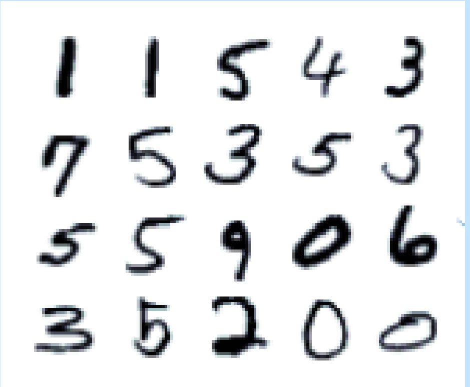
Real World Reinforcement Learning

John Langford

Tutorial Slides: http://hunch.net/~rwil
Vowpal Wabbit: http://hunch.net/~vw
Decision Service: http://ds.microsoft.com

With help from many!

The Supervised Learning Paradigm



Training examples

1	1	5	4	3
7	5	3	5	3
5	5	9	0	6
3	5	2	0	0

Training labels

Supervised Learner

Accurate digit classifier

2

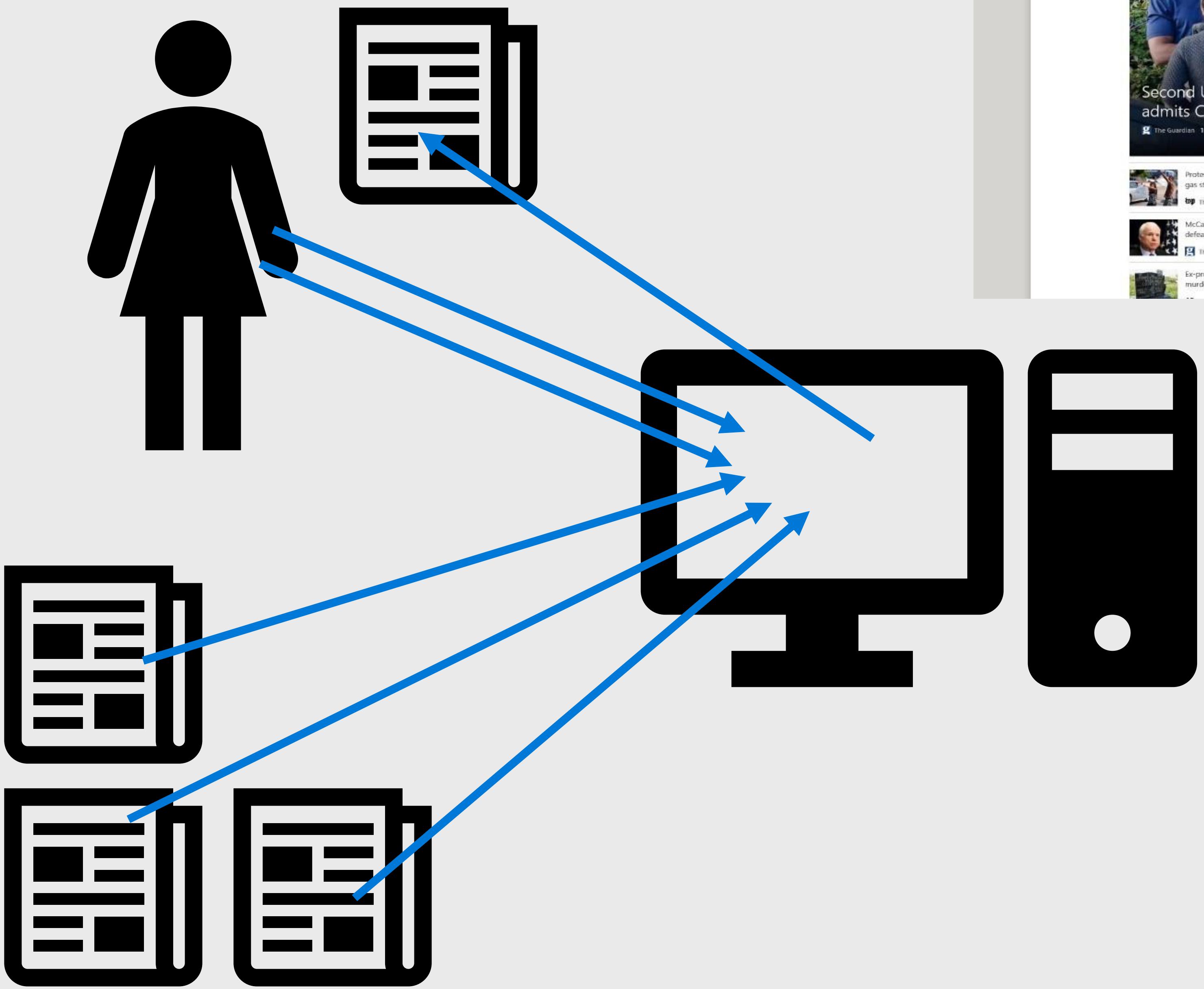
Supervised Learning is cool

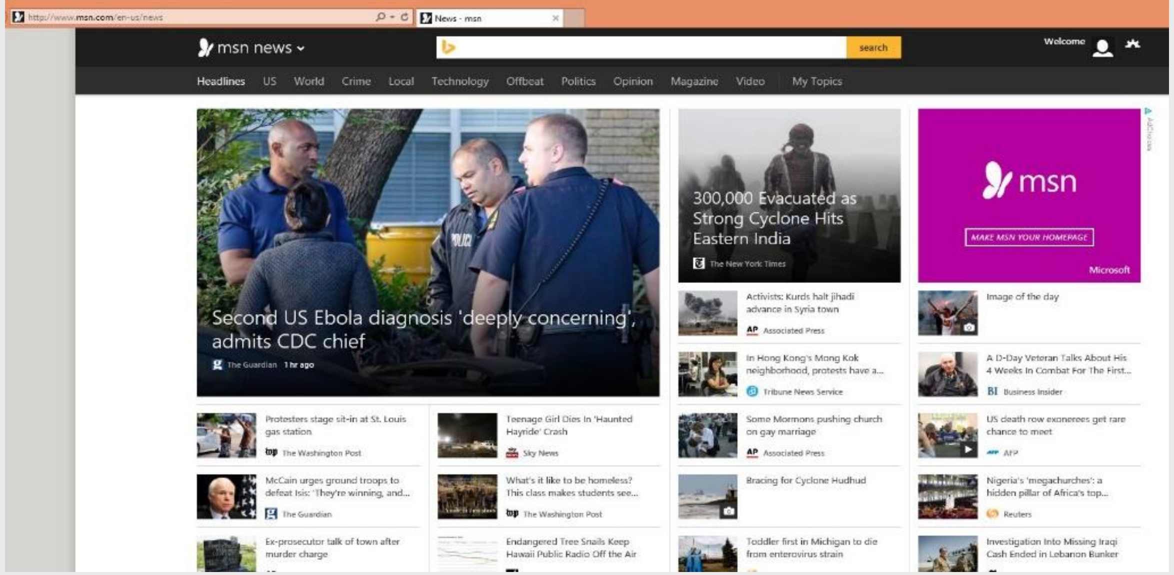
A problem is solved if:

A human can tell the right answer.

(Many times)

How about news?





A standard pipeline

1. Collect $(user, article, click)$ information.
2. Build $features(user, article)$
3. Learn $\hat{P}(click|features(user, article))$
4. Act: $\arg\max_{\{articles\}} \hat{P}(click|features(user, article))$
5. Deploy in A/B test for 2 weeks
6. A/B test fails ☹ Why?

Q: What goes wrong?

