Enhancing Human Learning

using Stochastic Optimal Control and RL

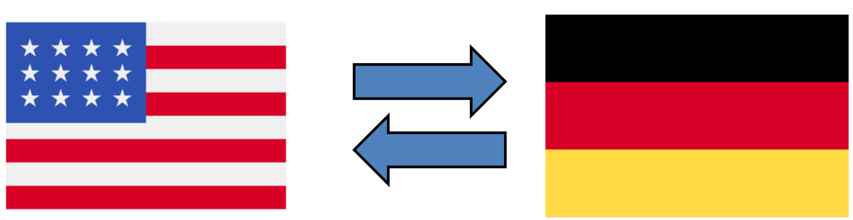
HUMAN-CENTERED MACHINE LEARNING

http://courses.mpi-sws.org/hcml-ws18/

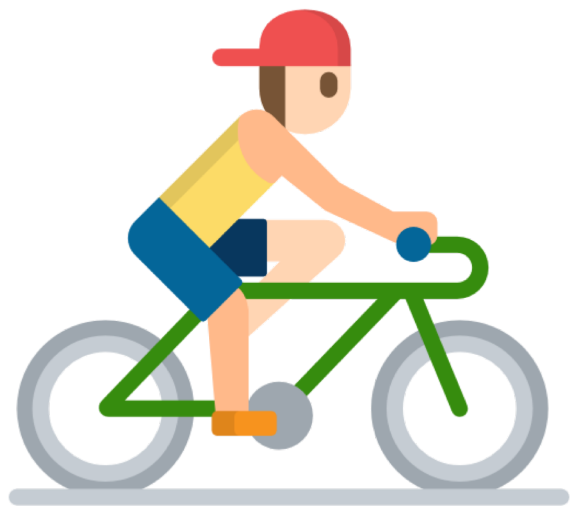
MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

What is *learning?*

➢ Declarative versus Procedural learning



Learning a new language's vocabulary



Learning how to ride a bike

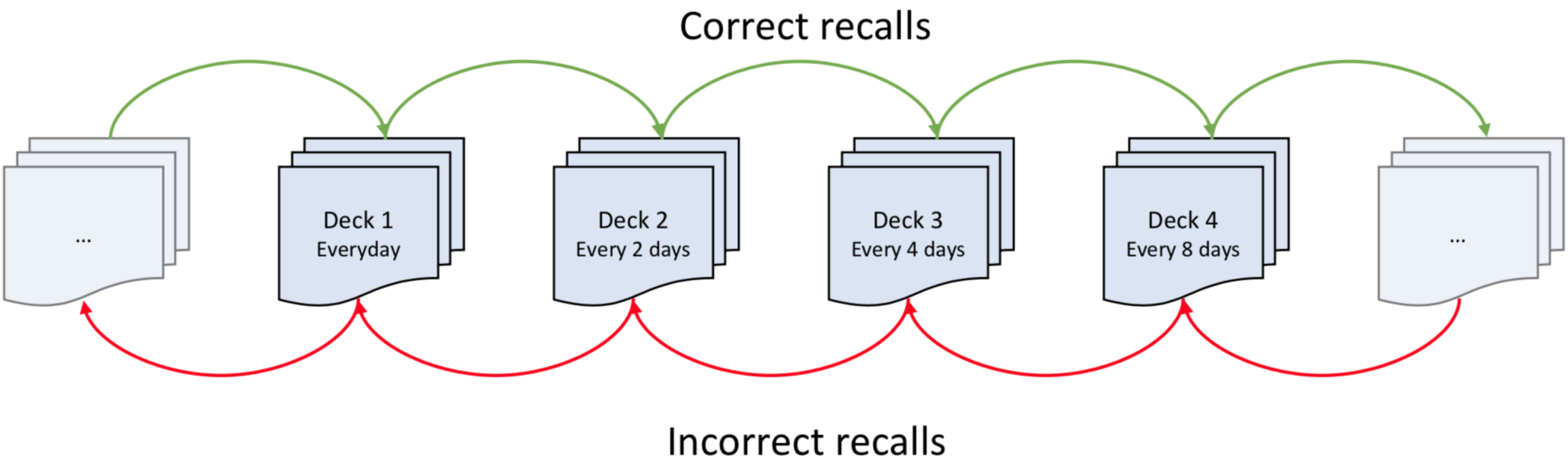
How do humans *learn?*

- Declarative versus Procedural learning
- Repetition is important!

How do humans *learn?*

- Declarative versus Procedural learning
- **Spaced** Repetition is important!

[Ebbinghaus 1885]



Leitner System for Flash cards

[Leitner 1974]

How do humans *learn* in today's world?

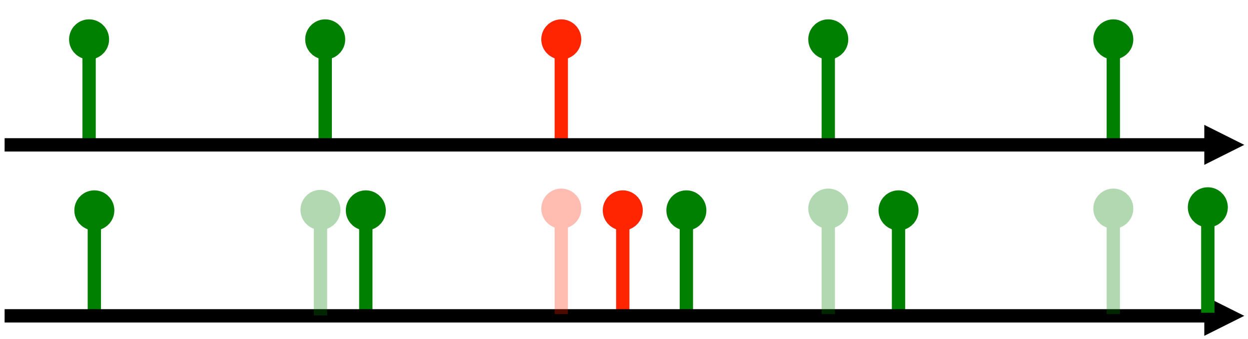
➢ Computer assisted learning:

duolingo Anki synap

The platforms decide *when to schedule* reviews based on the user's history and item.

Uniform

Or?



