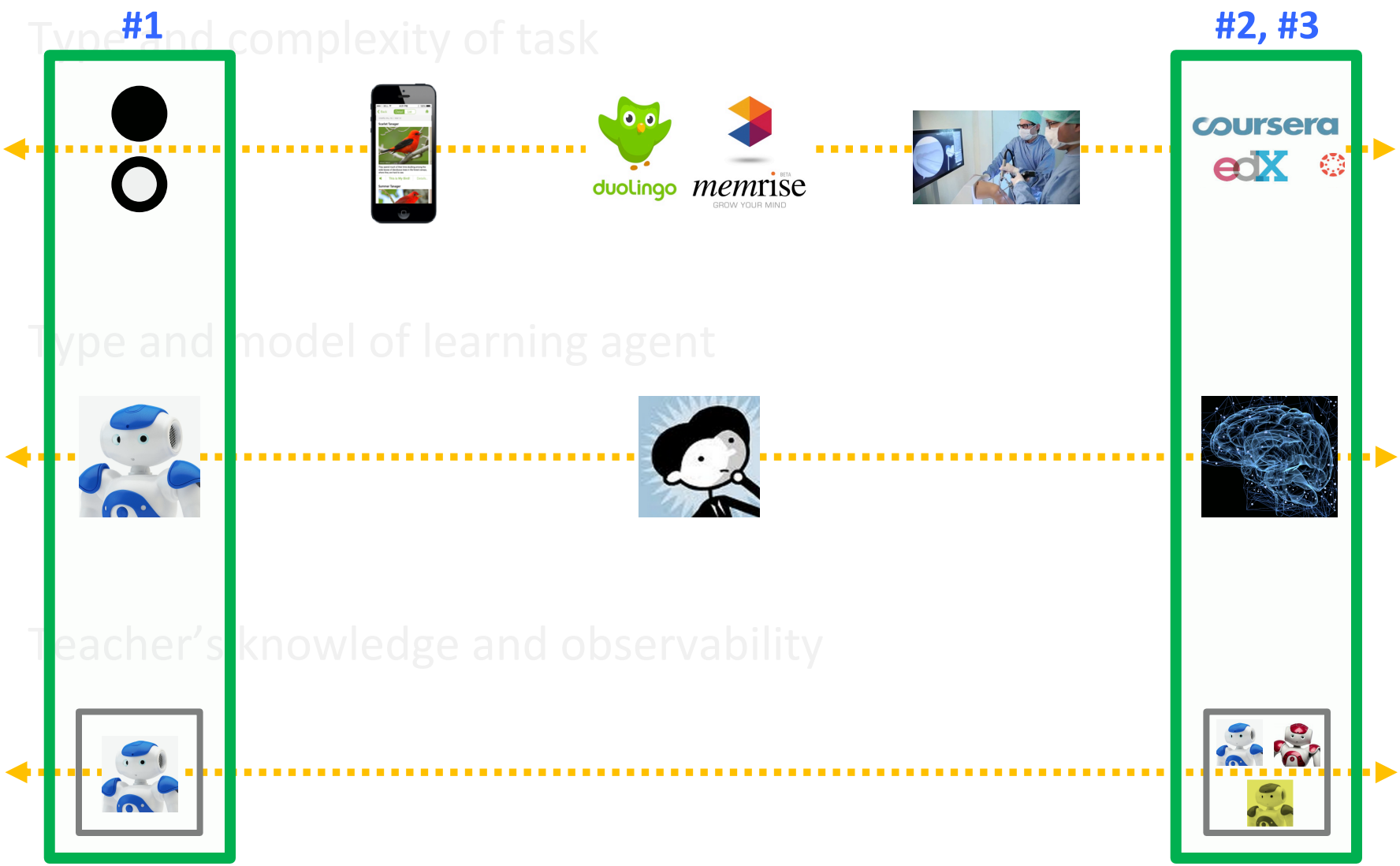
- Well-studied cognitive model, used in real-world applications
- Generic model for complex learning tasks (e.g., learning Algebra)

Limitations of using cognitive models

- Difficult to design optimal teaching policies
- Generic models but might not capture fine-grained task details

