Machine Teaching

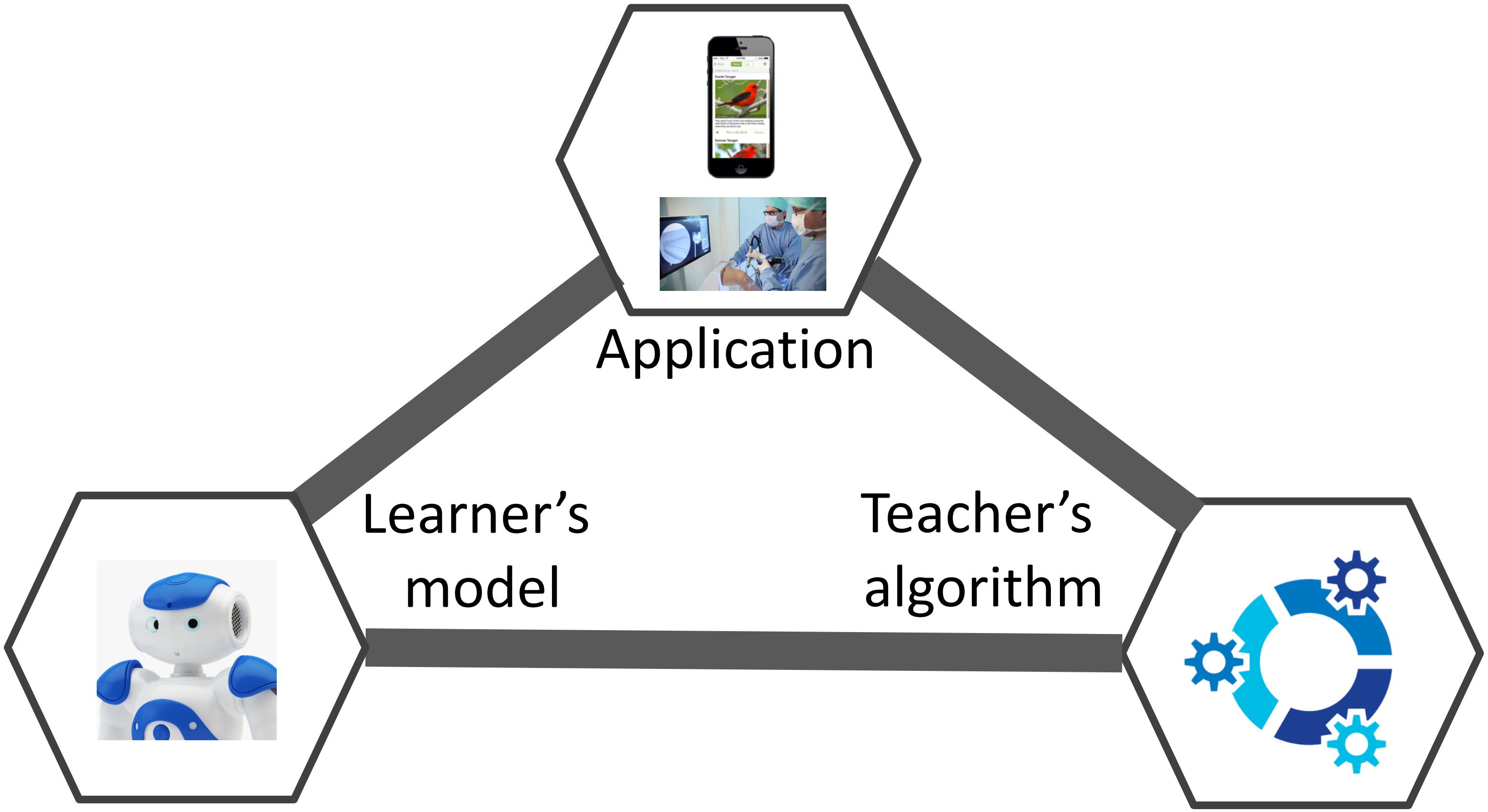
Adish Singla

CMMRS, August 2019

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

MAX-PLANCK-GESELLSCHAFT

Machine Teaching: Key Components

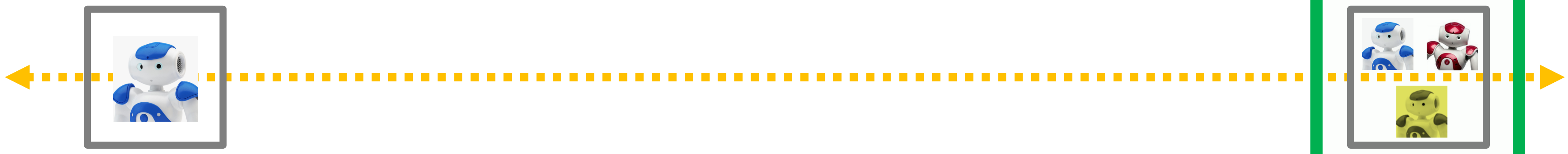


Machine Teaching: Problem Space

- Type and complexity of task

- Type and model of learning agent

- Teacher's knowledge and observability



Cognitive Model of Skill Acquisition

Cognitive tutors

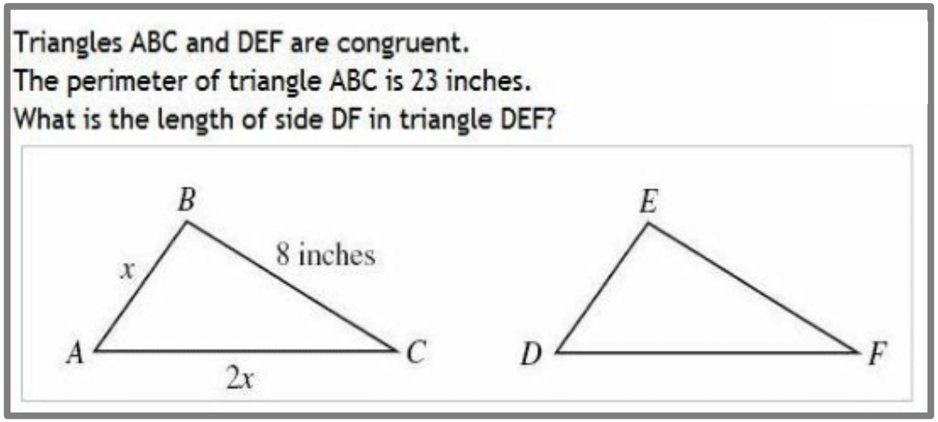
- Used by millions of students for K-12 education
  - https://www.carnegielearning.com/
  - https://new.assistments.org/

