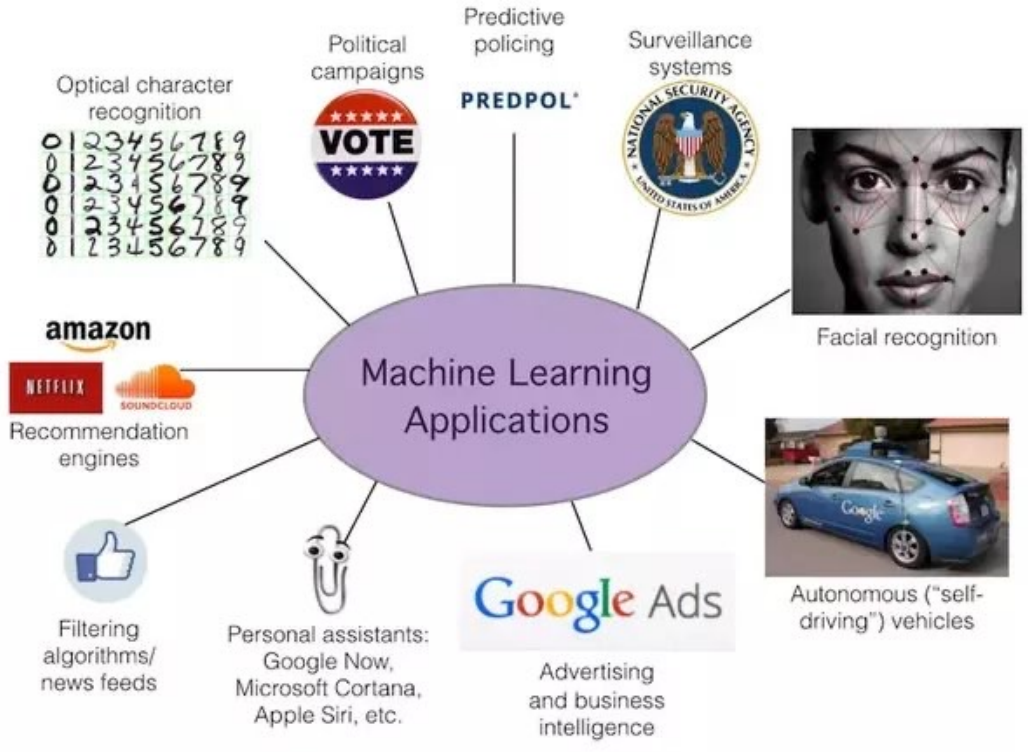
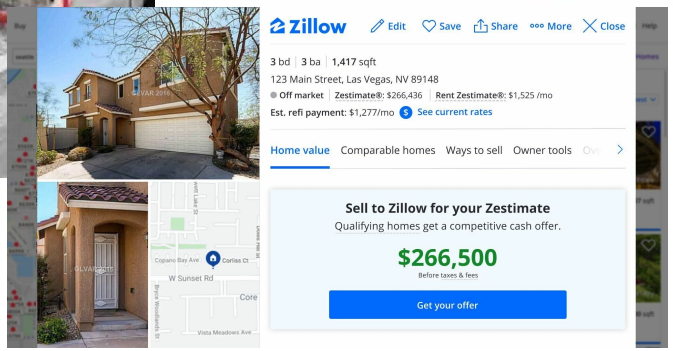
Controlling and Tracking Nonstationarity in Machine Learning

Yu-Xiang Wang

COMPUTER SCIENCE
UC SANTA BARBARA
Computing. ReInvented.

~~AI~~ Machine Learning has revolutionized almost every aspect of our daily life

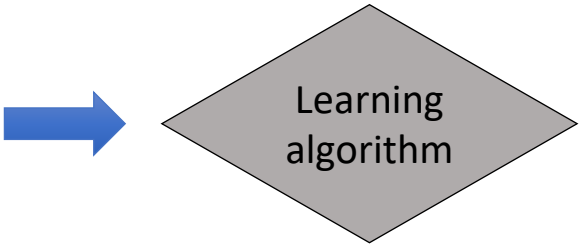


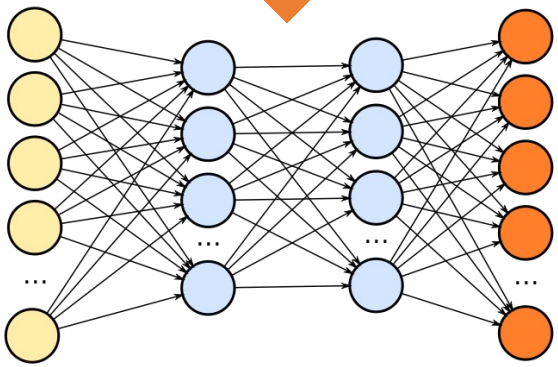


ML is capable of fitting very complex functions for accurate predictions, and generalize.