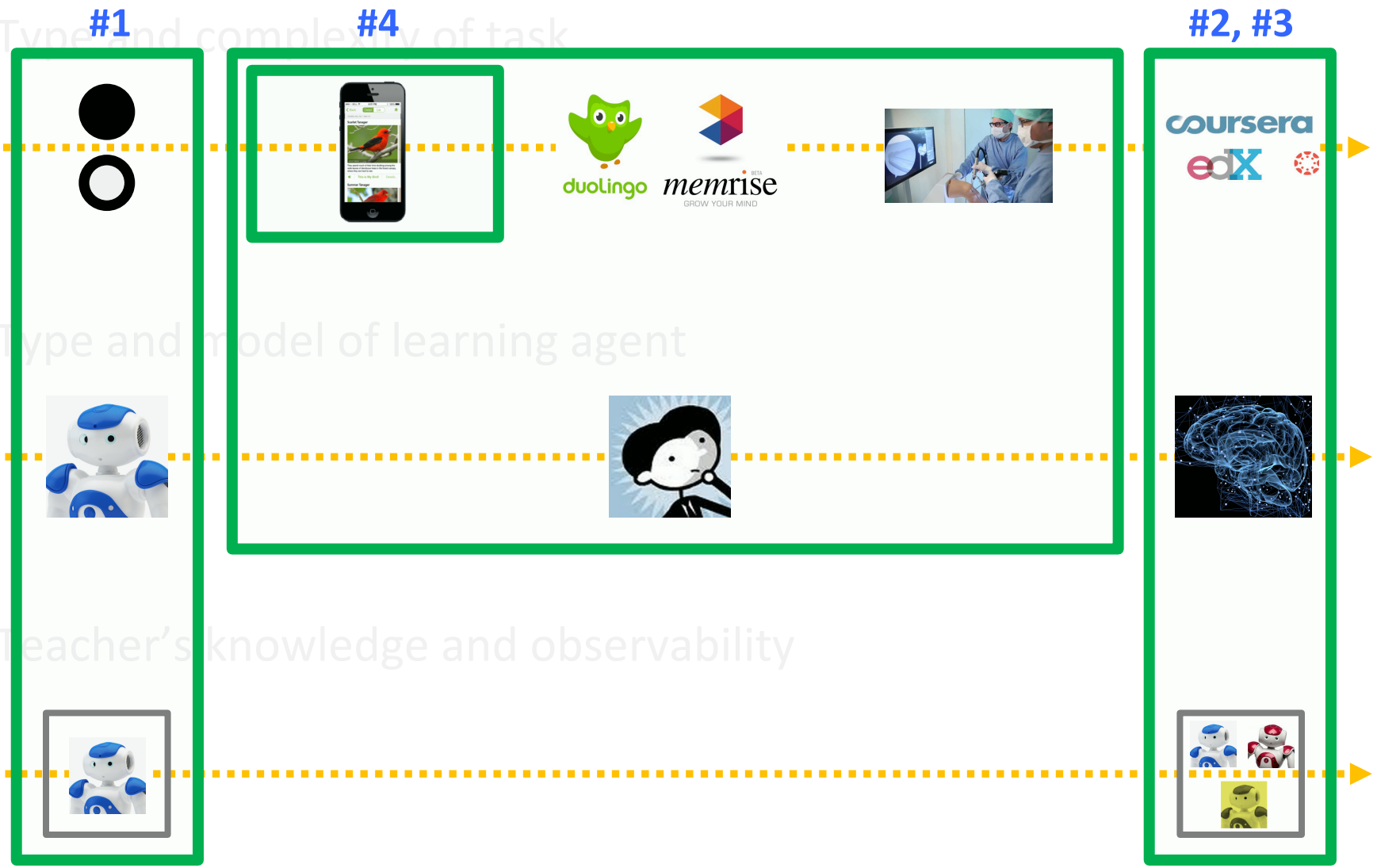
Machine Teaching: Problem Space

- Type and complexity of task
- Type and model of learning agent
- Teacher's knowledge and observability