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

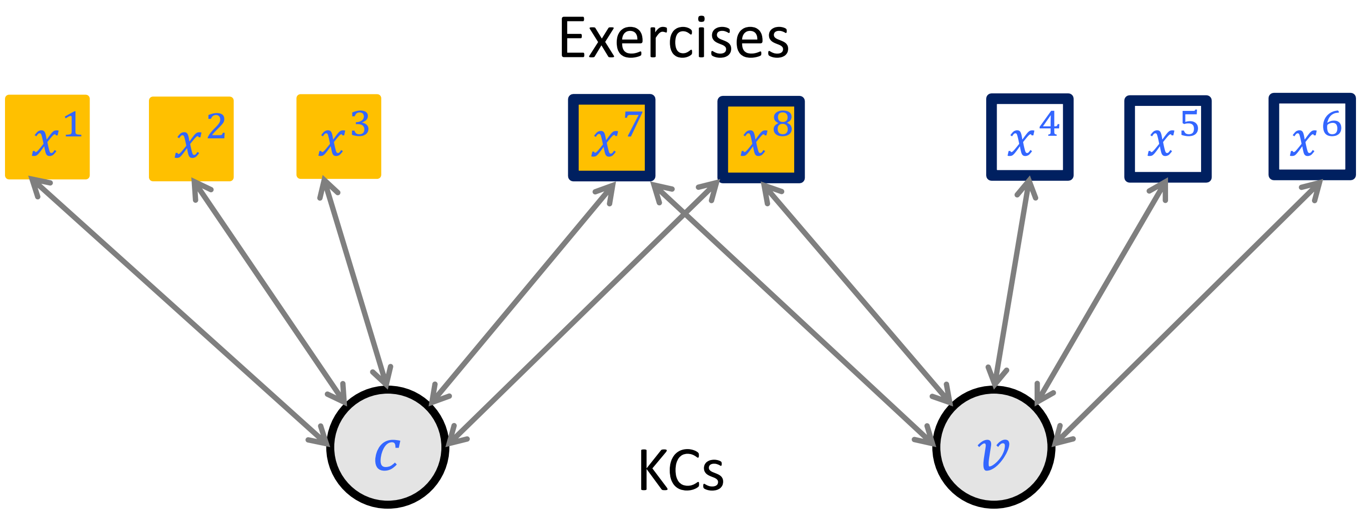
Task: Geometry and Algebra

Knowledge components (KCs) and exercises


Triangles ABC and DEF are congruent.
The perimeter of triangle ABC is 23 inches.
What is the length of side DF in triangle DEF?

$k = c$: 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* $P_{\text{init}}^{k} := P(h_0^k = 1)$

Learner's response (two parameters per KC)

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

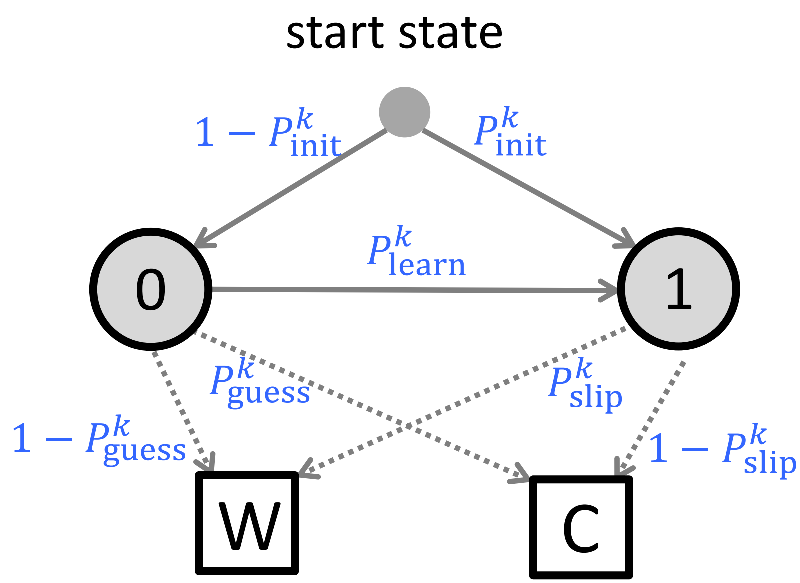
Learner's update (one parameter per KC)

- Conditional probability of *learning* $P_{\text{learn}}^{k} := P\left(h_t^k = 1 \mid h_{t-1}^k = 0\right)$

BKT Learner Model: HMM Representation

Hidden Markov Model (HMM) for a single KC k

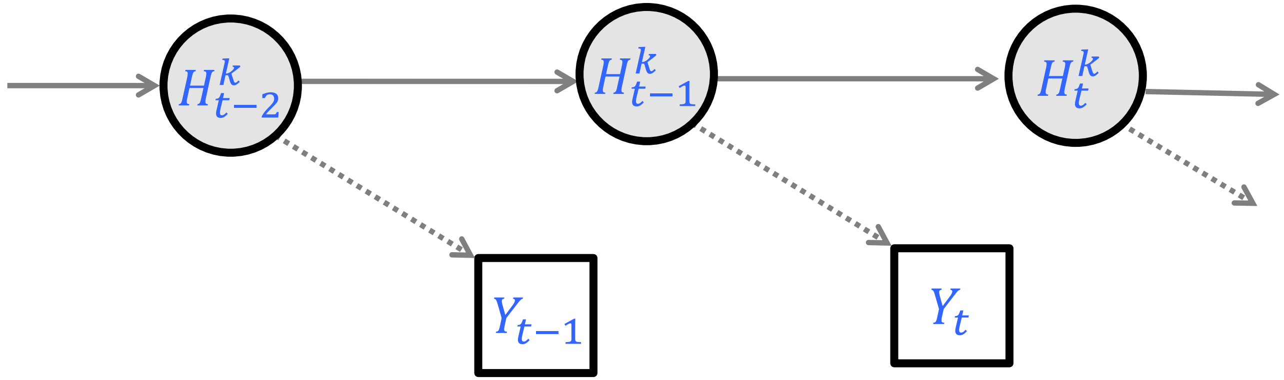
h^k knowledge states (unobserved)



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

BKT Learner Model: DBN Representation

Dynamic Bayesian Network (DBN) for a single KC k



$P(H_0^k = 1)$

P_{init}^k

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

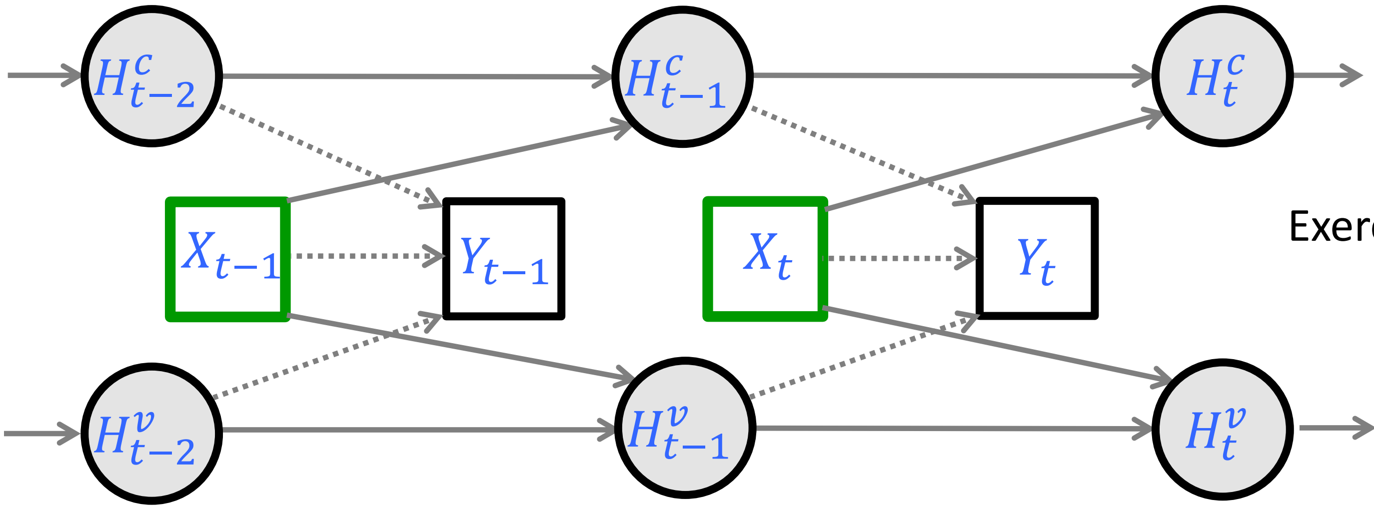
$H_{t-1}^k = 0$	P_{guess}^k
$H_{t-1}^k = 1$	$1 - P_{\text{slip}}^k$