Strategy to optimize spaced repetition

Data representation	**Student feedback and memory dynamics**	**Optimizing recall**	**Experiments**
Temporal point processes	System of stochastic equations with jumps	Optimal control of jumps	Duolingo

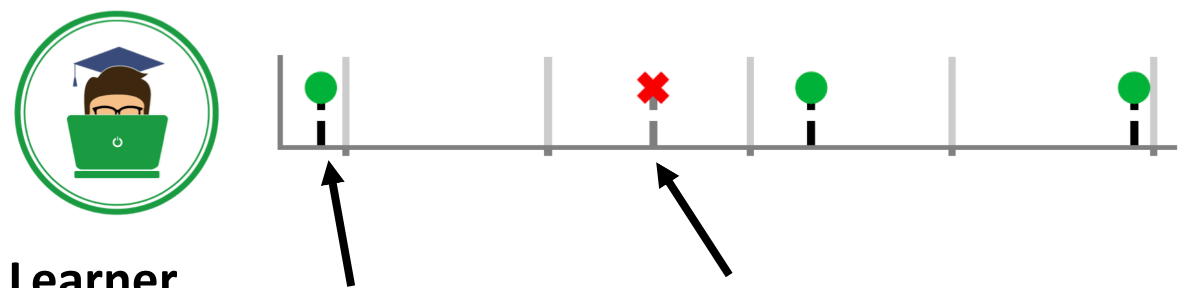
Optimizing spacing between repetition

Agent

duolingo

Online learning platform

Environment



Learner

Review & successful recall

Review & unsuccessful recall

When to review to maximize recall probability?

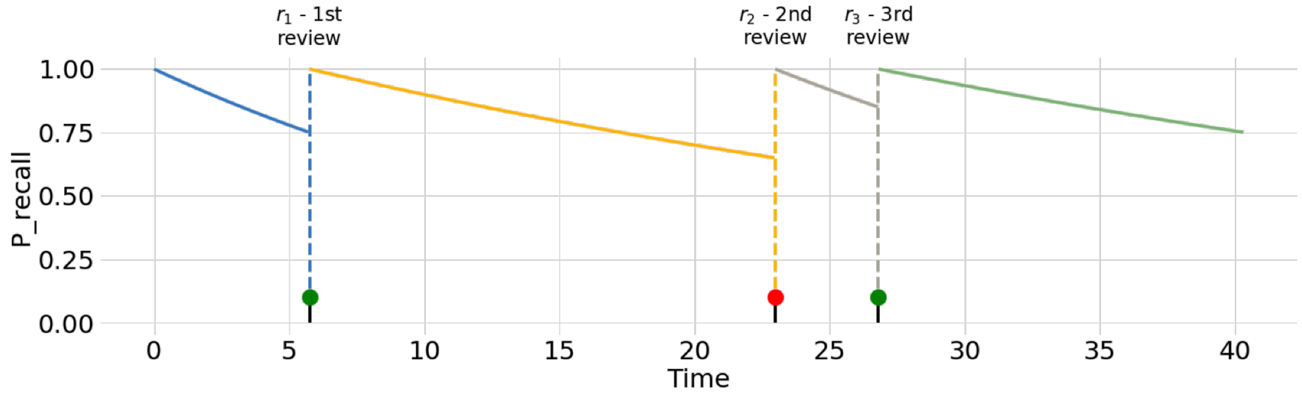
$\lambda_i(t) \longrightarrow N_i(t)$

Design (optimal) reviewing intensities

Marks

Memory Model: Intuition

- Memory strength decays with time
- Resets to *maximum* immediately after review
- Recall is probabilistic



Memory Model: A Mathematical Model

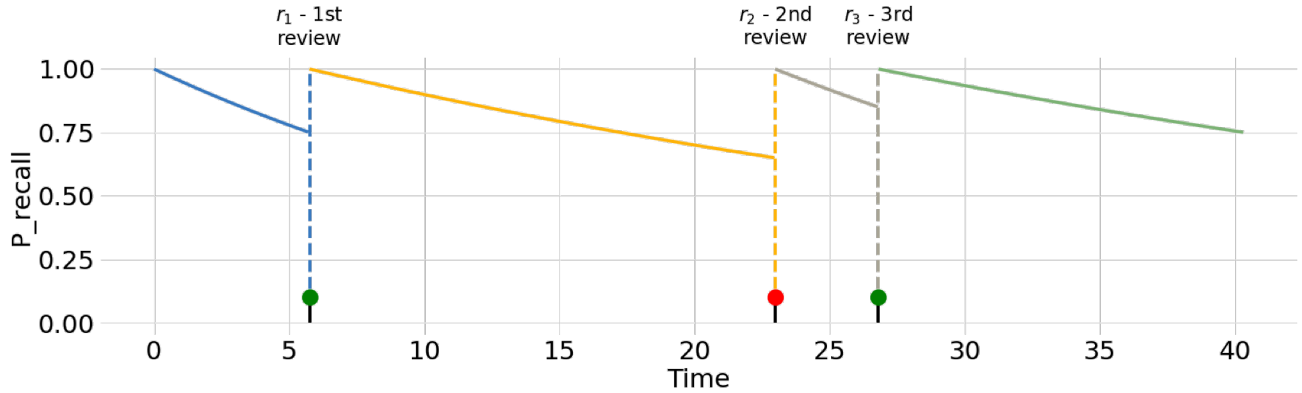
$m(t)$: Probability of recall

$n(t)$: Memory decay rate.

η : Time since last review.

$$m(t) = e^{-n(t)\times\eta}$$

$$n(t) = \begin{cases} (1-\alpha)\times n(t^-) & \text{if recalled} \\ (1+\beta)\times n(t^-) & \text{if forgotten} \end{cases}$$



Memory Model: SDE with jumps

$m(t)$: Probability of recall

$n(t)$: Memory decay rate.

η : Time since last review.

$$m(t) = e^{-n(t) \times \eta}$$

$$n(t) = \begin{cases} (1-\alpha) \times n(t^-) & \text{if recalled} \\ (1+\beta) \times n(t^-) & \text{if forgotten} \end{cases}$$

$$dm(t) = -m(t)n(t)dt + (1 - m(t))dN(t)$$

$$dn(t) = [-\alpha r(t)n(t) + \beta\big(1 - r(t)\big)n(t)]dN(t)$$

Memory Model: Inferring parameters

$m(t)$: Probability of recall

$n(t)$: Memory decay rate.

η : Time since last review.

$$m(t) = e^{-n(t) \times \eta}$$

$$n(t) = \begin{cases} (1-\alpha) \times n(t^-) & \text{if recalled} \\ (1+\beta) \times n(t^-) & \text{if forgotten} \end{cases}$$

$$dm(t) = -m(t)n(t)dt + (1 - m(t))dN(t)$$

$$dn(t) = [-\alpha r(t)n(t) + \beta\big(1 - r(t)\big)n(t)]dN(t)$$

We can estimate parameters α and β from data.