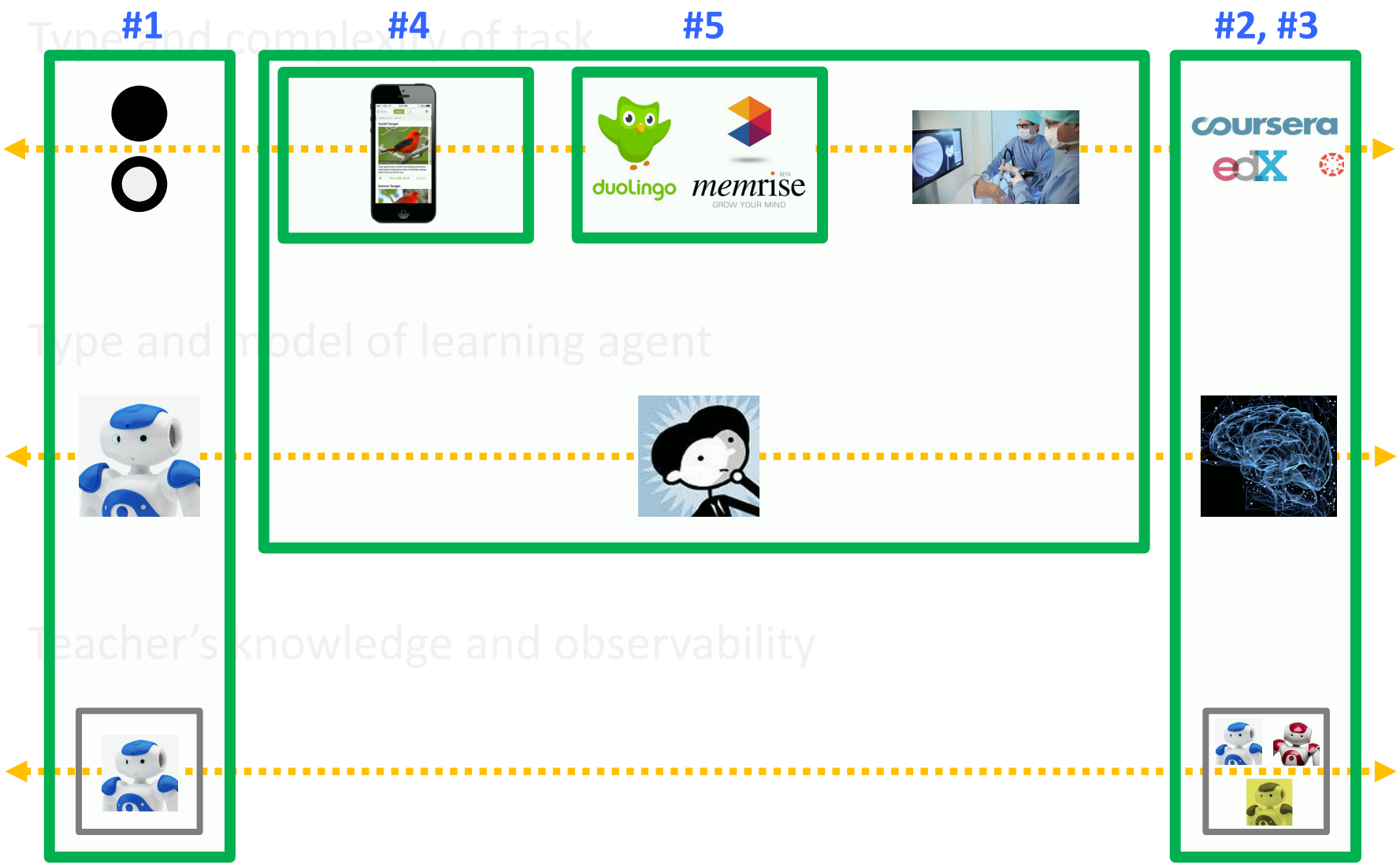


Machine Teaching: Problem Space

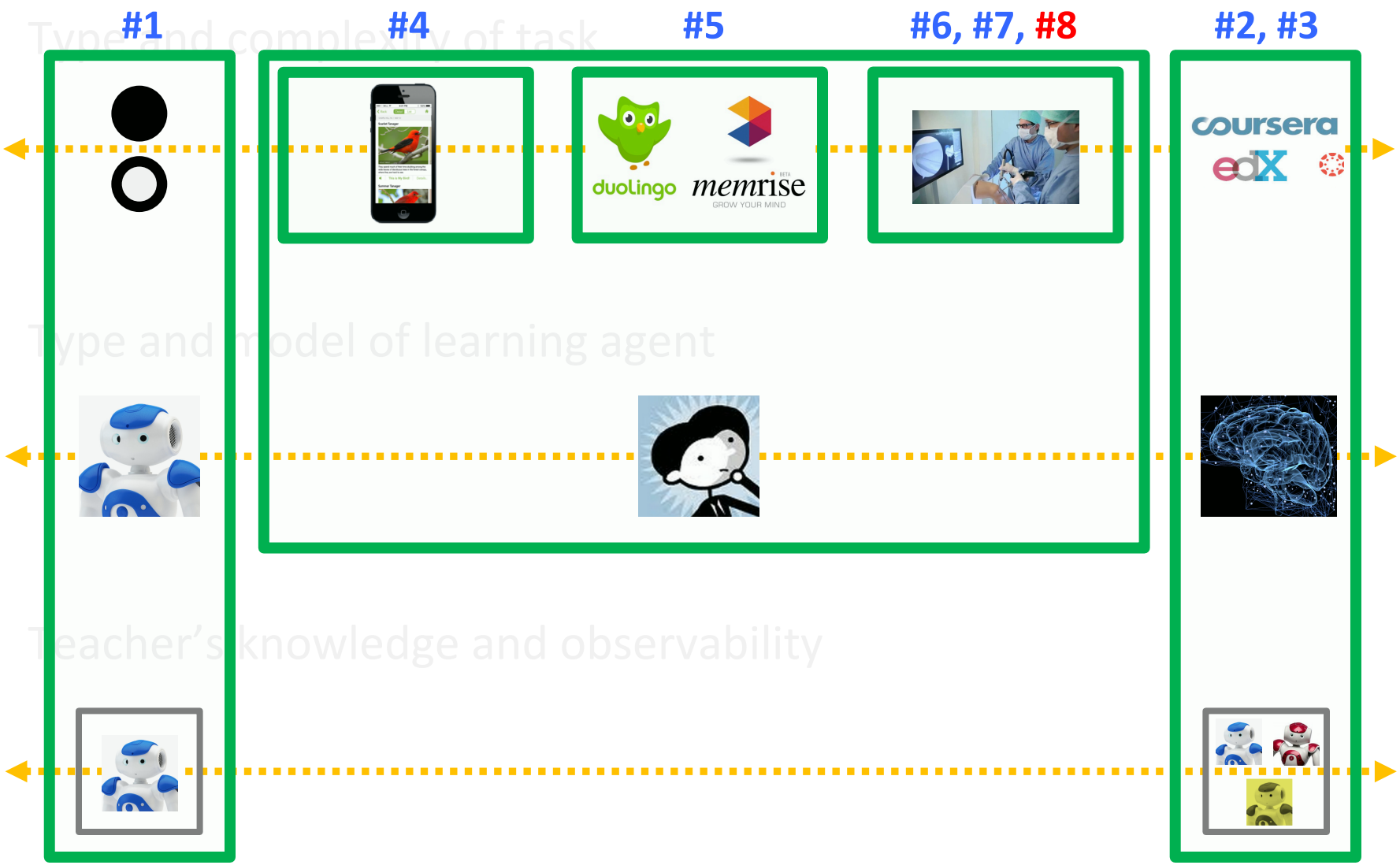
- Type and complexity of task
- Type and model of learning agent
- Teacher's knowledge and observability



Machine Teaching: Problem Space



Machine Teaching: Problem Space



Sequential Decision Making: Ingredients

Key ingredients

- A sequence of actions with long term consequences
- Delayed feedback
  - Safely reaching the destination in time
  - Successfully solving the exercise
  - Winning or losing a game
- Main components
  - **Environment** representing the problem
  - **Student** is the learning agent taking actions
  - **Teacher** helping the student to learn faster