$P\left(H_t^k = 1 \mid H_{t-1}^k\right)$

$H_{t-1}^k = 0$	P_{learn}^k
$H_{t-1}^k = 1$	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

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

Derivation:

$$\begin{aligned}
P\left(Y_t^k = 1 \mid D_{t-1}\right) = \; & P\left(Y_t^k = 1, H_{t-1}^k = 1 \mid D_{t-1}\right) + P\left(Y_t^k = 1, H_{t-1}^k = 0 \mid D_{t-1}\right) \\
= \; & P\left(Y_t^k = 1 \mid H_{t-1}^k = 1, D_{t-1}\right) \cdot P\left(H_{t-1}^k = 1 \mid D_{t-1}\right) \\
& + P\left(Y_t^k = 1 \mid H_{t-1}^k = 0, D_{t-1}\right) \cdot P\left(H_{t-1}^k = 0 \mid D_{t-1}\right) \\
= \; & P\left(Y_t^k = 1 \mid H_{t-1}^k = 1\right) \cdot P\left(H_{t-1}^k = 1 \mid D_{t-1}\right) \\
& + P\left(Y_t^k = 1 \mid H_{t-1}^k = 0\right) \cdot P\left(H_{t-1}^k = 0 \mid D_{t-1}\right) \\
= \; & (1 - P_{\text{slip}}^k) \cdot \theta_{t-1}^k + P_{\text{guess}}^k \cdot (1 - \theta_{t-1}^k)
\end{aligned}$$

BKT Teacher

Inferring learner's knowledge

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

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

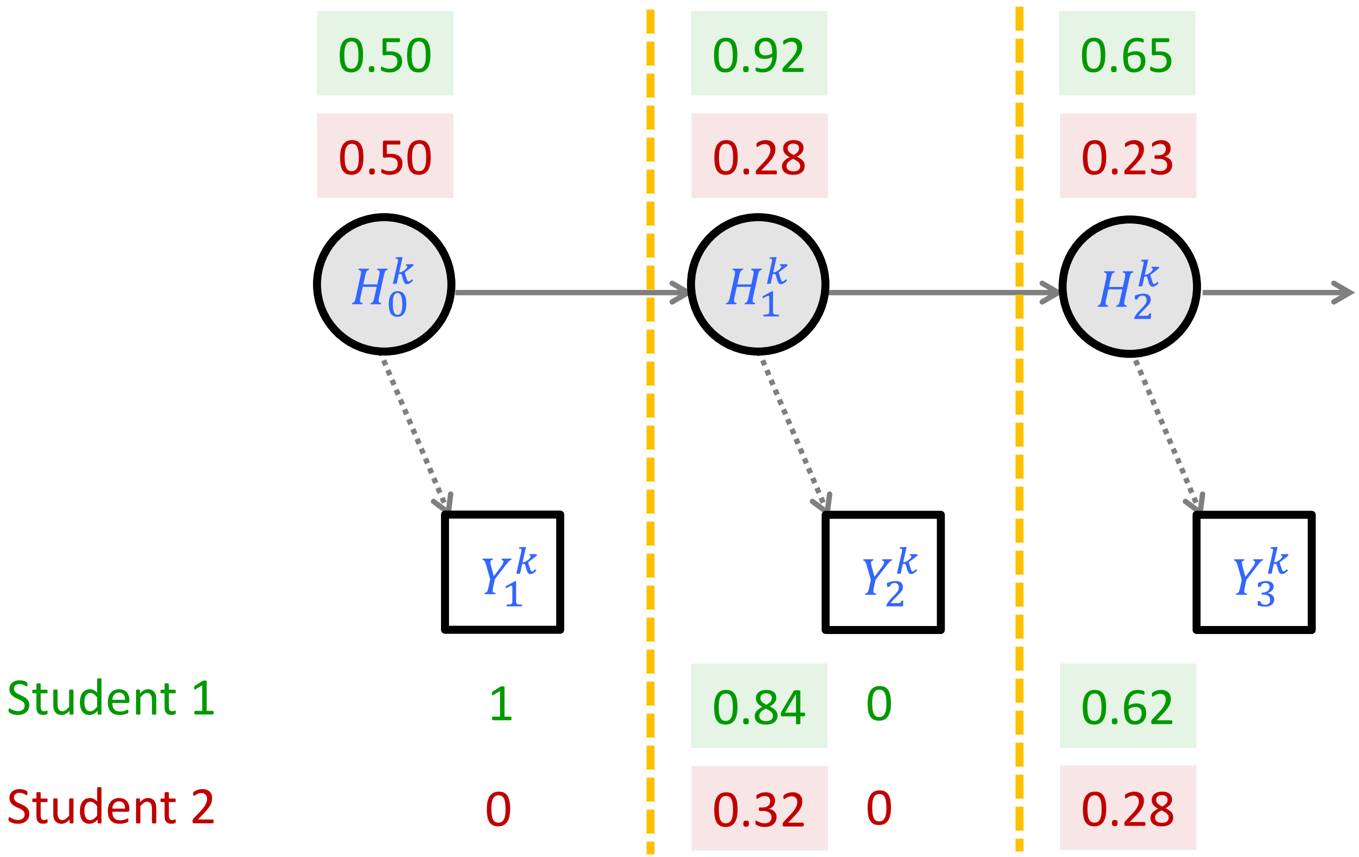
Computing $\hat{\theta}_{t-1}^k$ by applying Bayes rule