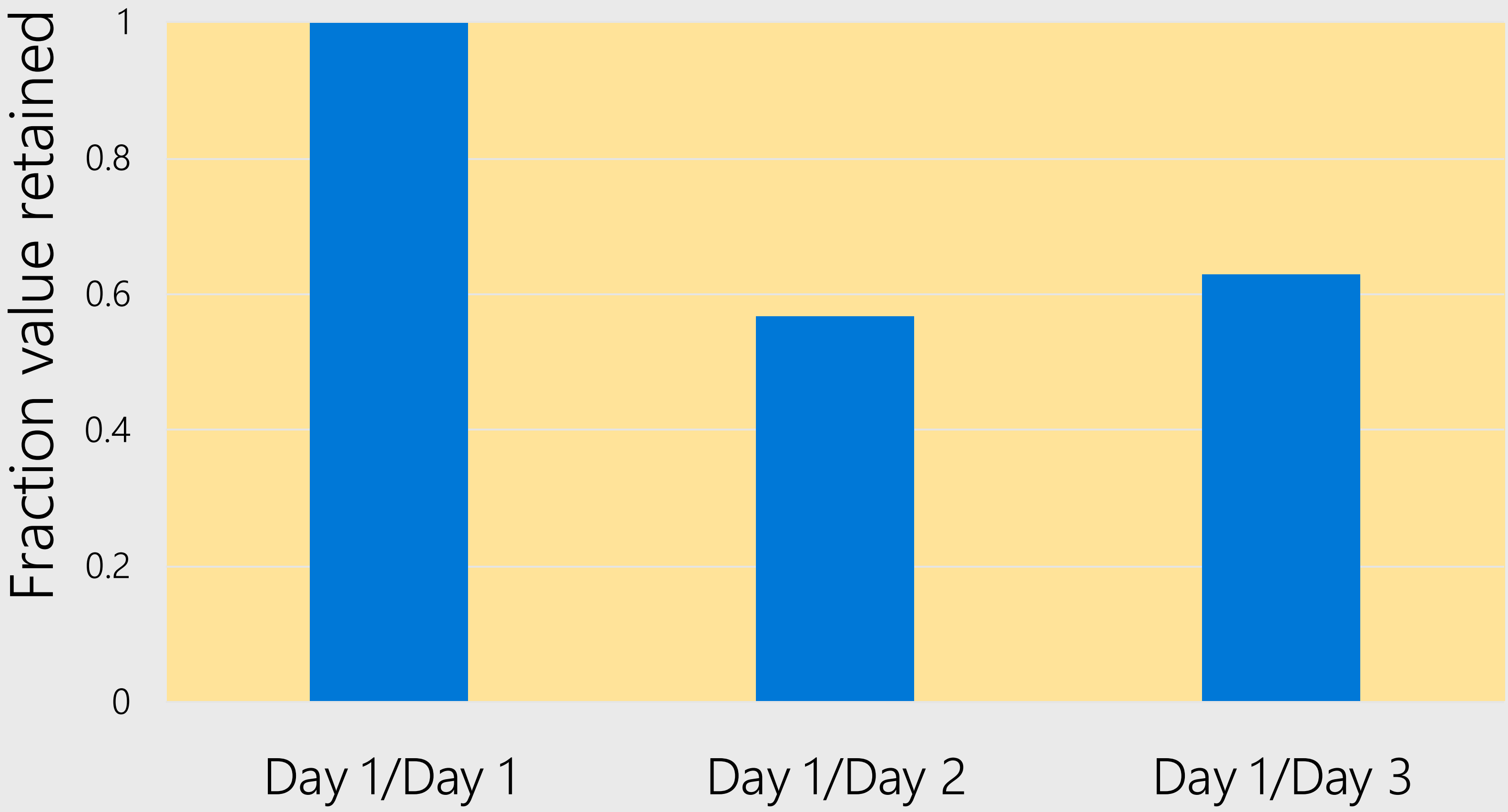
Is Ukraine 

interesting to John  ?

A: Need Right Signal for Right Answer

Q: What goes wrong?

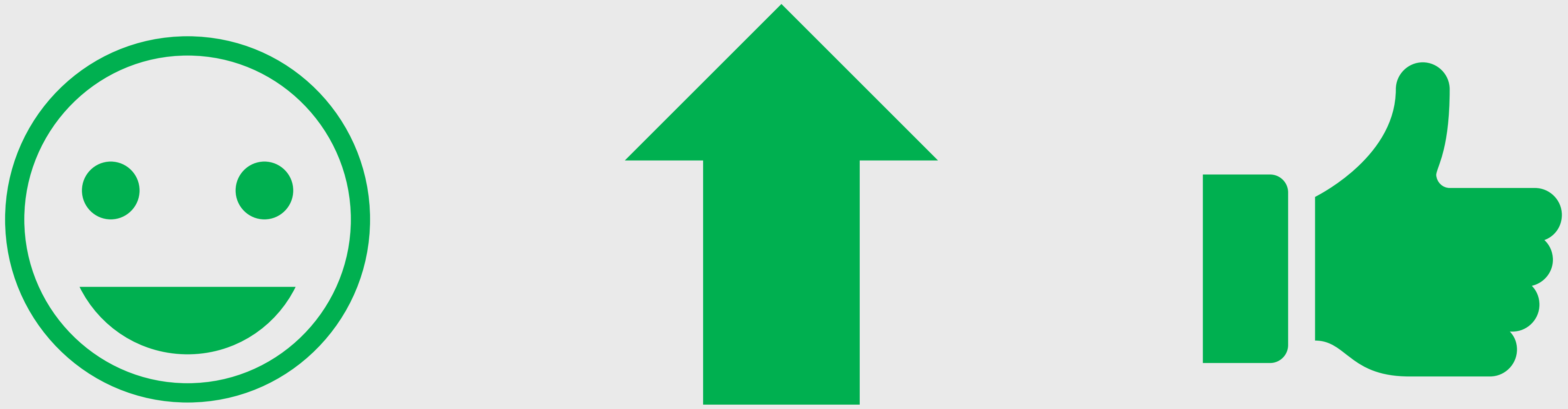
Model value over time



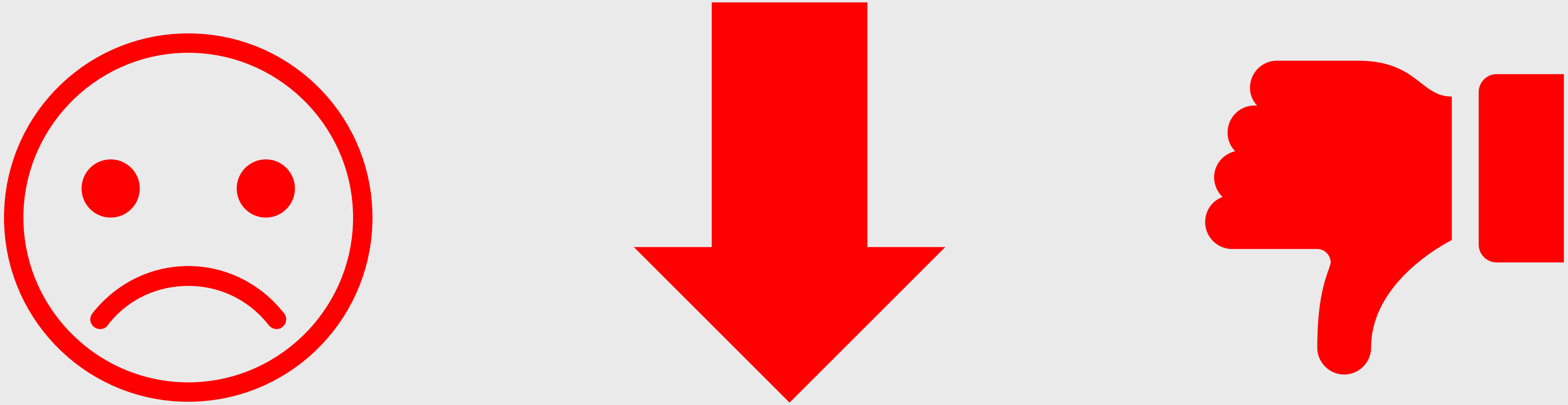
A: The world changes!