[Settles et al. 2016]

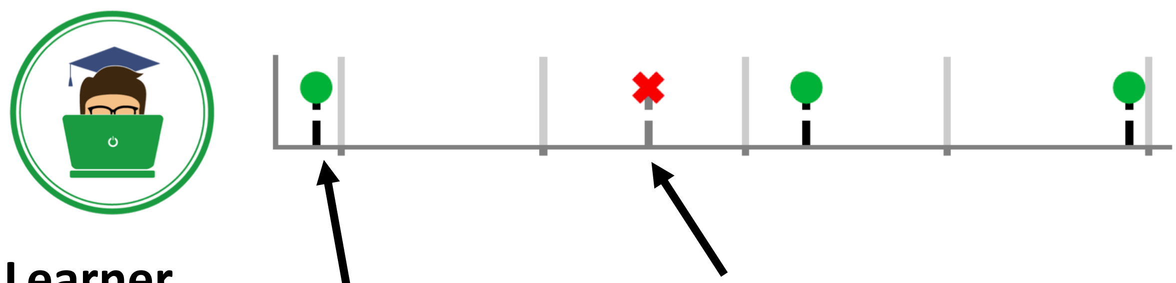
Optimizing spaced repetition: The Scheduler

Agent

duolingo

Online learning platform

Environment



Learner

Review & successful recall

Review & unsuccessful recall

When to review to maximize recall probability?

$$\underbrace{\text{When to review}}_{\lambda_i(t) \longrightarrow N_i(t)} \text{ to maximize } \underbrace{\text{recall probability}}_{\text{Marks}}\text{?}$$

Design (optimal) reviewing intensities

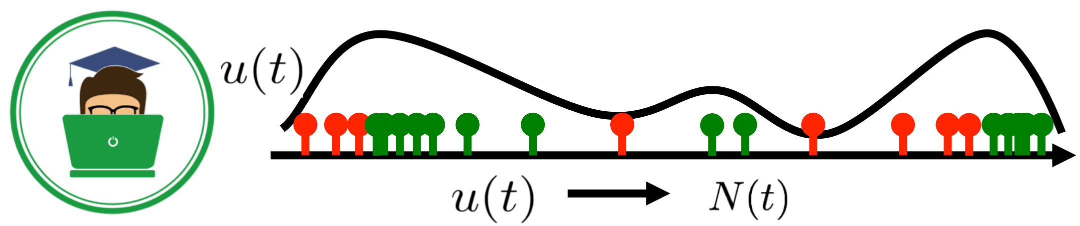
Representing actions of the teacher as MTPPs

Agent

duolingo

Online learning platform

Environment



Learner

- We will control the *rate* of reviewing $u(t)$
- For simplicity, we will consider the problem for just **one item**.

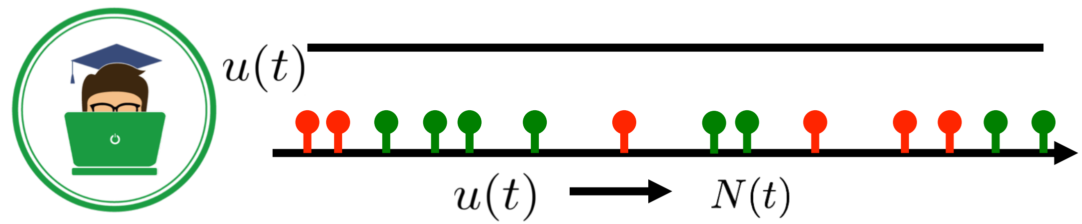
Spaced repetition: Uniform baseline

Agent

Online learning platform

$$u(t) = \mu$$

Environment



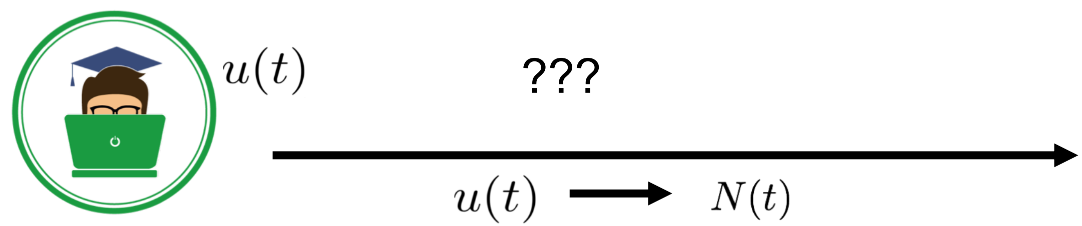
Learner

Does not exploit spacing effect at all.

Spaced repetition: Threshold Heuristic

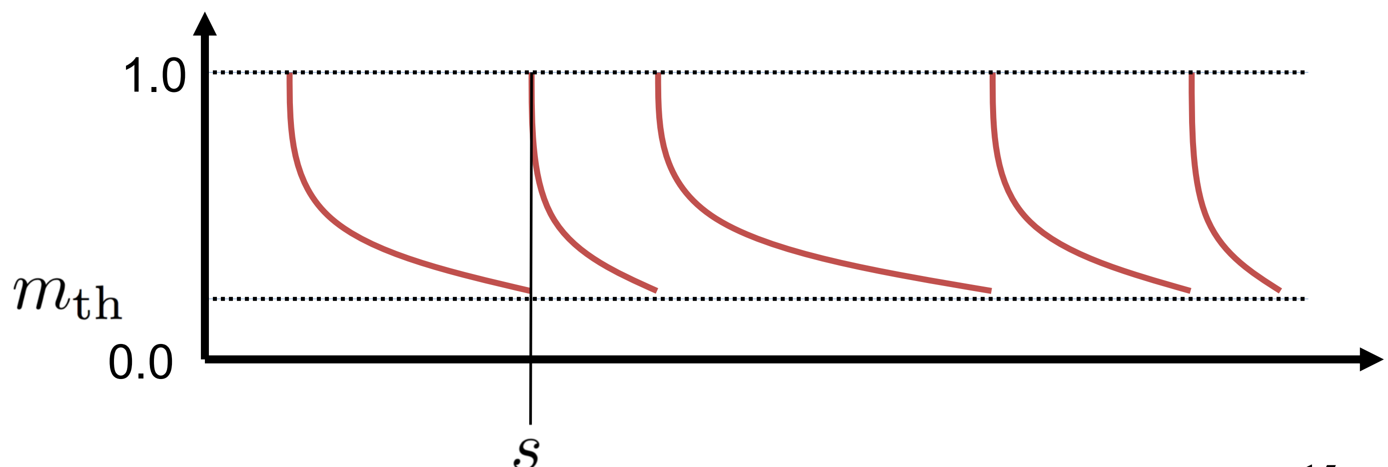
Agent

Environment



Learner

Online learning platform



"desirable difficulty"

[Bjork, 1994]

$$m_{\text{th}} = m(s)$$

Spaced repetition: Threshold Heuristic

Agent

duolingo

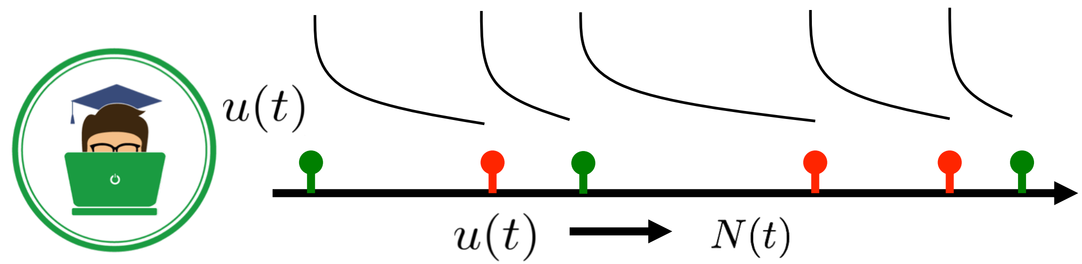
Online learning platform

"desirable difficulty"