Sequential Decision Making: Environment

Markov Decision Process $M := (S, A, P, S_{init}, S_{end}, R)$

- S: states of the environment
- A: actions that can be taken by agent
- $\mathrm{P}(s'|s, a)$: the transition of the environment when action is taken
- S_{init}: defines a set of initial states
- S_{end}: defines a set of terminal states
- $R(s, a)$: reward function

Sequential Decision Making: Policy

Agent's policy π

- $\pi(s) \rightarrow a$: A deterministic policy
- $\pi(s) \rightarrow P(a)$: A stochastic policy

Utility of a policy

- Expected total reward when executing a policy π is given by

$$U^{\pi} = \mathbb{E}_{P,\pi}\left[\sum_{\tau} R(s_{\tau}, a_{\tau})\right]$$

- Agent's goal is to learn an optimal policy

$$\pi^{*} = \text{argmax}_{\pi}\, U^{\pi}$$

An Example: Car Driving Simulator

- State s represented by a feature vector $\phi(s)$

  (location, speed, acceleration, car-in-front, HOV, …)

- Action a could be discrete/continuous

  {left, straight, right, brake, speed+, speed-, …}

- Transition $P(s'|s,a)$ defines how world evolves

  (stochastic as it depends on other drivers in the environment)

- $R(s,a)$ defines immediate reward, e.g.,
  - 100 if $s \in S_{end}$
  - -1 if $s \notin S_{end}$
  - -10 if s represents ``accident"

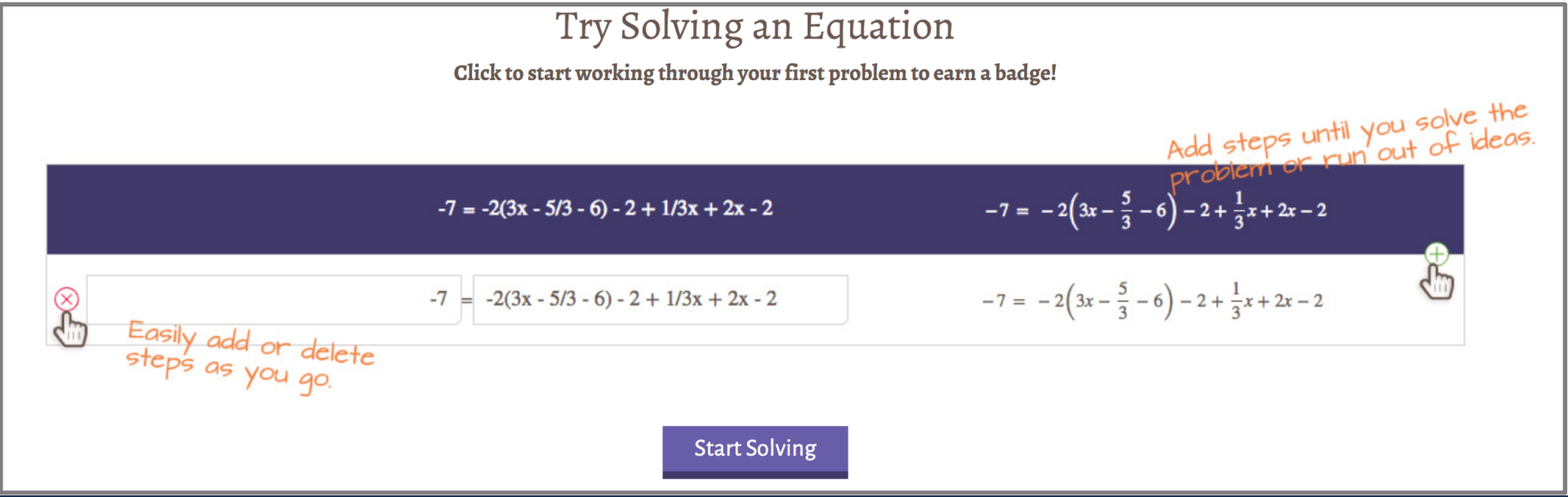
- Policy π^* dictates how an agent should drive