HOW?

GOOD

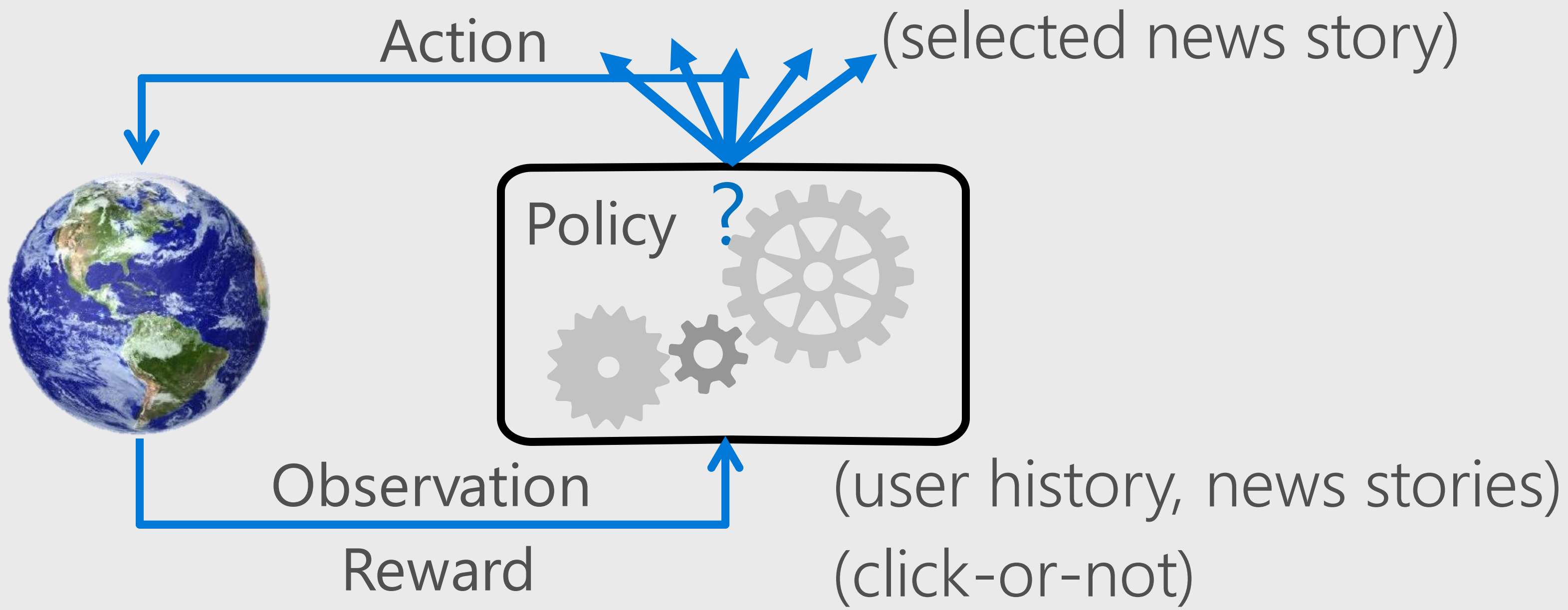


BAD

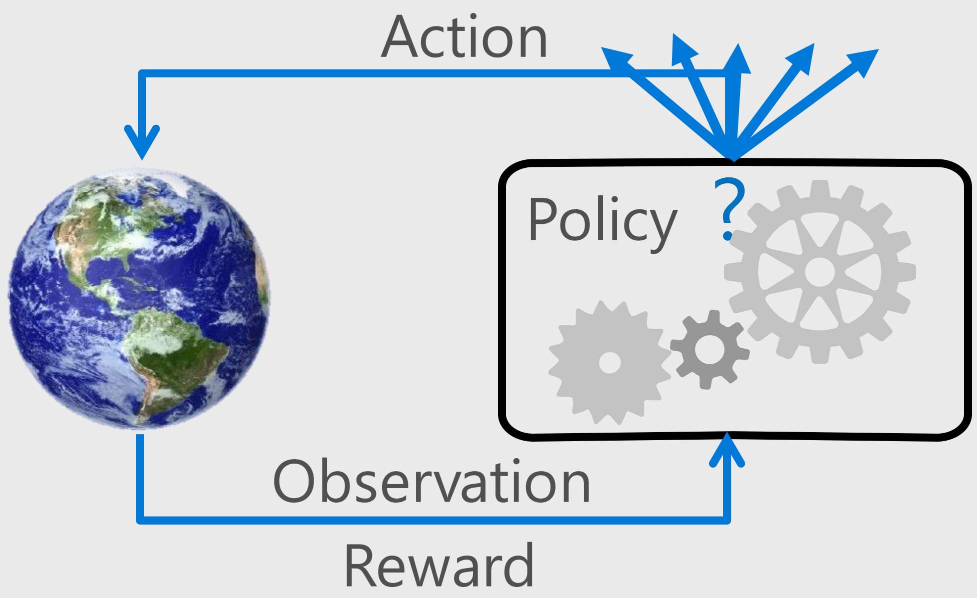


How do you learn from Reward signal?