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

BKT Teacher

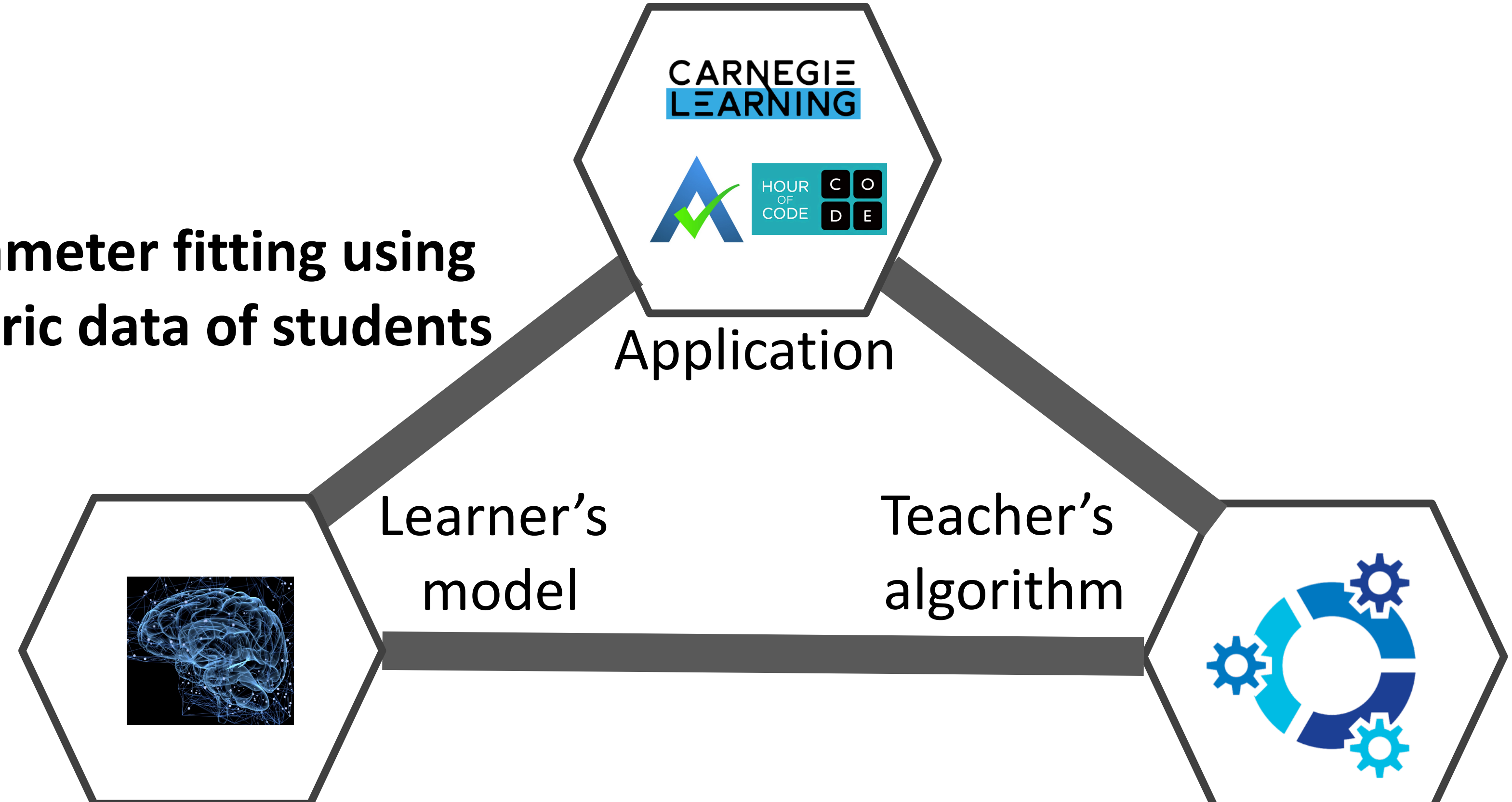
An example of prediction and inference

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



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

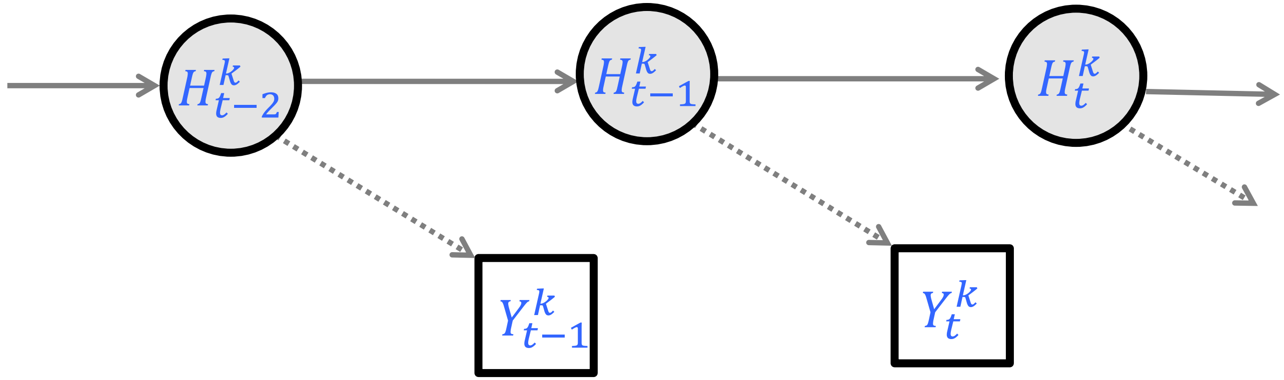
- Forgetting
- Individualization per student
- Skill discovery
  - exercises to skills mapping
  - Inter-skill similarity and prerequisite structure

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



$P(H_0^k = 1)$

P_{init}^k

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

$H_{t-1}^k = 0$	P_{guess}^k
$H_{t-1}^k = 1$	$1 - P_{\text{slip}}^k$

$P\left(H_t^k = 1 \mid H_{t-1}^k\right)$

$H_{t-1}^k = 0$	P_{learn}^k
$H_{t-1}^k = 1$	$1 - P_{\text{forget}}^k$

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)

$\textbf{BKT}_1$	$\textbf{BKT}_2$	**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 $\not\Rightarrow$ 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

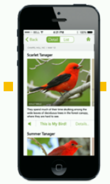
- Type and complexity of task

- Type and model of learning agent

- Teacher’s knowledge and observability

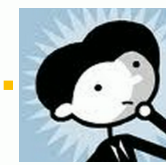
Machine Teaching: Problem Space

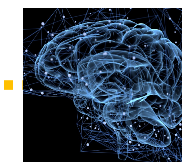
- Type and complexity of task



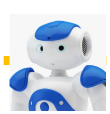
[ICML ’14]
[AAAI ’19]

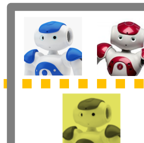
- Type and model of learning agent



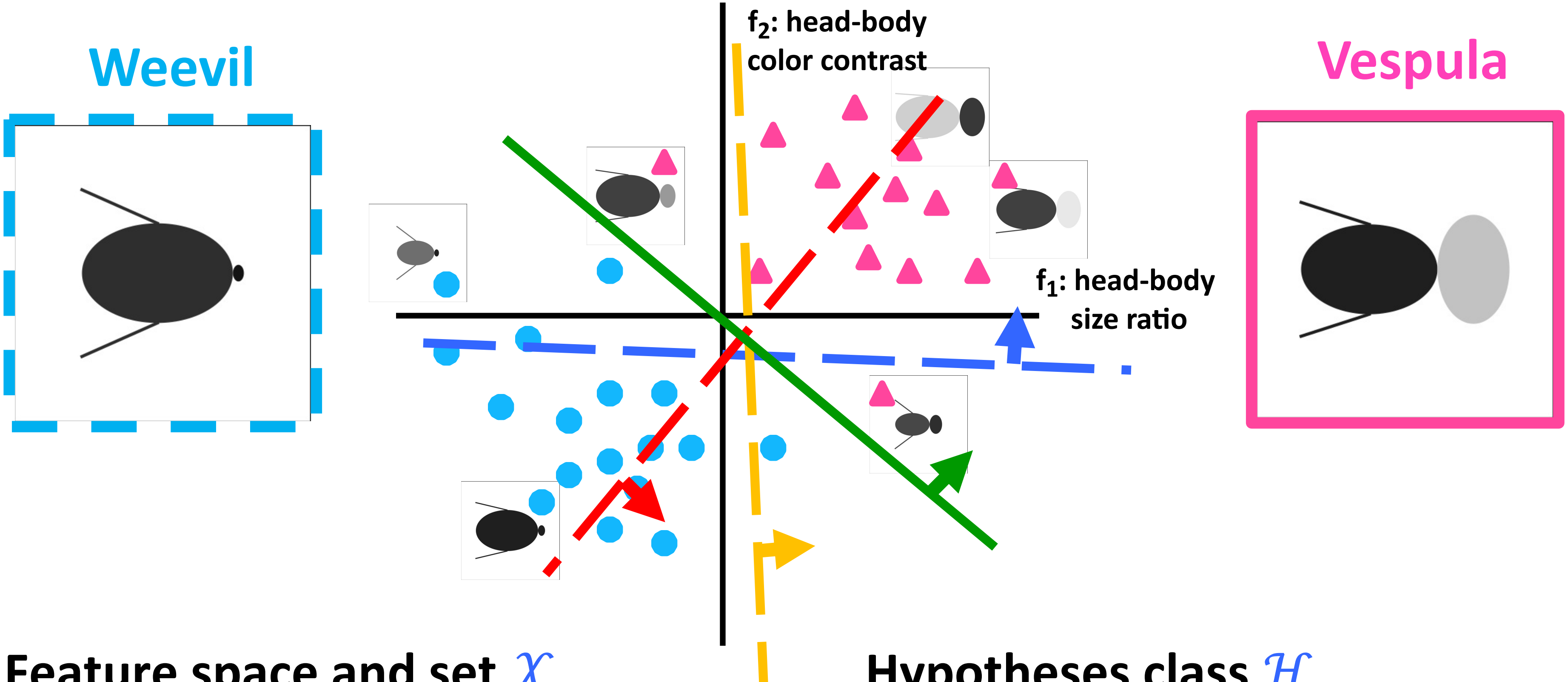


- Teacher’s knowledge and observability





Setup: Weevil and Vespula (WV)