An Example: Tutoring System for Algebra

- State s represents the current layout of variables
- Action a could be {move, combine, distribute, stop, …}
- Transition $P(s'|s,a)$ is deterministic
- $R(s,a)$ defines immediate reward, e.g.,
  - 100 if $s \in S_{end}$
  - -1 if $s \notin S_{end}$

ASSISTments

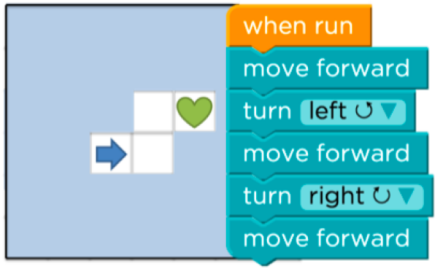
EMMY'S Workshop



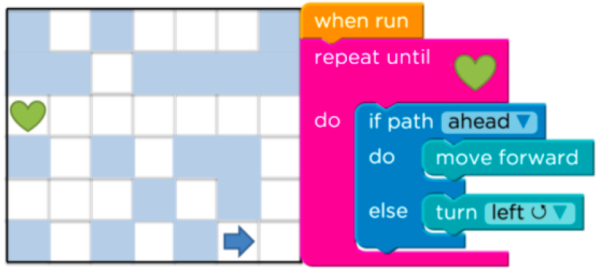
An Example: Tutoring System for Coding

- State s could be represent
  - raw source code
  - abstract syntax tree (AST)
  - execution behavior
  - …
- Action a could be eligible updates (e.g., allowed by the interface)

HoC Problem 4



HoC Problem 18



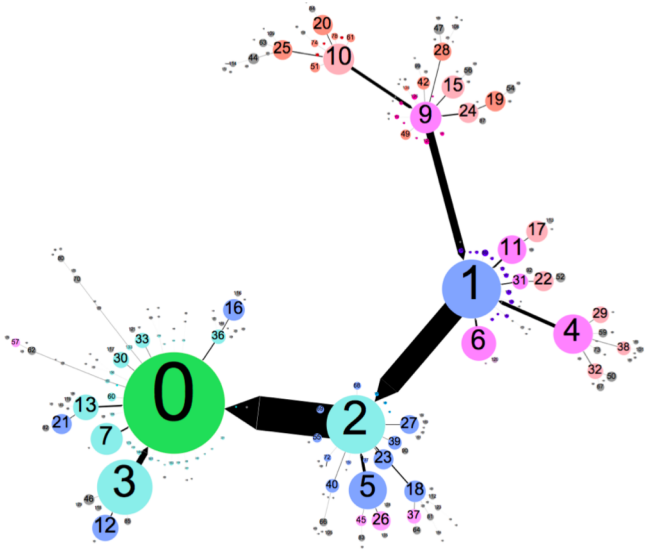


Image credits: [Piech et al. @ LAK'15]

Learning Settings: Rewards

- Standard setting in reinforcement learning (RL)
- P is known, R is known
  - Mode-based planning algorithms (e.g., Value iteration)
- P, R are both unknown
  - Model-free learning algorithms (e.g., Q-learning)

(Book) Reinforcement Learning: An Introduction [Barto and Sutton 2018]

Learning Settings: Demonstrations

- Also known as "**Imitation Learning**"
  - Learning via observing behavior of another agent
- A popular framework: Inverse reinforcement learning (IRL)
  - Recover reward function explaining observed demonstrations
  - E.g., Maximum Causal Entropy algorithm (MCE-IRL)