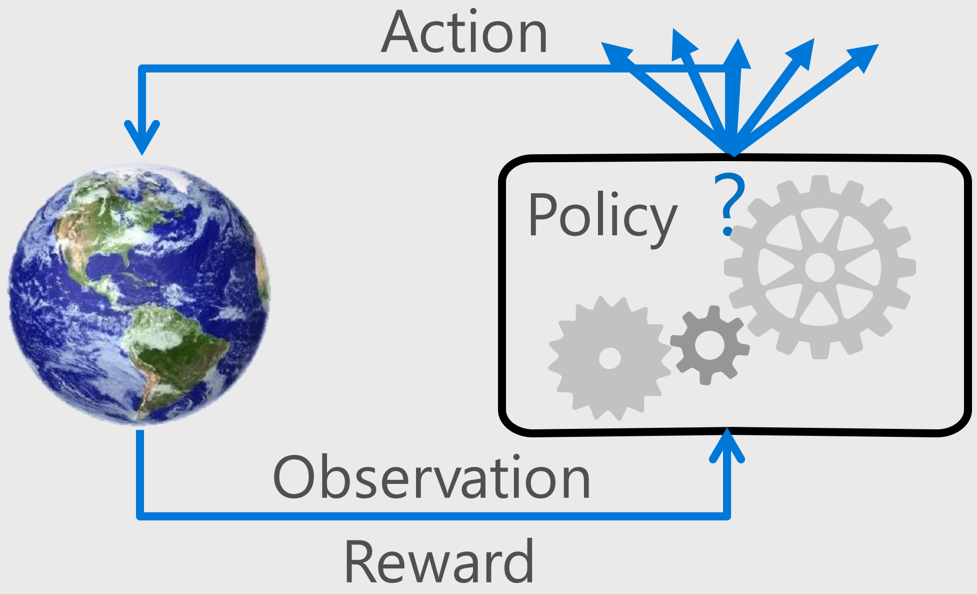
Reinforcement Learning



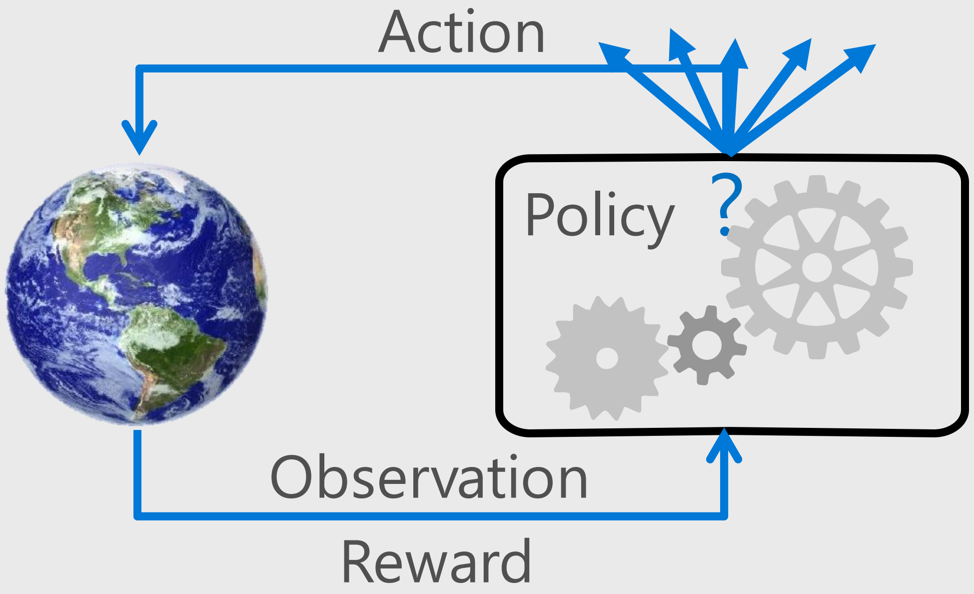
Multistep Reinforcement Learning



Multistep Reinforcement Learning



Multistep Reinforcement Learning KL02, KKL03, SLWLL06, DaLM09, CKADaL15, CHRDaL16, KAL16, JKALS17, MLA17, DaLMS18?...



Goal: maximize sum of rewards.

Applications: 

A simple problem breaking all common multistep Reinforcement Learning algos




A boy's fire starts to die down

So he goes searching for wood

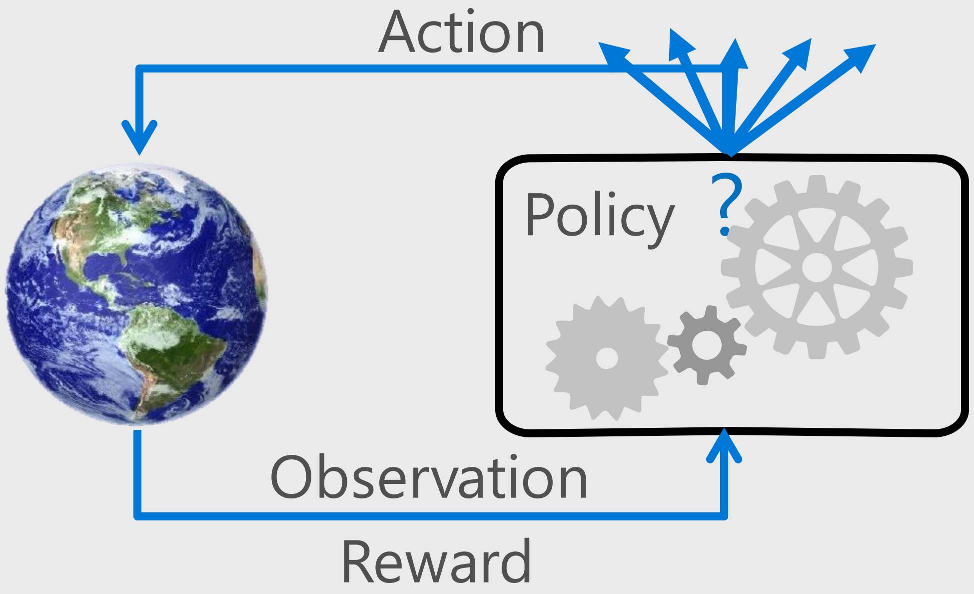
But he gets cold far from the fire

So he returns to the fire

The boy is warm for awhile

But then the fire goes out

Contextual Bandits LZ07, BL09, LWLS10, SLLK10, DuLL11, DuHKKLRZ11, LWLW11, BLLRS, ADuKLS12, DELL12, BLS14, AHKLLS14, ABCLLLMORSS16, AKADuL17, …



Goal: maximize sum of rewards.

Applications: Recommendation, Personalization, etc…

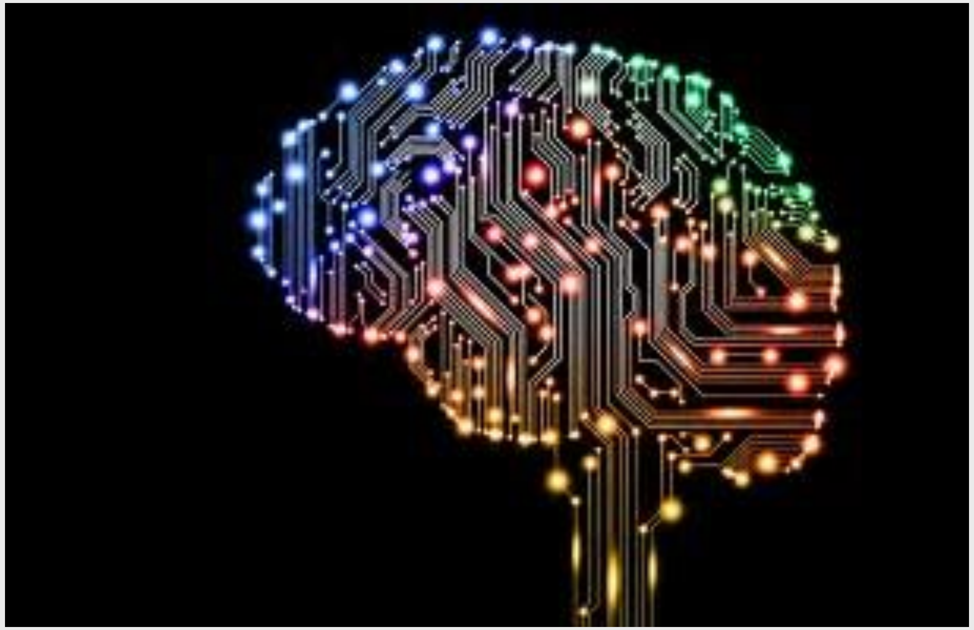
Why Else?

A: $$$

Use free interaction data

rather than expensive labels

Why else?



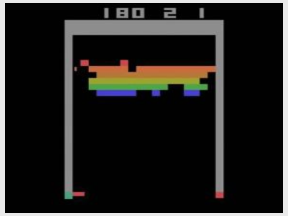
~~AI: A function programmed with data~~

AI: An economically viable digital agent that explores, learns, and acts

Flavors of Interactive Learning

Multistep Reinforcement Learning:

Special Domains

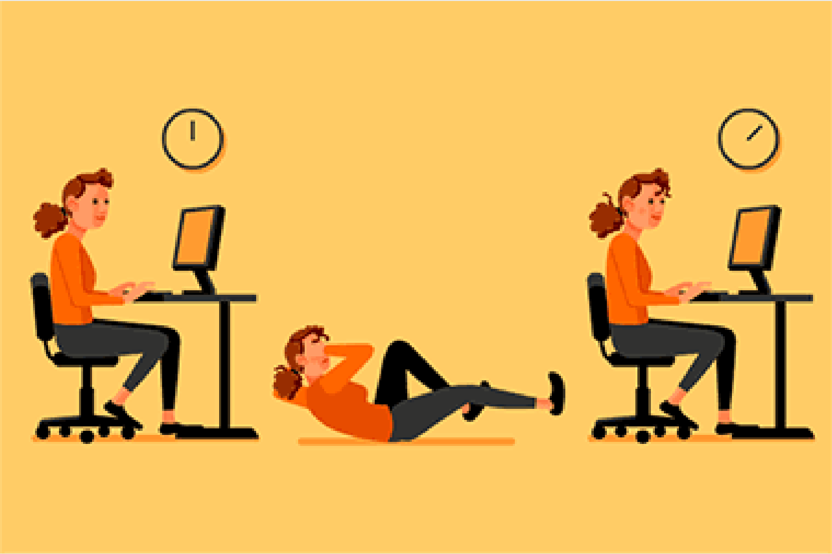


+Right Signal, -Nonstationary Bad, -$$$ +AI

Contextual Bandits: Immediate Reward RL

±Rightish Signal, +Nonstationary ok, +$$$, +AI

Ex: Which advice?