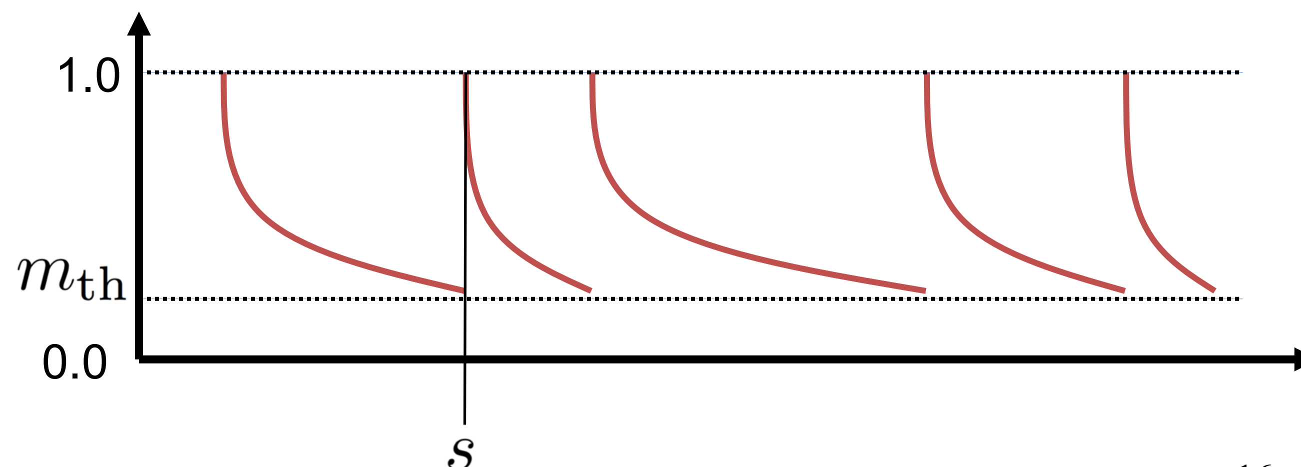
[Bjork, 1994]

$$m_{\text{th}} = m(s)$$

Environment



Learner



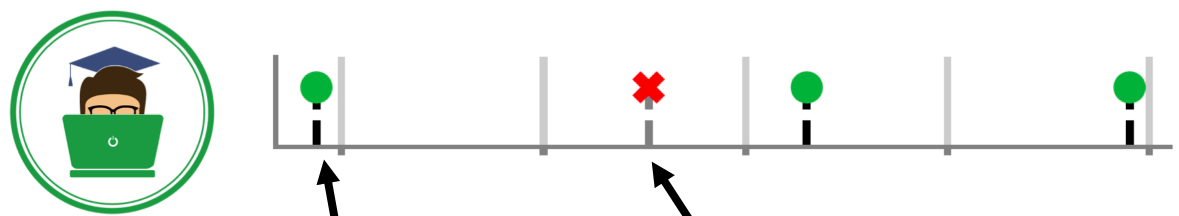
Optimizing spacing between repetition

Agent

duolingo

Online learning platform

Environment



Learner

Review & successful recall

Review & unsuccessful recall

When to review to maximize recall probability?

$\lambda_i(t) \longrightarrow N_i(t)$

Design (optimal) reviewing intensities

Marks

Optimization Objective

Objective trades off high recall and high reviewing rate.

$$\underset{u(t_0,t_f]}{\text{minimize}} \quad \mathbb{E}_{(N,r)(t_0,t_f]}\left[\int_{t_0}^{t_f}\left(\underbrace{\frac{1}{2}(1-m(\tau))^2}_{\text{high recall}}+\underbrace{\frac{1}{2}qu^2(\tau)}_{\text{high reviewing rate}}\right)d\tau\right]$$

$$\text{subject to} \quad u(t) \geq 0 \ \forall t \in (t_0, t_f)$$

Dynamics defined by Jump SDEs

$$\begin{cases} dm(t) = -m(t)n(t)dt + (1-m(t))dN(t) \\ dn(t) = [-\alpha r(t)n(t) + \beta\big(1-r(t)\big)n(t)]dN(t) \end{cases}$$

Stochastic Optimal Control: Cost-to-go

$$\underset{u(t_0,t_f]}{\text{minimize}} \;\; \mathbb{E}_{(N,r)(t_0,t_f]}\left[\int_{t_0}^{t_f}\left(\frac{1}{2}(1-m(\tau))^2+\frac{1}{2}qu^2(\tau)\right)d\tau\right]$$

$$\text{subject to} \;\; u(t)\geq 0 \;\forall t\in(t_0,t_f)$$

Dynamics defined by Jump SDEs

$$\begin{cases} dm(t) = -m(t)n(t)dt + (1-m(t))dN(t) \\ dn(t) = [-\alpha r(t)n(t) + \beta\big(1-r(t)\big)n(t)]dN(t) \end{cases}$$

$$J(n(t),m(t),t) \;\; = \min_{u(t,t_f]} \mathbb{E}_{(N(s),r(s))|_{s=t}^{s=t_f}}\left[\phi(m(t_f),n(t_f)) \;+\int_t^{t_f}\ell(m(\tau),u(\tau))d\tau\right].$$

$$\ell(m(t),n(t),u(t)) = \frac{1}{2}(1-m(t))^2+\frac{1}{2}qu^2(t),$$

Stochastic Optimal Control: Bellman principle

$$J(n(t), m(t), t) \quad = \min_{u(t,t_f]} \mathbb{E}_{(N(s),r(s))|_{s=t}^{s=t_f}} \left[\phi(m(t_f), n(t_f)) \ + \int_t^{t_f} \ell(m(\tau), u(\tau)) d\tau \right].$$

$$\ell(m(t), n(t), u(t)) = \frac{1}{2}(1 - m(t))^2 + \frac{1}{2} q u^2(t),$$

Lemma. The **optimal cost-to-go** satisfies **Bellman's Principle of Optimality**

$$J(n(t), m(t), t) = \min_{u(t,t+dt]} \mathbb{E}[J(n(t+dt), m(t+dt), t+dt)] + \ell(n(t), m(t), u(t))$$

Proof same as before.