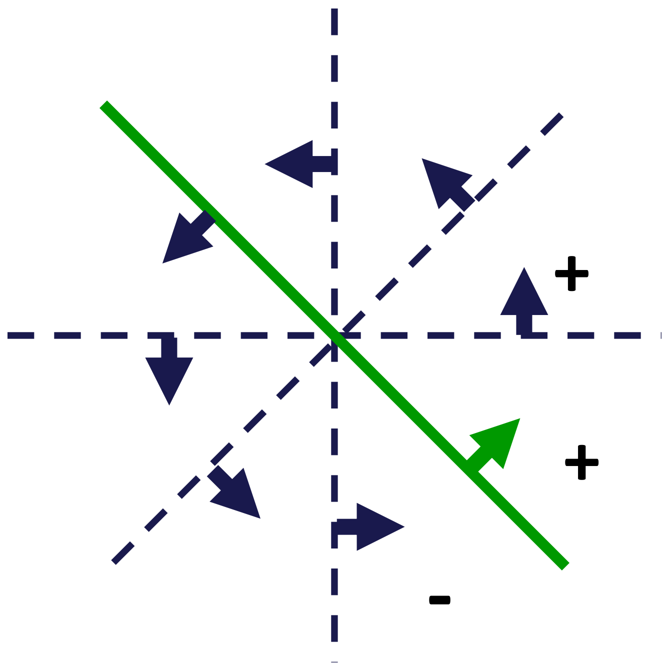
Feature space and set $\mathcal{X}$

- f$_1$: head-body size ratio
- f$_2$: head-body color contrast

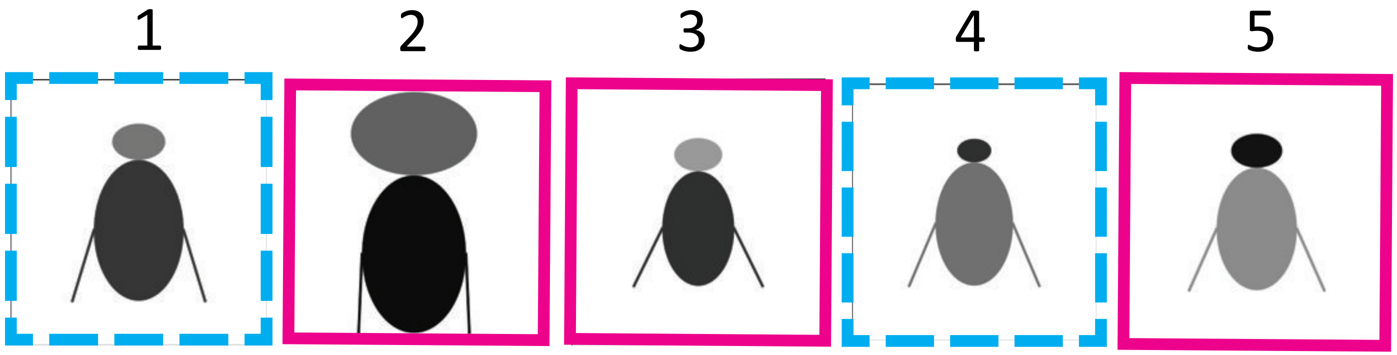
Hypotheses class $\mathcal{H}$

- **Green**: target hypothesis h^*
- **Blue**: ignoring feature f$_1$
- **Yellow**: ignoring feature f$_2$
- **Red**: wrongly using feature f$_2$

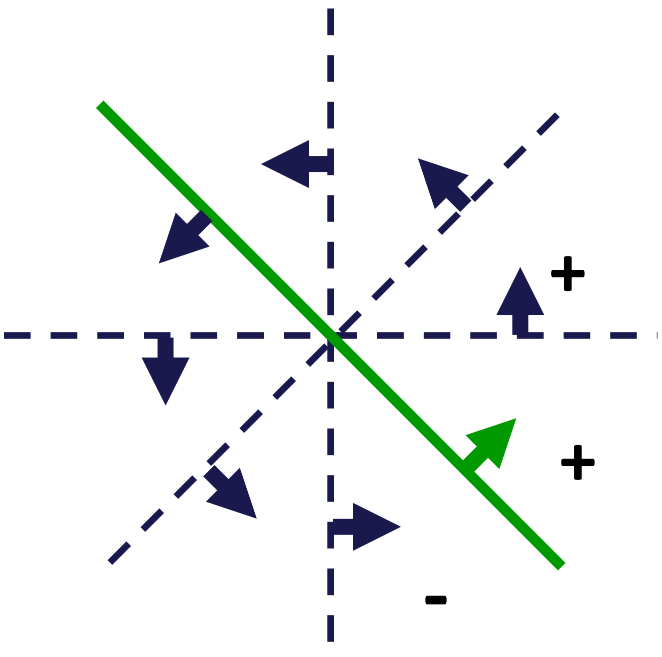
Learner: Classical Model



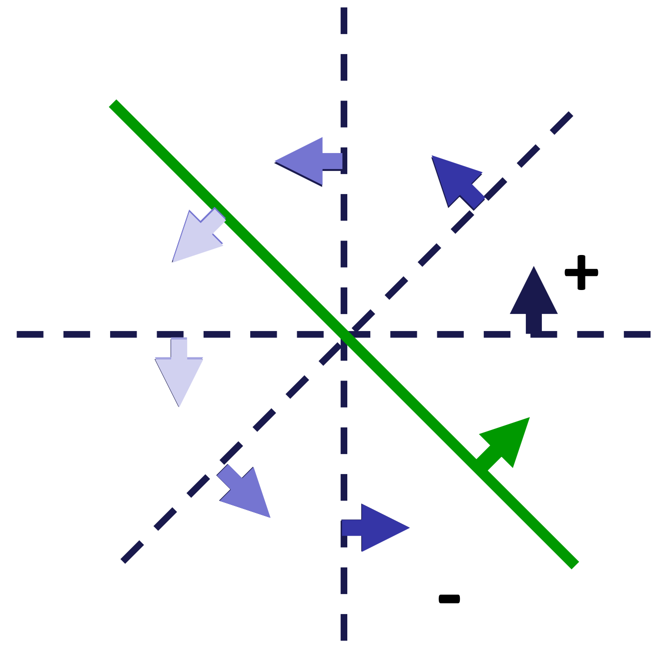
- Classical model [Goldman, Kearns '95]
    - Hypotheses eliminated upon inconsistency
- Optimal teaching sequence := Set Cover
- Picks "difficult" (confusing?) examples