Repeatedly:

1. Observe features of user+advice
2. Choose an advice.
3. Observe steps walked

Goal: Healthy behaviors

Many real-world applications ☺

News Rec: [LCLS '10]

Ad Choice: [BPQCCPRSS '12]

Ad Format: [TRSA '13]

Education: [MLLBP '14]

Music Rec: [WWHW '14]

Robotics: [PG '16]

Wellness/Health: [ZKZ '09, SLLSPM '11, NSTWCSM '14, PGCRRH '14, NHS '15, KHSBATM '15, HFKMTY '16]

A problem is solved if:

An outcome value can be measured.

(Many times)

Contextual Bandits

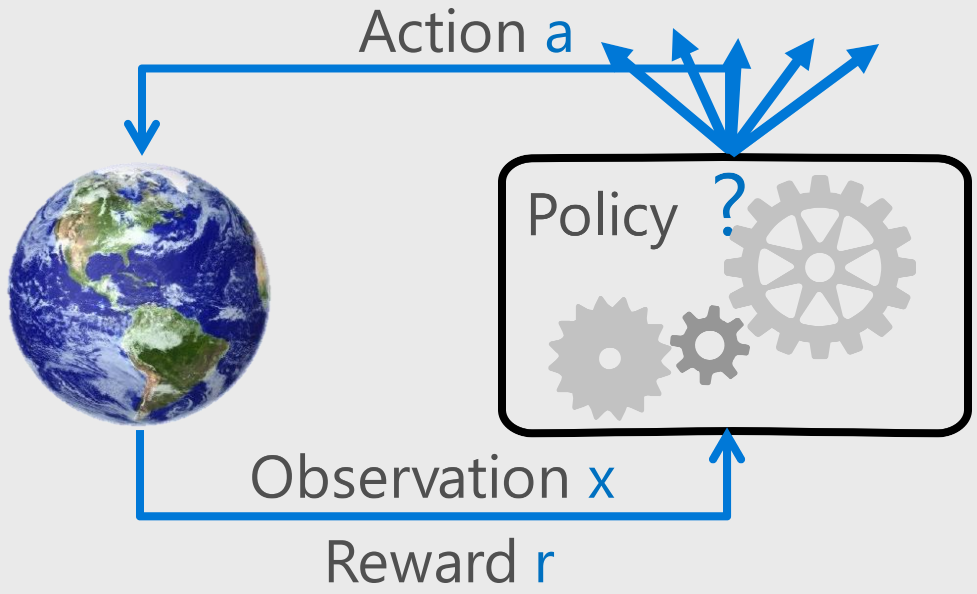
1) Good fit for many real problems

Outline

What can we do?

1) Evaluate?
2) Learn?
3) Explore?

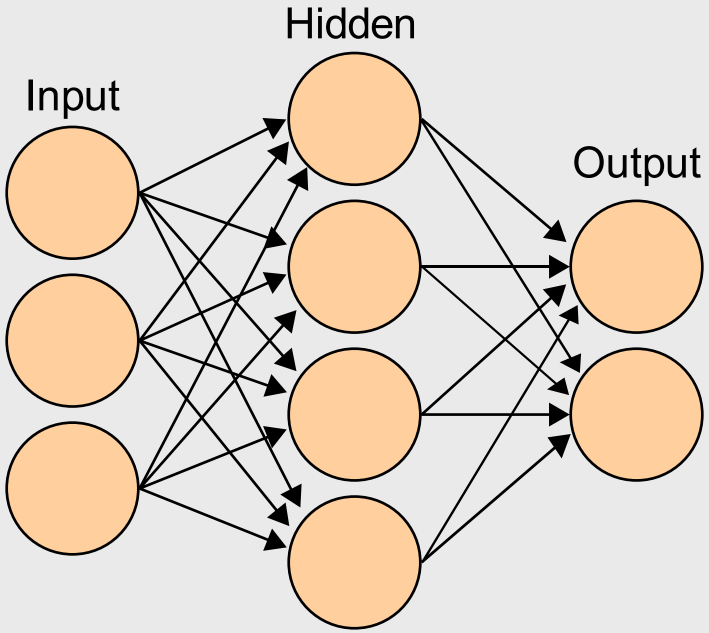
Contextual Bandits

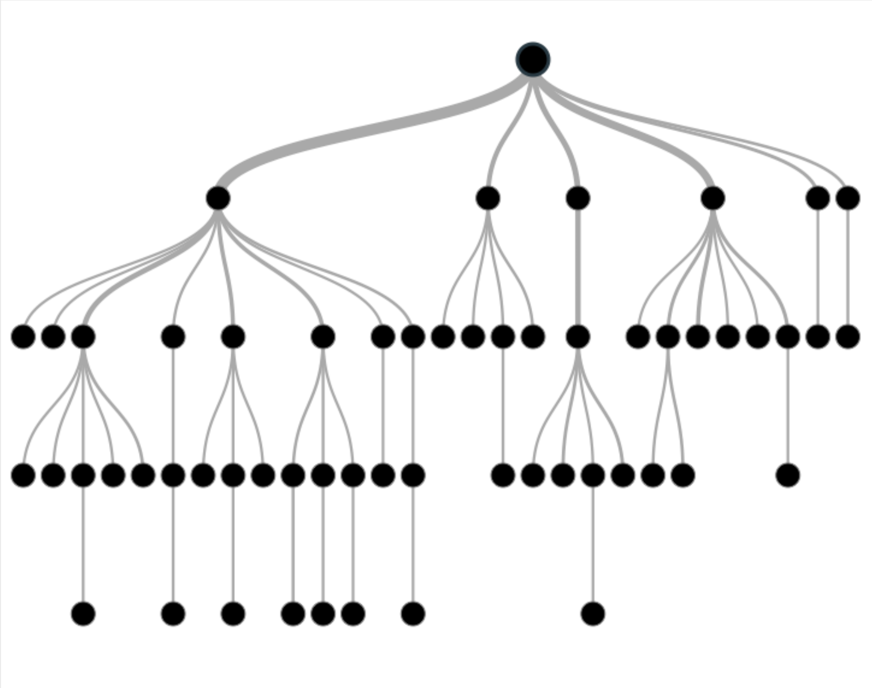


Goal: maximize sum of rewards.

Policies

Policy maps features to actions.