Stochastic Optimal Control: Solution

$$J(m(t), n(t), t) = \min_{u(t,t+dt]} \mathbb{E}[J(m(t+dt), n(t+dt), t+dt)] + \ell(m(t), n(t), u(t))\,dt$$

$$0 = \min_{u(t,t+dt]} \mathbb{E}[dJ(m(t), n(t), t)] + \ell(m(t), n(t), u(t))\,dt.$$

$$dJ(m,n,t) = J_t(m,n,t) - nmJ_m(m,n,t) + [J(1,(1-\alpha)n,t)r(t) + J(1,(1+\beta)n,t)(1-r) - J(m,n,t)]dN(t).$$

$$u_d^*(t) = q^{-1}\left[J_d(m(t),n(t),t) - J_d(1,(1-\alpha)n(t),t)m(t) - J_d(1,(1+\beta)n(t),t)(1-m(t))\right]_+$$

Optimal solution (MEMORIZE): $u(t) = q^{-\frac{1}{2}}(1 - m(t))$

It only depends on the recall failure prob.!

http://learning.mpi-sws.org/memorize/

Evaluating Memorize: Dataset

- **Natural experiment** on Duolingo:
  - 12 million sessions
  - 5.3 million unique (user, word) pairs

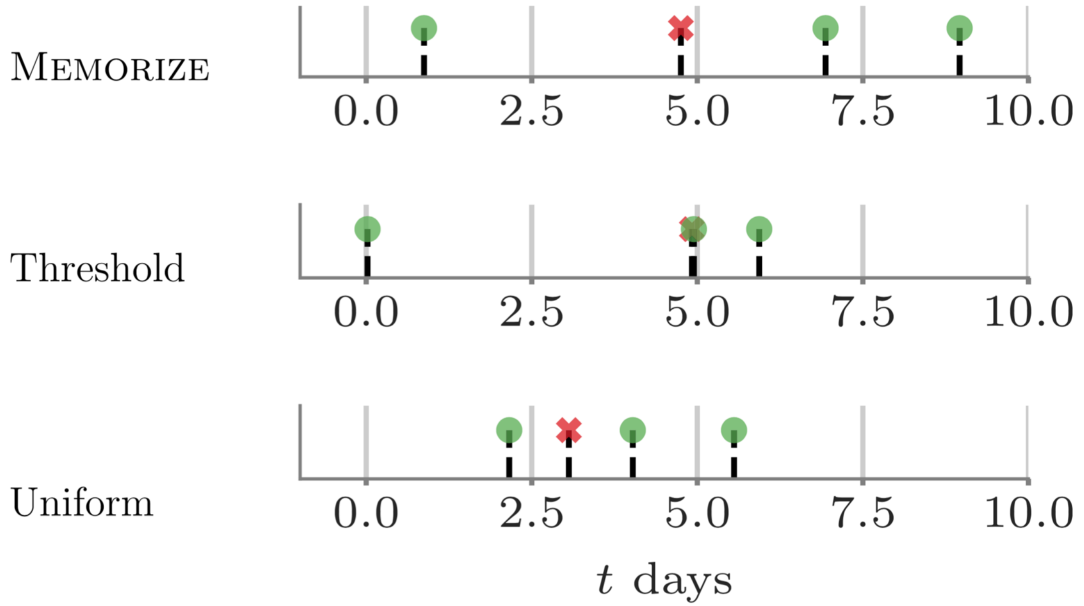
[Settles & Meeder, 2016]

http://learning.mpi-sws.org/memorize/

Evaluating Memorize: Metric

- **Natural experiment** on Duolingo:
  - 12 million sessions
  - 5.3 million unique (user, word) pairs

- Find (user, item) pairs closest to each scheduler using top-quantile by likelihood.



Evaluating Memorize: Metric

- **Natural experiment** on Duolingo:
  - 12 million sessions
  - 5.3 million unique (user, word) pairs

- Find (user, item) pairs closest to each scheduler using top-quantile by likelihood.

- Relative *empirical forgetting rate* as metric:
  - Treat first n – 1 sessions as “study”
  - Treat last attempt as the “test”, calculate forgetting rate

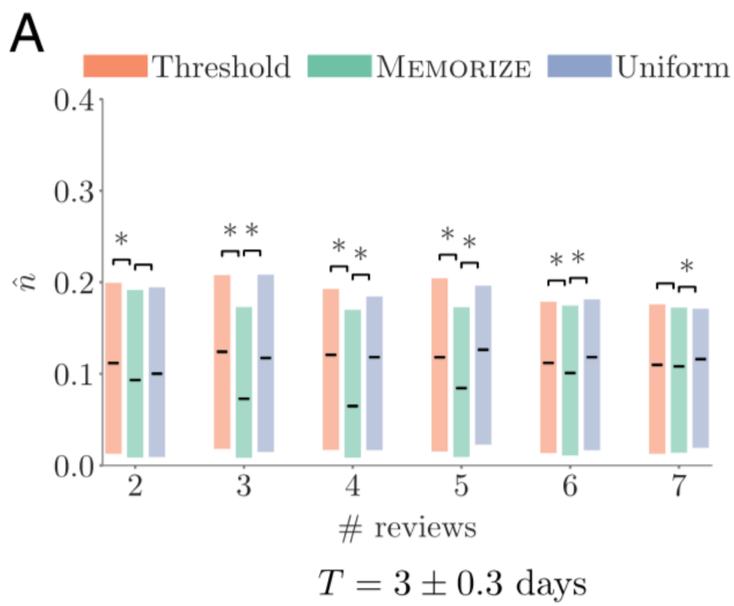
$$\hat{n} = -\log(\hat{m}(t_n))/(t_n - t_{n-1}),$$

Evaluating Memorize: Results

Control for:

- Number of reviews: n
- Duration of study: $T = t_{n-1} - t_1$

Lower is better



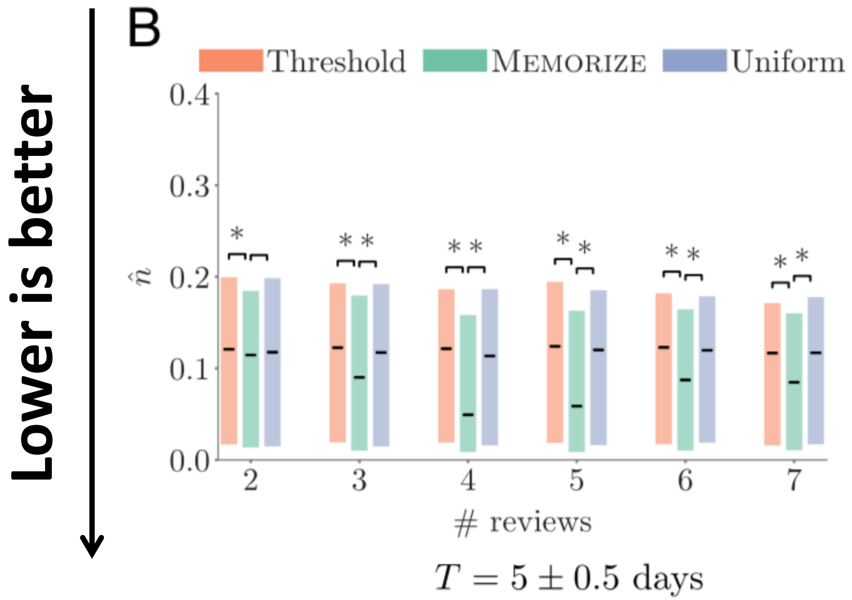
http://learning.mpi-sws.org/memorize/

[Tabibian *et. al.* 2019]