Learner: Our Robust Model



Classical “**noise-free**” model:
Hypotheses **eliminated**
upon inconsistency



Our “**robust**” model:
Hypotheses **less likely**
upon inconsistency

Learner: Our Robust Model

Hypotheses class $\mathcal{H}$

- Set of functions $h : \mathcal{X} \rightarrow \mathbb{R}$
- Label assigned by h is $\mathrm{sgn}\big(h(x)\big)$

Learner's update

- Given labeled examples $(x_\tau, y_\tau)_{\tau=1,2,\ldots,t}$, learner update weights as

$$P_t(h) \propto P_0(h) \prod_{\substack{\tau=1,2,\ldots,t \\ y_\tau \neq \mathrm{sgn}(h(x_\tau))}} l(y_\tau; h, x_\tau)$$

Inconsistent examples

likelihood function

- Learner selects a new hypothesis as $h_t \sim P_t(h)$

Learner: Our Robust Model

Example of a likelihood function

- Given a labeled example (x, y), define

$$l(y; h, x) = \frac{1}{1+\exp(-\alpha \cdot y \cdot h(x))}$$

  where α is a scaling factor
- $\alpha \rightarrow \infty$ reduces to elimination of inconsistent hypotheses

Teacher: Optimization Problem

Expected error

- Let $\overleftrightarrow{S}$ be a sequence of examples shown; the expected error rate is

$$\mathbb{E}[\mathrm{err} \mid \overleftrightarrow{S}] = \sum_{h \in \mathcal{H}} P\left(h \mid \overleftrightarrow{S}\right) \cdot \mathrm{err}\,(h, h^*)$$

Distribution over learner's h after showing examples $\overleftrightarrow{S}$

Fraction of examples where h and h^* disagree

Optimization problem

- Find smallest sequence of examples to achieve a desired error rate

$$\overleftrightarrow{S}^{\mathrm{opt}} = \underset{\overleftrightarrow{S}}{\mathrm{argmin}}\, |\overleftrightarrow{S}| \text{ s.t. } \mathbb{E}[\mathrm{err} \mid \overleftrightarrow{S}] \le \epsilon$$

Teacher: Optimization Problem

- Step 0: Expected error rate is a set function: $\mathbb{E}[\text{err} \mid \overleftrightarrow{S}] = \mathbb{E}[\text{err} \mid S]$
- Step 1: Maximize reduction in error

$$R(S) = \mathbb{E}[\text{err} \mid \emptyset] \; - \; \mathbb{E}[\text{err} \mid S] = \sum_{h \in \mathcal{H}} \big(P(h \mid \emptyset) - P(h \mid S)\big) \cdot \text{err}\,(h, h^*)$$

Designing submodular surrogate objective

- Step 2: Replace $R(.)$ with a surrogate objective $F(.)$:

$$F(S) = \sum_{h \in \mathcal{H}} \big(Q(h \mid \emptyset) - Q(h \mid S)\big) \cdot \text{err}\,(h, h^*)$$

where $Q(h \mid S)$ is the **unnormalized** posterior

- **Theorem**: $F(.)$ satisfies submodularity. It is sufficient to optimize F to get guarantees on the original teaching problem.

Teacher: Algorithm

Iterative greedy algorithm

- **Input:** $\mathcal{H}, \mathcal{X}, h^*$

  Prior $P_0(\mathcal{H})$, learner model parameter α

  Desired error ϵ
- **Initialize:** set $\mathrm{S} \leftarrow \emptyset$
- While $\mathrm{F(S)} < \mathbb{E}[\mathrm{err}\,|\emptyset] - \epsilon \cdot P_0(h^*)$:
  - Select $x \leftarrow \mathrm{argmax}_{x' \in \mathcal{X}}\, F(x' \cup S) - F(S)$
  - Provide $x, \mathrm{sgn}\big(h^*(x)\big)$ to learner
  - Update $S \leftarrow S \cup \{x\}$

Teacher: Theoretical Guarantees

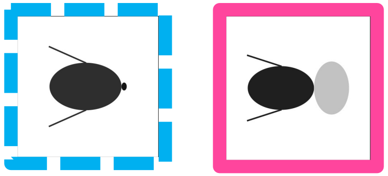
Approximation guarantees for the general case

Theorem: Fix ϵ. Let $z = P_0(h^*)$ be the prior probability of the target hypothesis. The algorithm terminates after at most $O\left(\left|\overleftrightarrow{S}^{\mathrm{opt}}\right| \cdot \log\left(\frac{2}{\epsilon \cdot z}\right)\right)$ examples such that learner's error is less than ϵ.

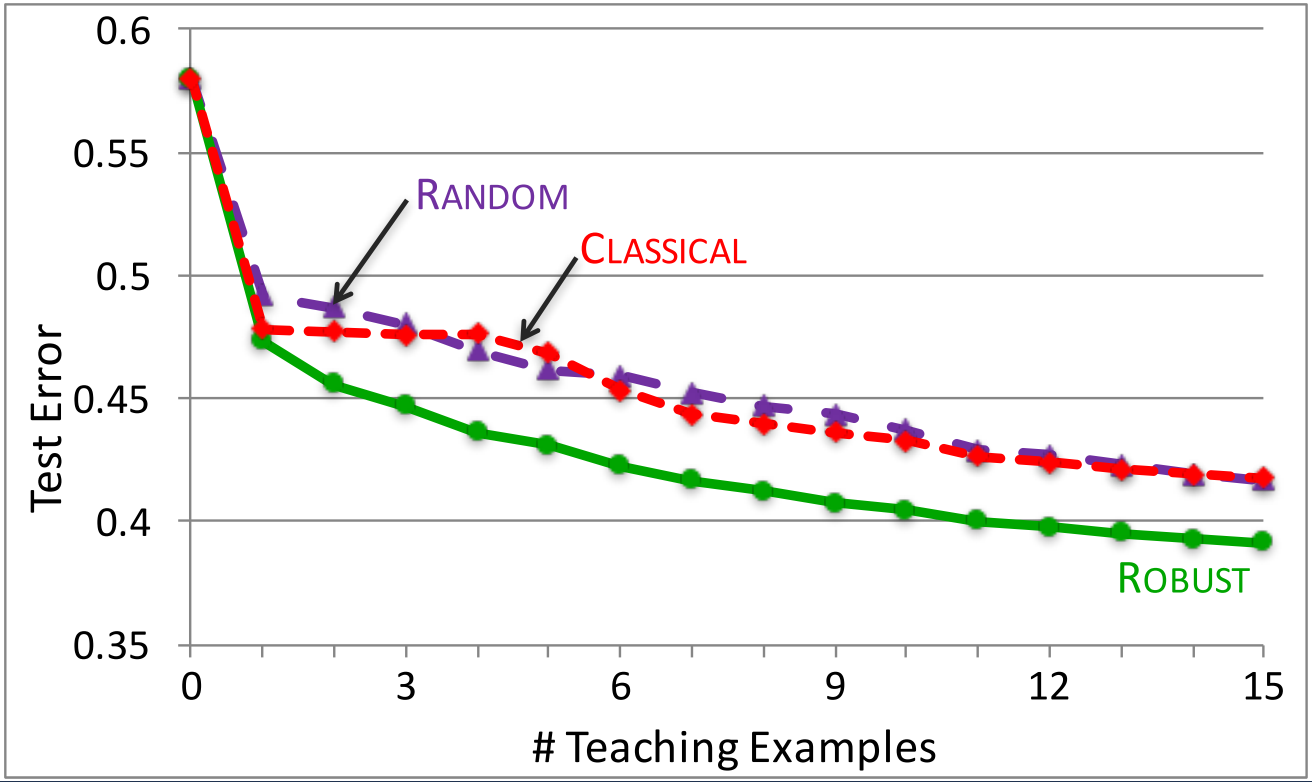
Teaching complexity for linear separators

Theorem: Suppose that the hypotheses are hyperplanes and $\mathcal{X}$ can be synthesized. Then, the teaching algorithm achieves learner's error less than ϵ after at most $O\left(\log^2\left(\frac{2}{\epsilon \cdot z}\right)\right)$ examples.