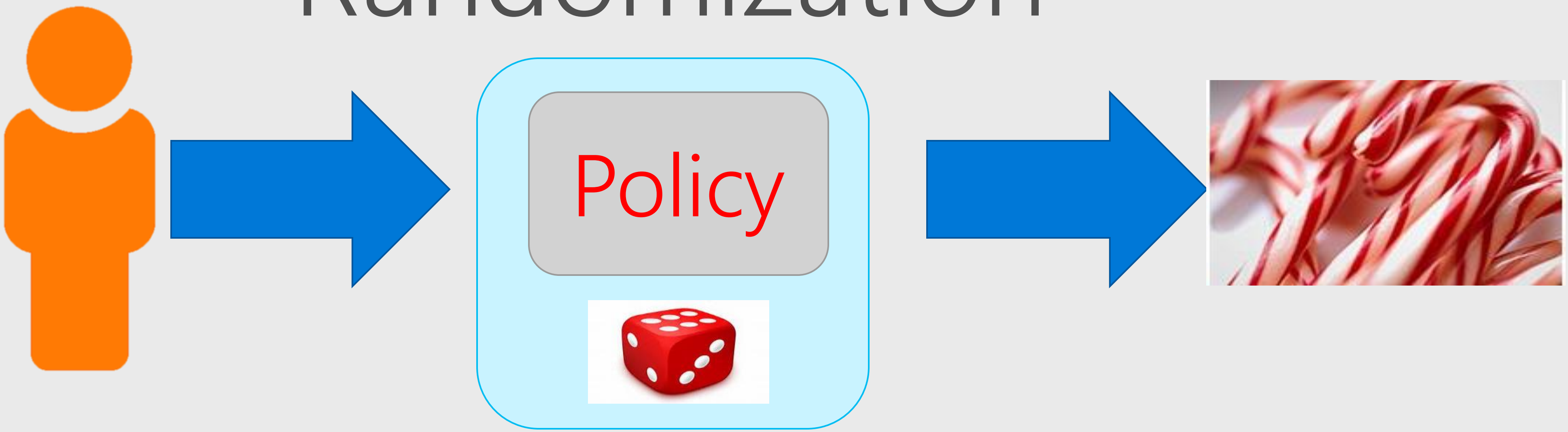




Policy = Classifier that *acts*.

Fundamental: Exploration needed

Randomization



Inverse Propensity Score(IPS) [HT '52]

Given experience $\{(x, a, p, r)\}$ and a policy $\pi: x \to a$, how good is π?

$$V_{\text{IPS}}(\pi) = \frac{1}{n} \sum_{(x,a,p,r)} \frac{r I(\pi(x) = a)}{p}$$

Propensity Score

What do we know about IPS?

Theorem: For all π, for all $D(x,\vec{r})$

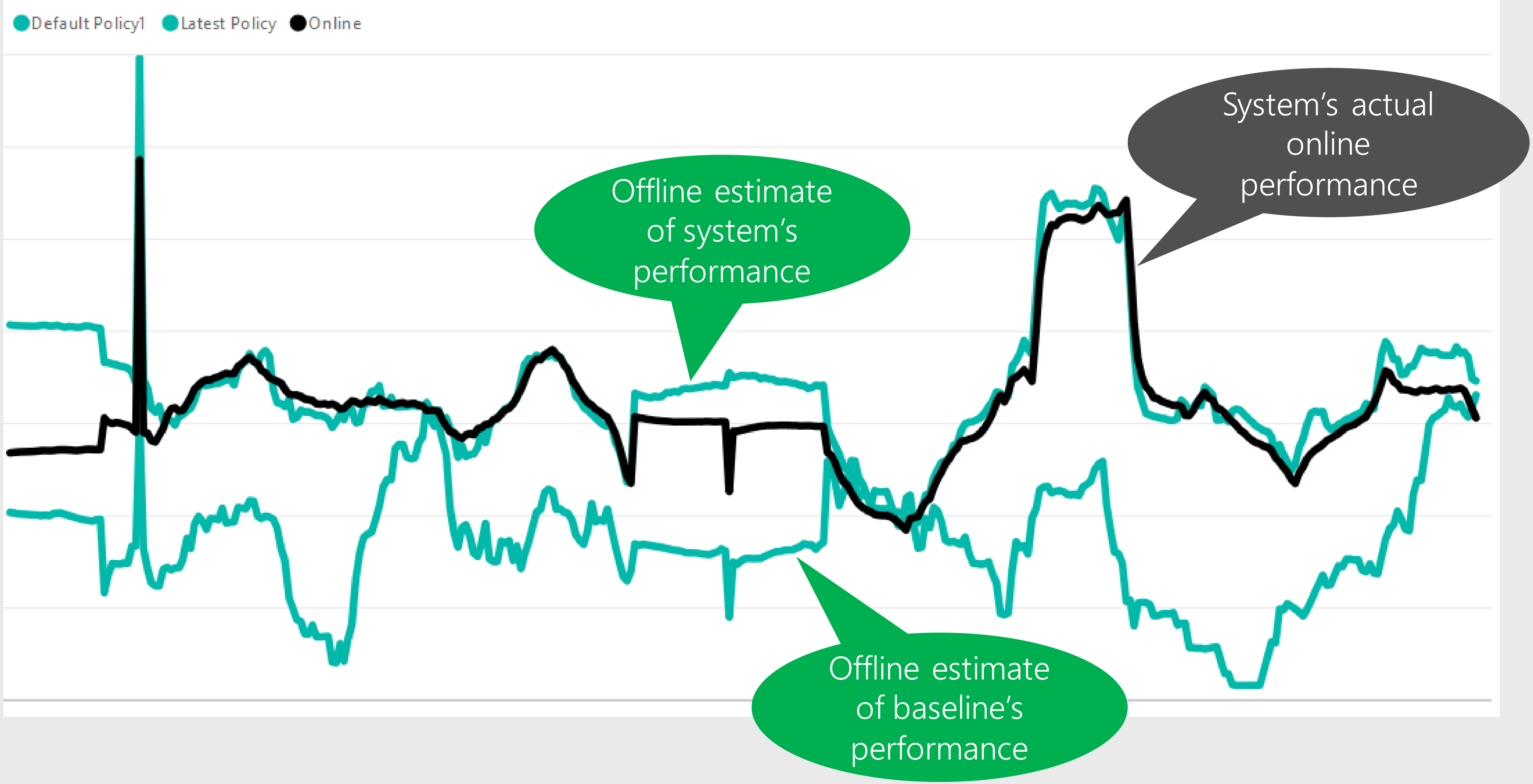
$$\mathrm{E}\left[r_{\pi(x)}\right] = \mathrm{E}[V_{\mathrm{IPS}}(\pi)] = \mathrm{E}\left[\frac{1}{n}\sum_{(x,a,p,r)}\frac{rI(\pi(x)=a)}{p}\right]$$

Proof: For all $(x,\vec{r})$, $E_{a\sim\vec{p}}\left[\frac{r_a I(\pi(x)=a)}{p_a}\right]$

$$= \sum_a p_a \frac{r_a I(\pi(x)=a)}{p_a}$$

$$= r_{\pi(x)}$$

Reward over time



Better Evaluation Techniques

Double Robust: [DLL ‘11]

Weighted IPS: [K ’92, SJ ‘15]

Clipping: [BL ’08]

Learning from Exploration ['Z 03]

Given Data $\{(x, a, p, r)\}$ how to maximize $\mathbf{E}[r_{\pi(x)}]$?

Maximize $\mathbf{E}\left[V_{\text{IPS}}(\pi)\right]$ instead!

$$r_a = \begin{cases} r/p & \text{if } \pi(x) = a \\ 0 & \text{otherwise} \end{cases}$$

Equivalent to:

$$r'_a = \begin{cases} 1 & \text{if } \pi(x) = a \\ 0 & \text{otherwise} \end{cases}$$

with importance weight $\frac{\boldsymbol{r}}{\boldsymbol{p}}$

Importance weighted multiclass classification!