Evaluating Memorize: Results

Control for:

- Number of reviews: n
- Duration of study: $T = t_{n-1} - t_1$



http://learning.mpi-sws.org/memorize/

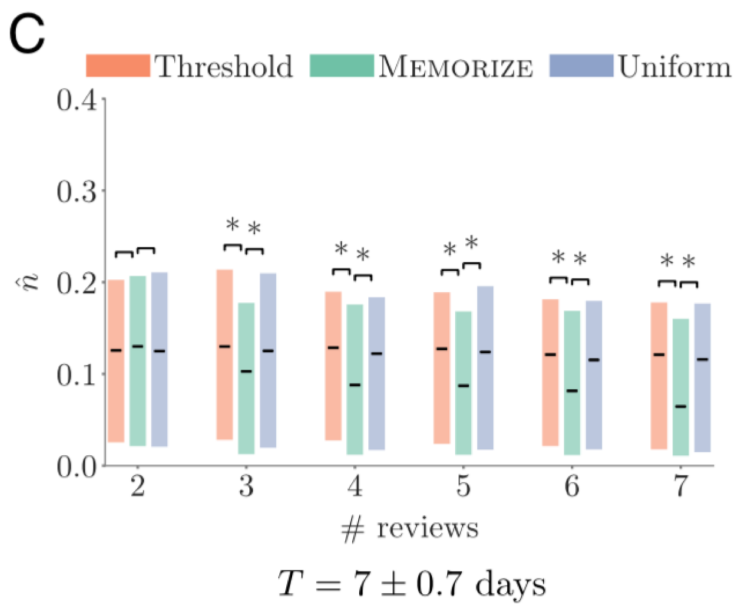
[Tabibian *et. al.* 2019]

Evaluating Memorize: Results

Control for:

- Number of reviews: n
- Duration of study: $T = t_{n-1} - t_1$

Lower is better



http://learning.mpi-sws.org/memorize/

[Tabibian *et. al.* 2019]

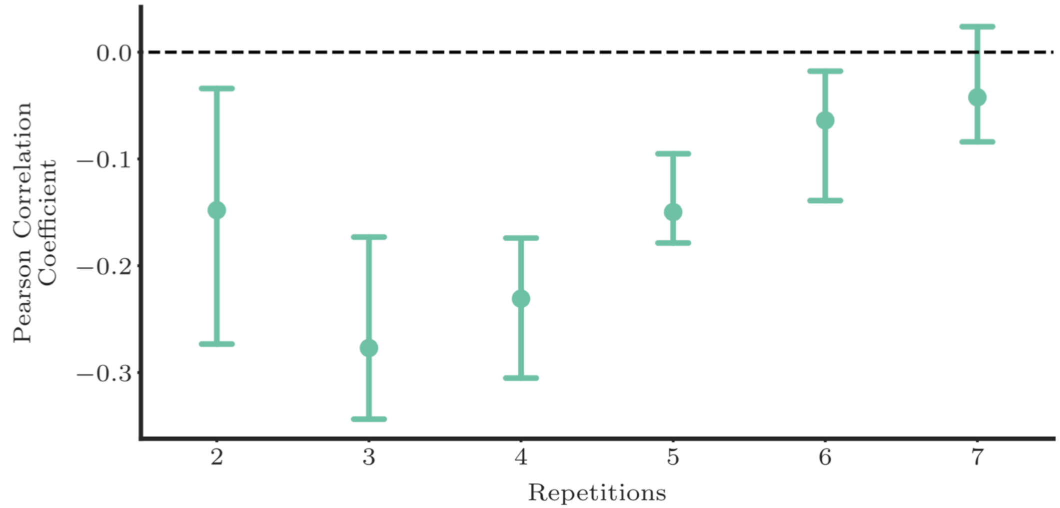
Evaluating Memorize: Results

Control for:

- Number of reviews: n
- Duration of study: $T = t_{n-1} - t_1$

Correlation between LL of following Memorize and $\hat{n}$

Lower is better



$T = 8 \pm 3.2$ days

http://learning.mpi-sws.org/memorize/

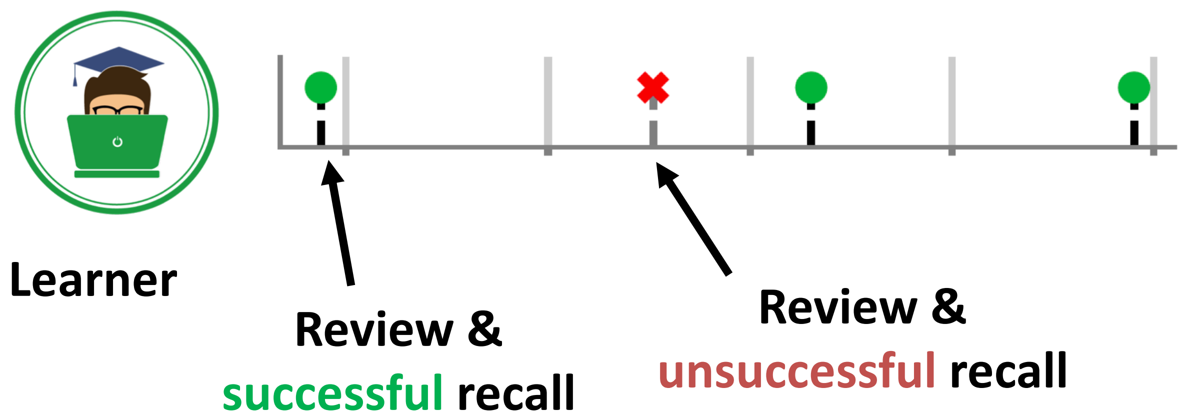
[Tabibian *et. al.* 2019]

Case for Reinforcement Learning

Agent

Online learning platform

Environment



The Memory Model is actually complicated:

- Massed repetition
- Dependence between items
- Multiscale Context Model

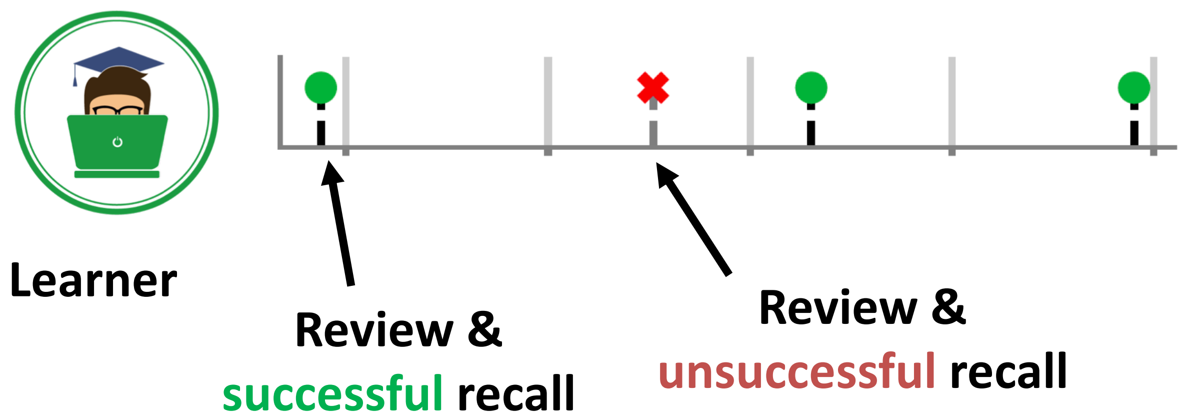
Case for Reinforcement Learning

Agent

duolingo

Online learning platform

Environment



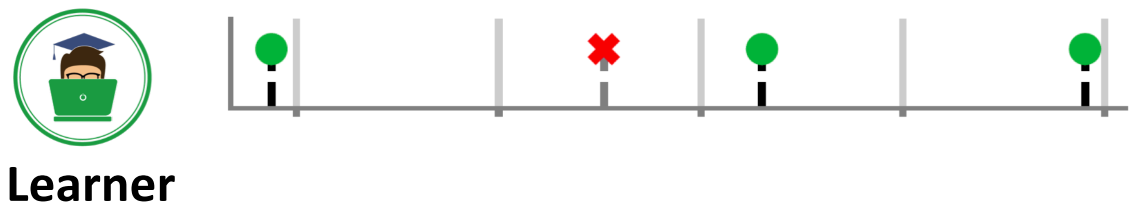
When to review to maximize recall probability?

Improve continuous retention ✓

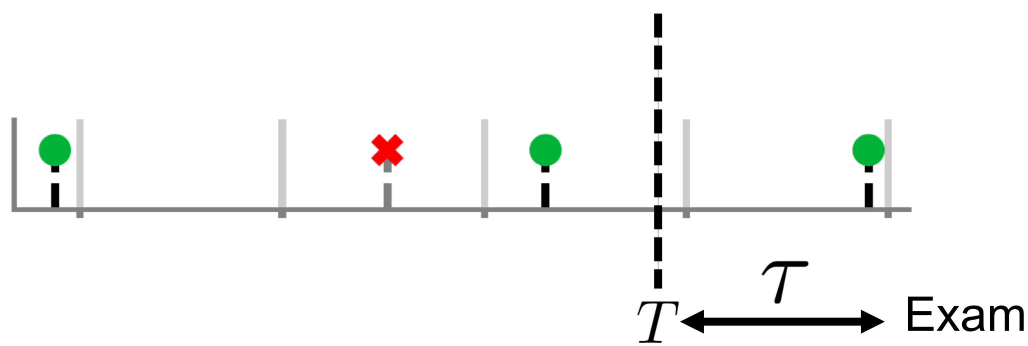
Improve test scores ✗

Complex Memory model and rewards

Agent | **Environment**



However, one may have access to test scores:



Key idea:

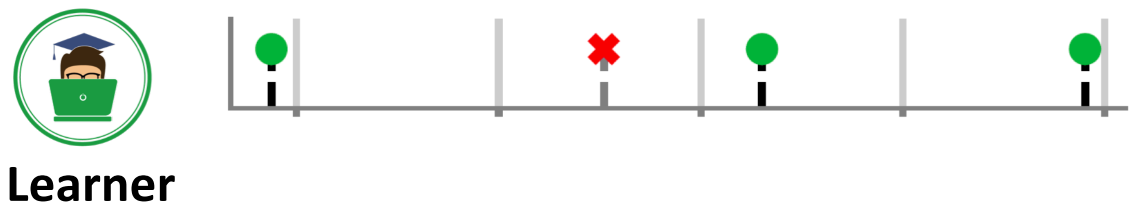
Think of the test score as rewards in a reinforcement learning setting!

[Upadhyay et al., 2018]

Teacher actions and Student feedback

Agent

Environment



$$p^*_{\mathcal{A};\theta} = (\lambda^*_\theta, m^*_\theta)$$

Policy

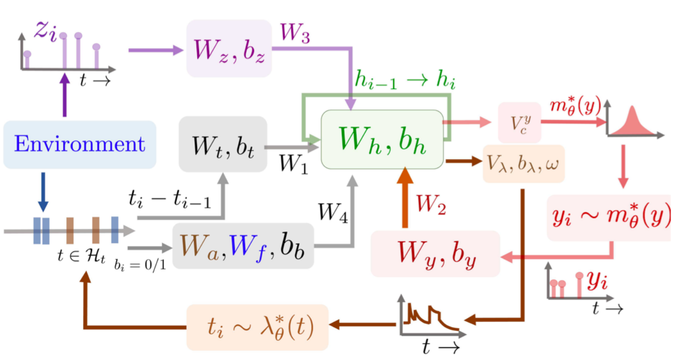
Intensity

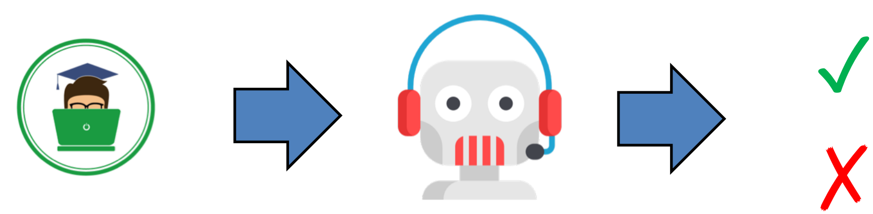
Item distribution

$$p^*_{\mathcal{F};\phi} = (\lambda^*_\phi, m^*_\phi)$$

We do not know the *feedback* distribution but we can *sample* from it…

Parametrized using RNNs





…and measure test scores (rewards)

[Upadhyay et al., 2018]

What is the goal in reinforcement learning?