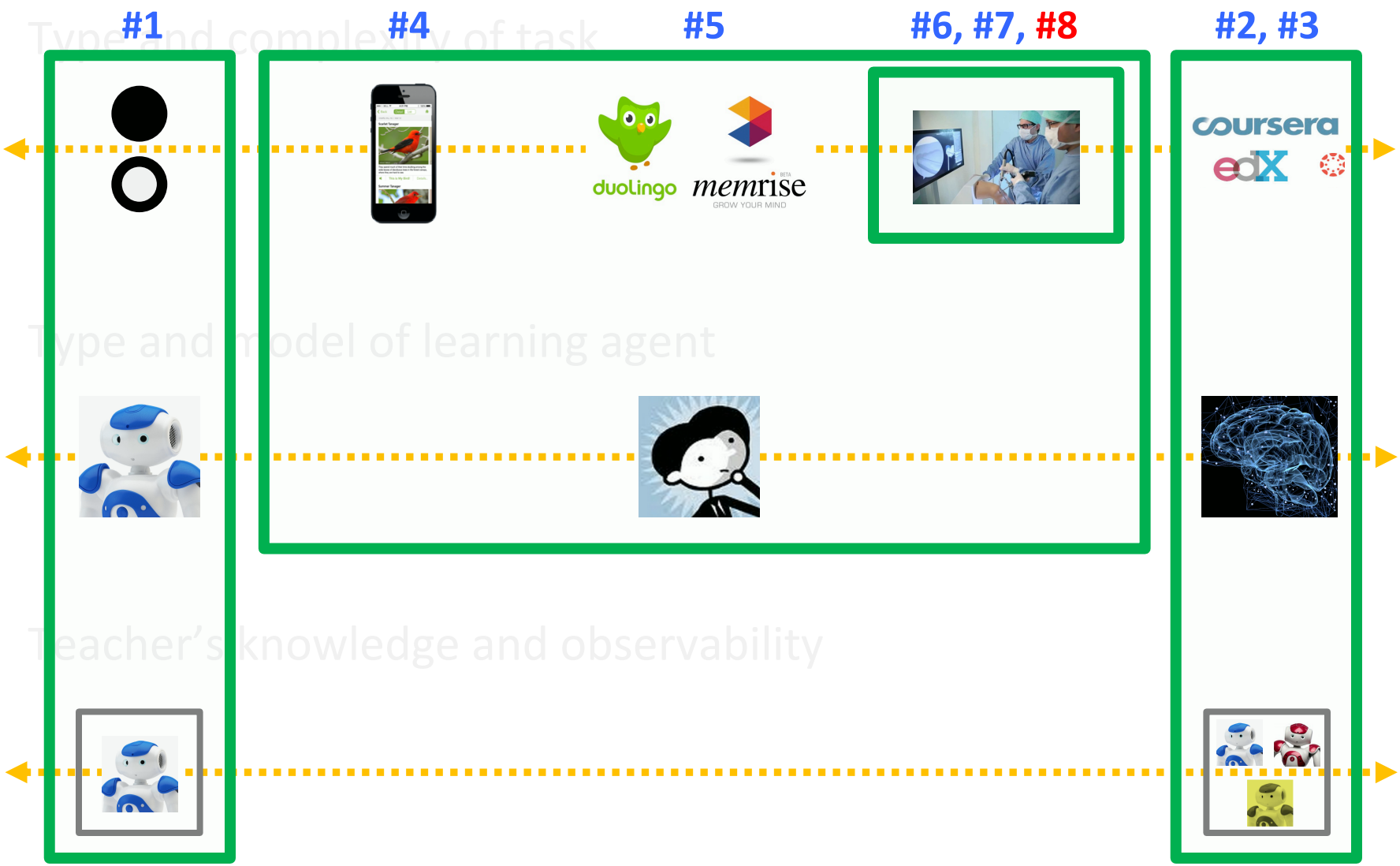
(Survey) An Algorithmic Perspective on Imitation Learning [Osa et al. 2018]

The Role of Teacher: Research Problems

Teaching via demonstrations	Teaching via reward signals
Optimizing curriculum of tasks	**Optimizing curriculum of tasks**
Optimizing sequence of demonstrations	**Providing advice (e.g., correcting errors)** #7
#6 **Accounting for model mismatch**	**Reward shaping (e.g., defining sub-goals)** #8

Machine Teaching: Problem Space

- Type and complexity of task
- Type and model of learning agent
- Teacher's knowledge and observability



Course Logistics

- Webpage
  https://machineteaching.mpi-sws.org/course-mt-w19.html

- Contact
  adishs@mpi-sws.org

- Slides
  https://machineteaching.mpi-sws.org/files/course-mt-w19/machineteaching-day1.pdf
  https://machineteaching.mpi-sws.org/files/course-mt-w19/machineteaching-day2.pdf