Vowpal Wabbit: Online/Fast learning

- BSD License, 10 year project
- Mailing List>500, Github>1K forks, >5K stars, >1K issues, >100 contributors
- Command Line/C++/C#/Python/Java/AzureML/Daemon

VW for Contextual Bandit Learning

echo “1:2:0.5 | here are some features” | vw --cb 2

Format: <action>:<loss>:<probability> | features…

Training on a large dataset:

vw --cb 2 rcv1.cb.gz --ngram 2 --skips 4 -b 24

Result: 0.048616

Better Learning from Exploration Data

Policy Gradient: [W '92]

Offset Tree: [BL '09]

Double Robust for learning: [DLL '11]

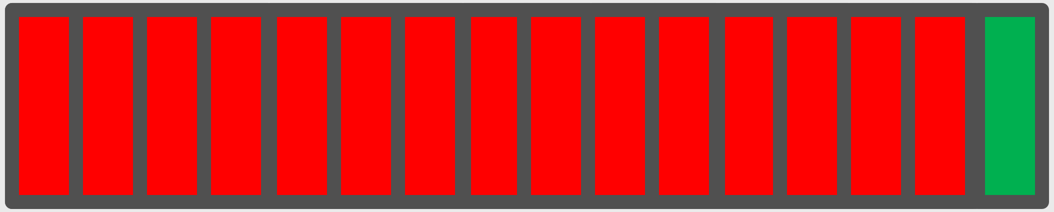
Multitask Regression: https://arxiv.org/abs/1802.04064

Weighted IPS for learning: [SJ '15]

Evaluating Online Learning

Problem: How do you evaluate an online learning algorithm Offline?

Answer: Use Progressive Validation [BKL '99, CCG '04]



Theorem:

1) Expected PV value = Uniform expected policy value.

2) Trust like a test set error.

How do you do Exploration?

Simplest Algorithm: ϵ-greedy.

With probability ϵ act uniform random

With probability $1 - \epsilon$ act greedily

Better Exploration Algorithms

Better algorithms maintain ensemble and explore amongst actions of this ensemble.

Thompson Sampling: [T ‘33]

EXP4: [ACFS ‘02]

Epoch Greedy: [LZ ‘07]

Polytime: [DHKKLRZ ‘11]

Cover&Bag: [AHKLLS ‘14]

Bootstrap: [EK ‘14]

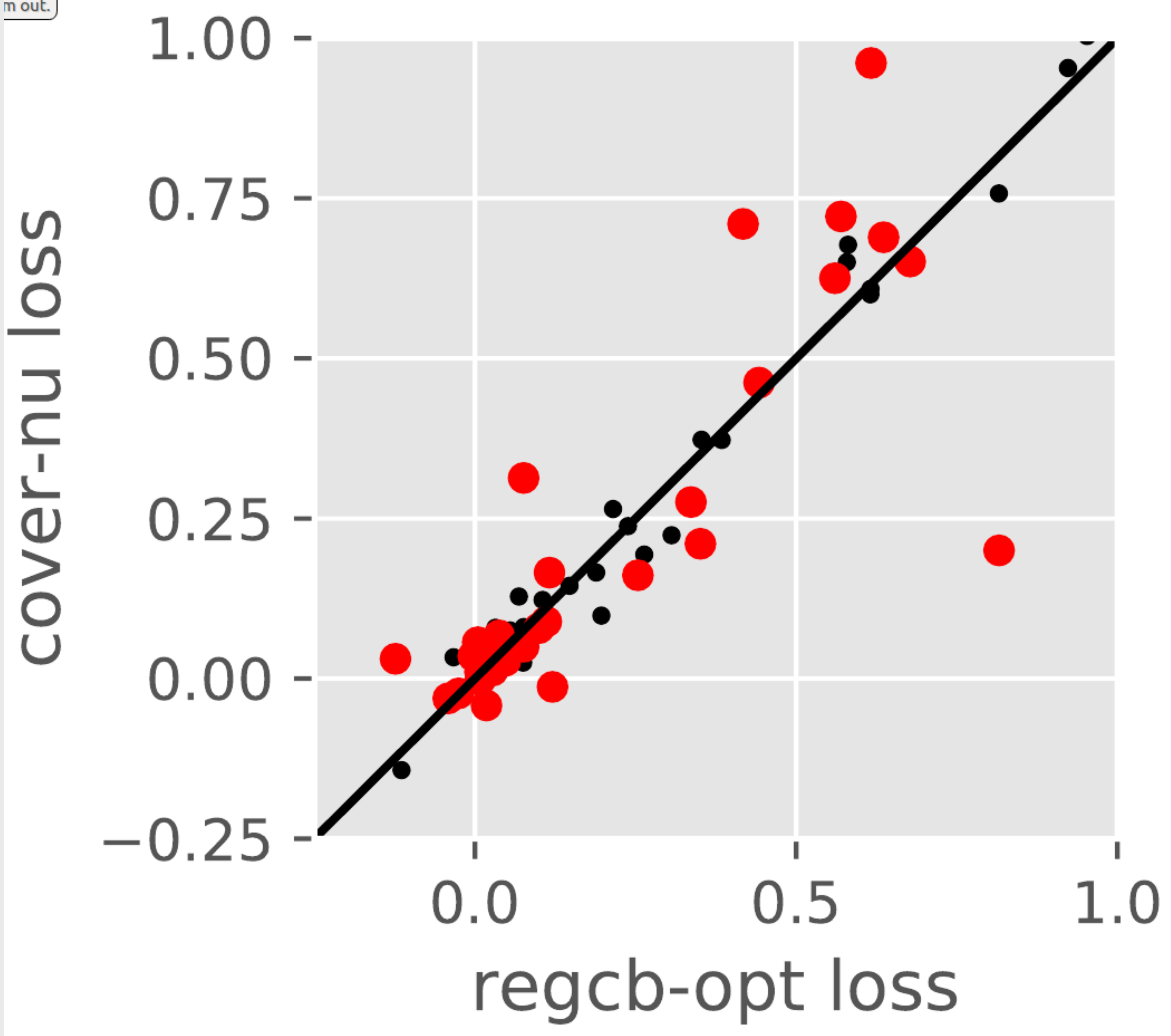
Which is best?

500 datasets say:

Regressor Conf.