We aim to maximize the average reward in a time window [0, T]:

$$\underset{p^*_{\mathcal{A};\theta}(\cdot)}{\text{maximize}} \quad \overbrace{\mathbb{E}_{\underbrace{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)}_{\text{Actions asynchronous, Feedback synchronous}}} \underbrace{[R^*(T)]}_{\text{Reward (Point)}}}^{J(\theta)}$$

Connection to optimal control:

$$J(n(t), m(t), t) = \min_{u(t,t_f]} \mathbb{E}_{(N(s), r(s))|_{s=t}^{s=t_f}} \left[\phi(m(t_f), n(t_f)) + \int_t^{t_f} \ell(m(\tau), u(\tau)) d\tau \right].$$

[Upadhyay et al., 2018]

Policy gradient

We use gradient descent to improve the policy, i.e., the intensity, over time:

$$\theta_{l+1} = \theta_l + \alpha_l \nabla_\theta J(\theta)|_{\theta=\theta_l}$$

We need to compute the gradient of an average. But the average depends on the parameters!

$$\nabla_\theta J(\theta) = \nabla_\theta \ \mathbb{E}_{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)} \left[R^*(T) \right]$$

Parameters!

[Upadhyay et al., 2018]

Reinforce trick to compute gradient

The reinforce trick allows us to overcome this implicit dependence:

$$\nabla_\theta J(\theta) = \nabla_\theta \ \mathbb{E}_{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)} \left[R^*(T) \right]$$

Appendix A in Upadhyay et al., 2018

$$\nabla_\theta J(\theta) = \mathbb{E}_{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)} \left[R^*(T) \nabla_\theta \log \mathbb{P}_\theta(\mathcal{A}_T) \right]$$

Parameters!

[Upadhyay et al., 2018]

Likelihood of action events

$$\nabla_\theta J(\theta) = \mathbb{E}_{\mathcal{A}_T \sim p^*_{\mathcal{A};\theta}(\cdot), \mathcal{F}_T \sim p^*_{\mathcal{F};\phi}(\cdot)} \left[R^*(T) \nabla_\theta \log \mathbb{P}_\theta(\mathcal{A}_T) \right]$$

Likelihood of posts by our broadcaster!

$$\mathbb{P}(\mathcal{A}_T) := \left(\prod_{e_i \in \mathcal{A}_T} \lambda^*_\theta(t_i) \right) \exp\left(- \int_0^T \lambda^*_\theta(s)\, ds \right)$$

The key remaining question is how to parametrize the intensity $\lambda^*_\theta(t)$

Parameters & functional form!

[Upadhyay et al., 2018]

Policy parametrization

Parameters

Output layer:

Last time of review

$$\lambda^*_\theta(t) = \exp\left(b_\lambda + w_t(t - t') + \boldsymbol{V}_\lambda \boldsymbol{h}_i\right)$$

$$\mathbb{P}[y_{i+1} = c] = \frac{\exp\langle V^y_{c,:}, h_i\rangle}{\sum_{l\in\mathbb{Z}} \exp\langle V^y_{l,:}, h_i\rangle}$$

Hidden layer:

$$\boldsymbol{h}_i = \tanh(\boldsymbol{W}_h \boldsymbol{h}_{i-1} + \boldsymbol{W}_1 \boldsymbol{\tau}_i + \boldsymbol{W}_4 \mathbf{b}_i + \boldsymbol{b}_h)$$

Input layer:

e_i Item

$$\boldsymbol{\tau}_i = \boldsymbol{W}_t(t_i - t_{i-1}) + \boldsymbol{b}_t$$

$$\mathbf{b}_i = \boldsymbol{W}_f e_i + \boldsymbol{b}_b$$

i-th event (i-1)-th event

[Upadhyay et al., 2018]

Sampling from the policy

$$\lambda_\theta^*(t) = \exp\left(b_\lambda + w_t(t - t') + \boldsymbol{V}_\lambda \boldsymbol{h}_i\right)$$

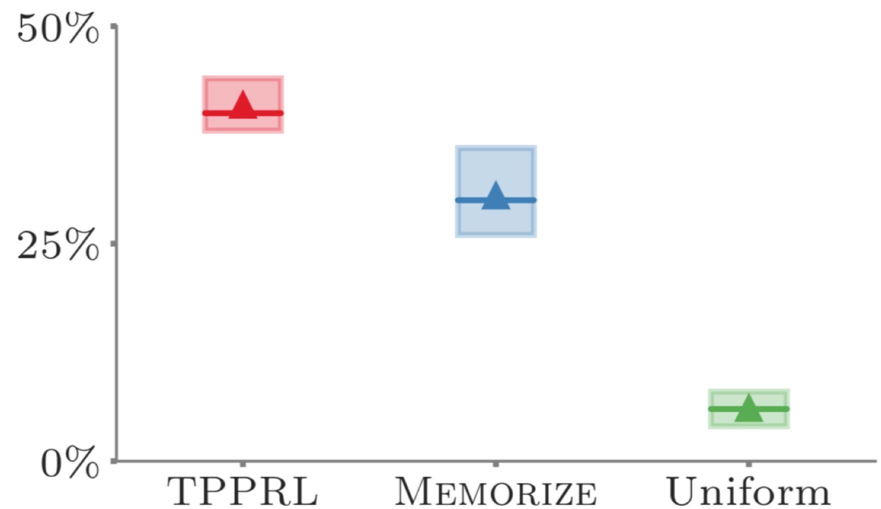
The intensity can increase or decrease every time an event by the other broadcasters take place:

- We cannot apply just superposition
- We can use inversion sampling: The CDF is a function by parts, where each part is defined once an event by the other broadcasters happens

Appendix C in
[Upadhyay et al., 2018]

Results: Improved test scores

Higher is better



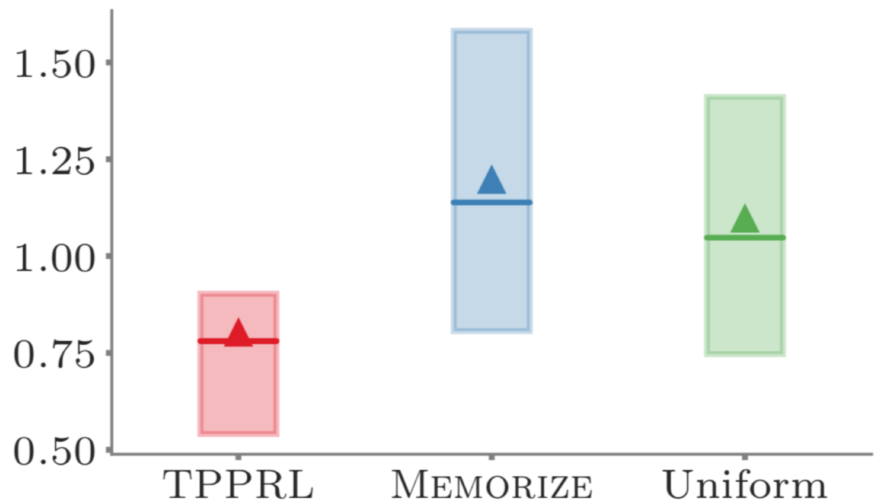
(a) Recall

RL

It *learns* the difficulty and memory model

[Upadhyay et al., 2018]

Results: Intuition



(b) Items' difficulty

RL
It *learns* the difficulty and memory model

[Upadhyay et al., 2018]