Results (WV): Simulated Learners



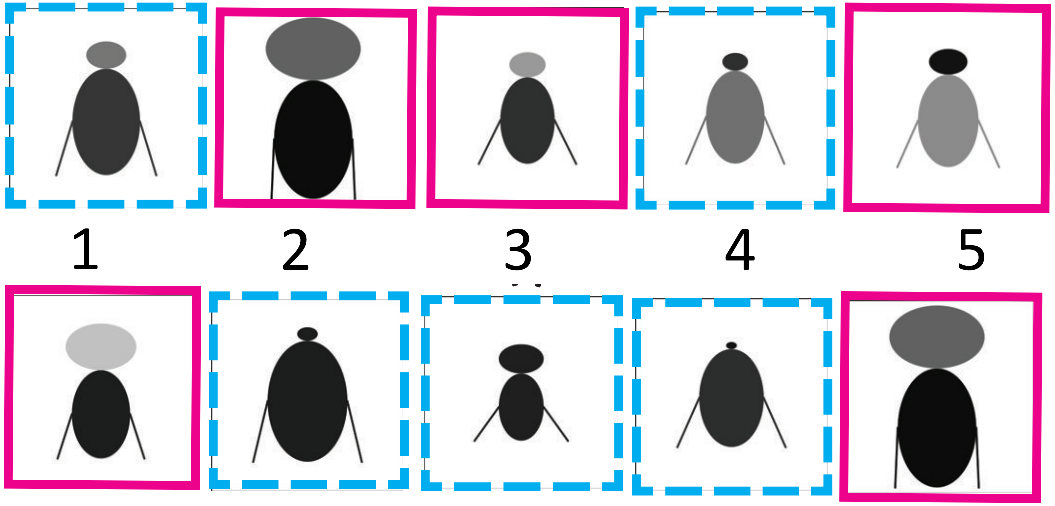
- $|\mathcal{X}| = 100, |\mathcal{H}| = 96$
- 100 simulated learners: varying α
  - Teacher considers a learner's model with $\alpha = 2$
- Test phase with 10 unseen examples

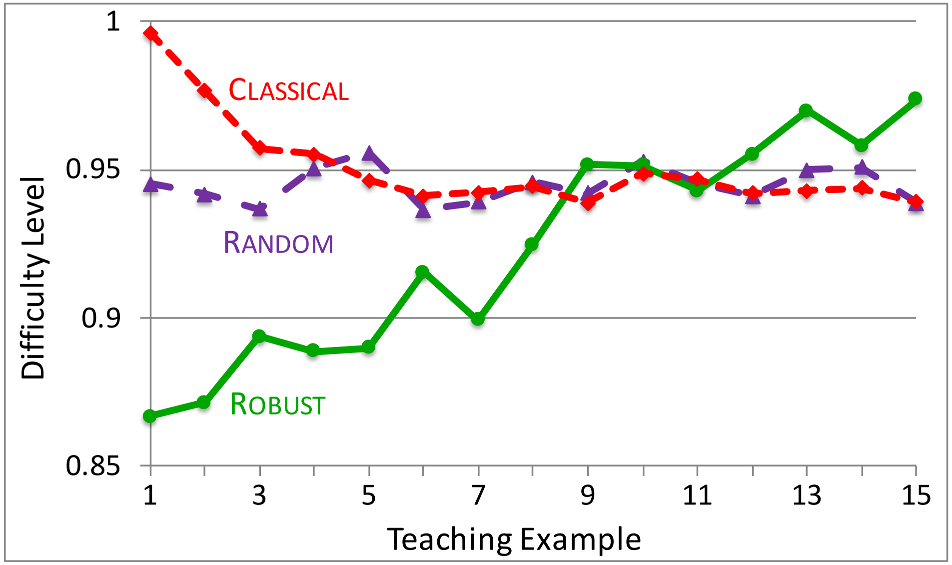


Results (WV): Teaching Curriculum

Classical Model
Hypotheses **eliminated** upon inconsistency

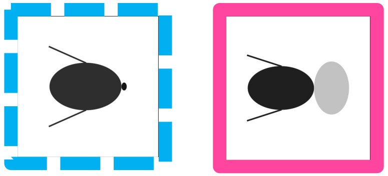
Robust Model
Hypotheses **less likely** upon inconsistency

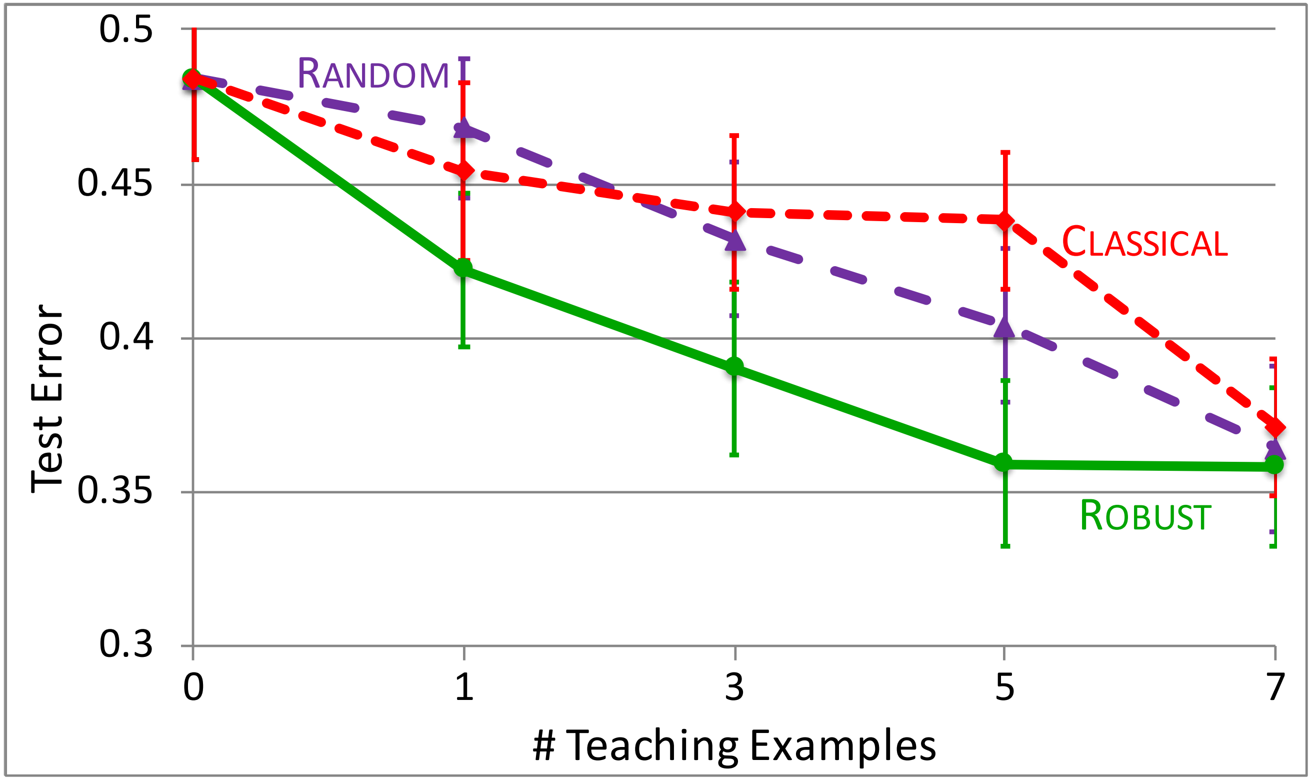




Results (WV): Human Learners

- 780 participants from a crowdsourcing platform
  - 60 per control group: (algorithm, length)
- Test phase with 10 unseen images