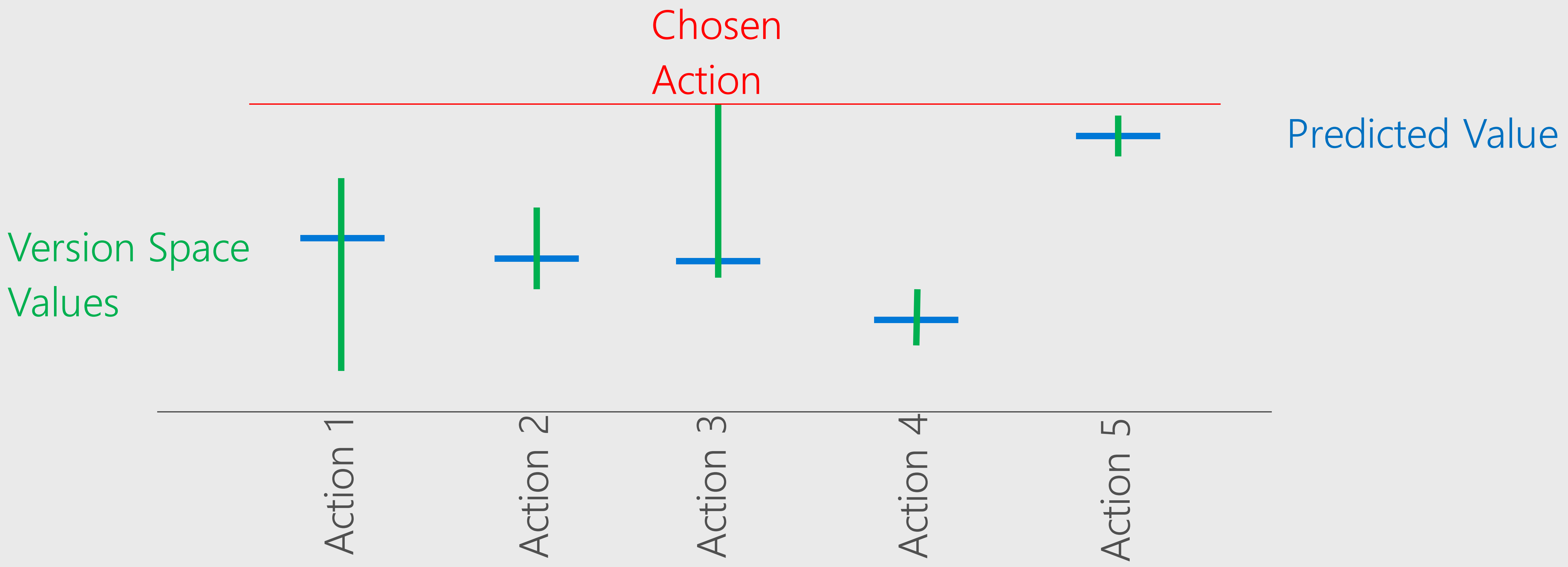
~= Cover

~= Greedy

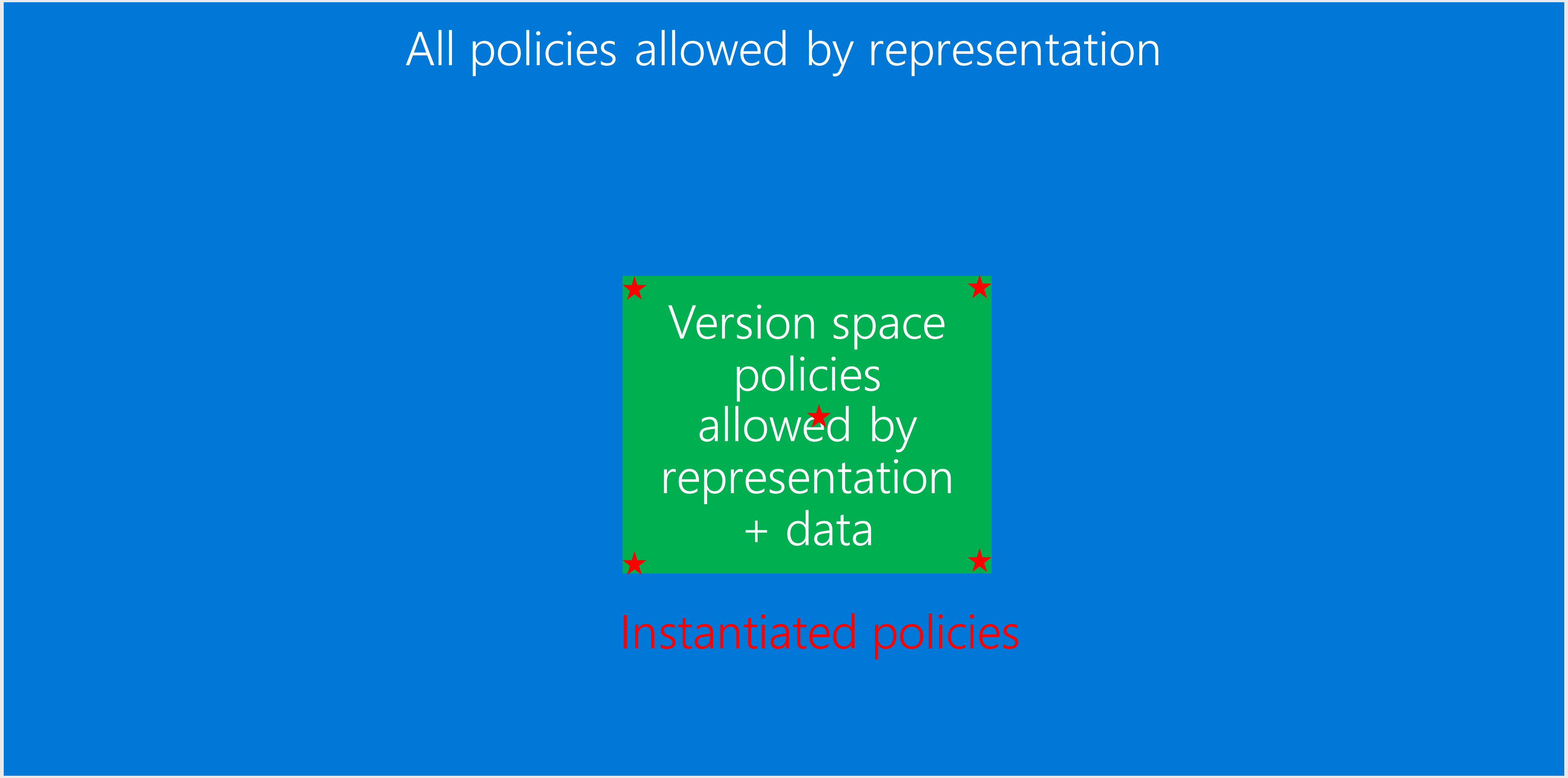


https://arxiv.org/abs/1802.04064

What is Regressor Confidence?



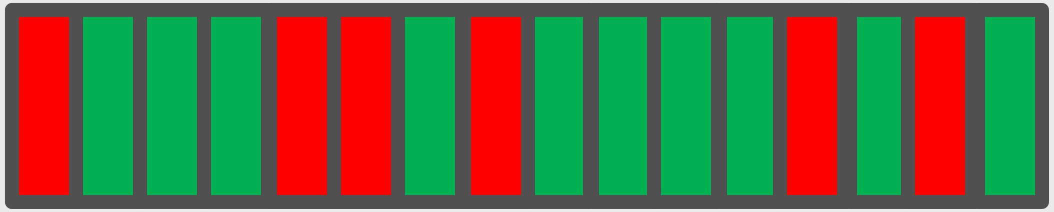
What is Cover?



Evaluating Exploration Algorithms

Problem: How do you take the choice of examples acquired by an exploration algorithm into account?

Answer: Rejection Sample from history. [DELL ‘12]



Theorem: Realized history is unbiased up to length observed.

Better versions: [DELL ‘14] & VW code

Contextual Bandits:

1) Good fit for many problems

2) Fundamental questions have useful answers

Decision Service [ABCHLLLMORSS '16]

https://github.com/Microsoft/mwt-ds/

- Open-source on Github
- Host and manage yourself

https://ds.microsoft.com

- Hosted as a Microsoft Cognitive Service
- Logging and model deployment managed
- Data logged to your Azure account

- Contextual bandits optimize decisions online
- Off-policy evaluation and monitoring

Eliminates bugs by design

- Log (x, a, p, key) at decision time
- Join with (r, key) after a prespecified time
- Learn on (x, a, p, r) after join

- Features in exploration and learning are same
- Logged action chosen by exploration
- No reward delay bias
- Always log probabilities
- Reproducible randomness

Systems survey

Decision Service [ABCHLLLMORSS '16]	NEXT [JJFGN '15]	StreamingBandit [KK '16]
Online CB with general policies	MAB, linear CB, dueling	Thompson Sampling
Off-policy eval/optimization	-	-
Open source and self-hosted on Azure	Open source and self-hosted on EC2	Open source and self-hosted locally
Managed on Azure	-	-