Setup: Endangered Woodpeckers (WP)

Least concerned

Endangered

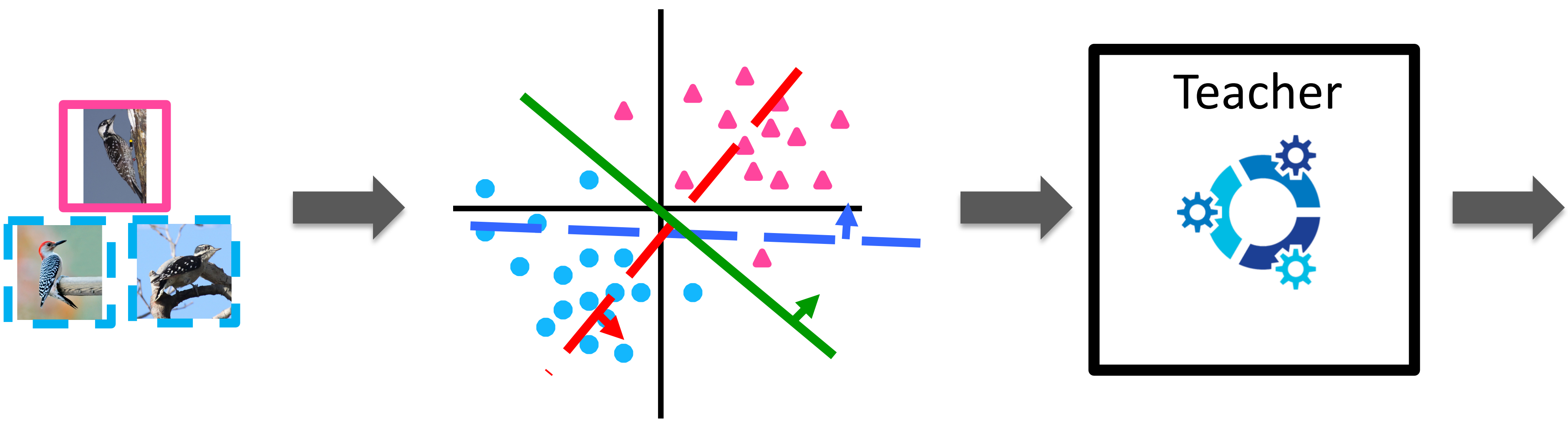
Downy WP

Red-bellied WP

Red-cockaded WP

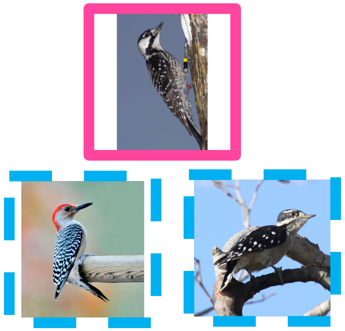
Setup: Endangered Woodpeckers (WP)

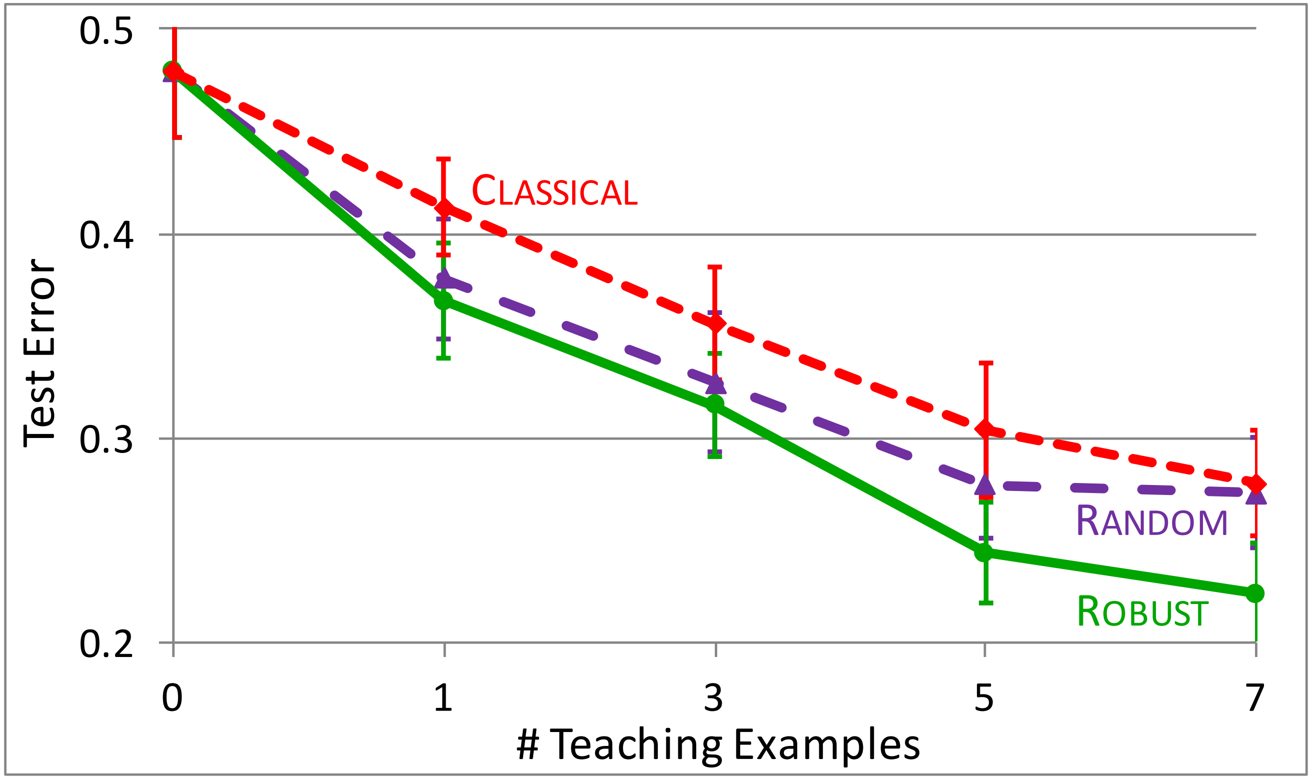
- What is suitable $\mathcal{X}$ and $\mathcal{H}$?
- Crowd-embedding [Wellinder et al. NIPS'10]
  - Set of $|\mathcal{X}| = 100$ images from [CUB-200 dataset]
  - Low dimensional embedding using human annotation data



Results (WP): Human Learners

- 520 participants from a crowdsourcing platform
  - 40 per control group: (algorithm, length)
- Test phase with 15 unseen images





Towards Large-scale Multiclass

- Richer interpretable teaching signals
- Adaptive models of teaching
- Limited memory

Machine Teaching: Problem Space