Cat

Dog





It's a "Cat"!

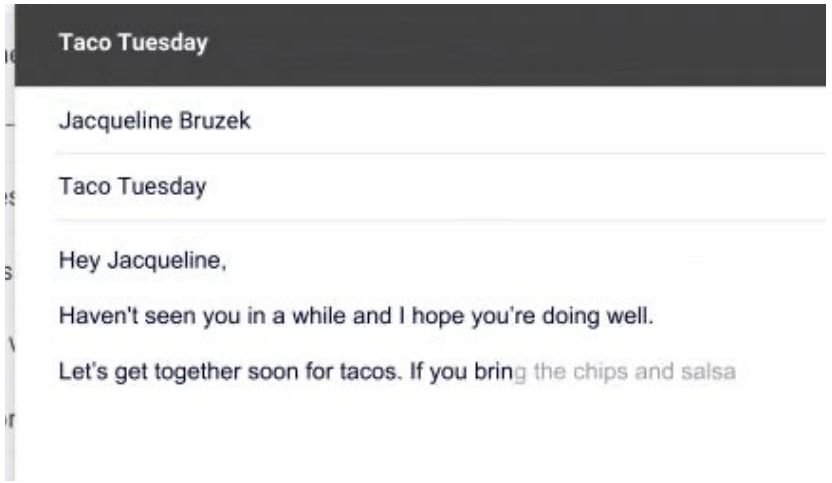
Statistical Learning Theory:
Accurate prediction when new data are from the same distribution.

Examples of typical real-life ML application

- Hospitals need to decide **who to test** based on symptoms and other patient attributes

- Train a classifier on historic records to predict the test outcome.
- The accuracy is high on a holdout set!

- Large tech wants to improve user experience on their popular email service



- Train a **large language model** with user data to **complete sentences**
- It seems to work great!

What could go wrong?

Challenge #1: Learning to Act -- Every machine learning problem is secretly a control problem

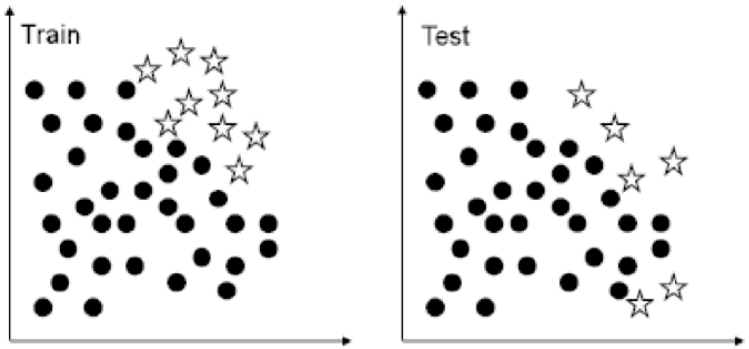
- If I test patients using the new rule, the distribution of patients receiving the test will be different!
- Should I still trust my classifier?

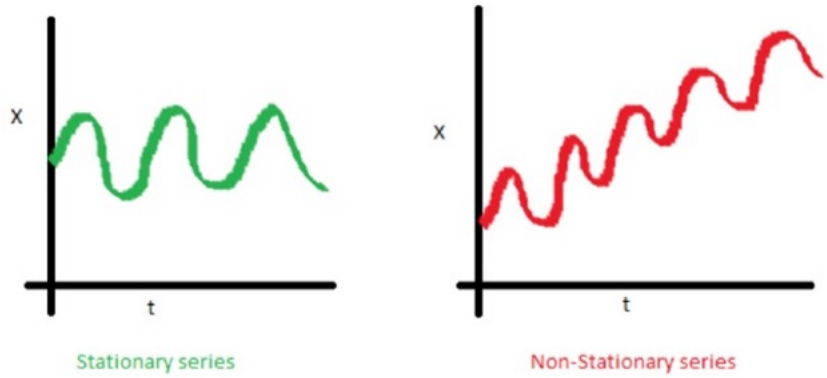
- If I deploy the new "Guess what you will write" prompt, what users will enter may change!
- Is the model fulfilling its own prophecy?

The ultimate goal is NOT prediction, but to:
minimize disease transmission / maximize user experience!

Why not model everything as a *Reinforcement Learning* problem instead?

Challenge #2: Distribution shifts
"Change is the only constant in life"





- Viruses mutate. A drug that passes a clinical trial in 2020 may become ineffective in 2021.
- Trendy topics change over time. Language models trained on older data may struggle to remain relevant.
- Stock prices are affected by events. A trading strategy can work amazingly well in one period but fail miserably when market condition changes.

We should be able to detect and track the changes somehow!

Two different types of nonstationarity

1. The distribution shift **caused** by our actions / new policies
   - Predict and control the changes

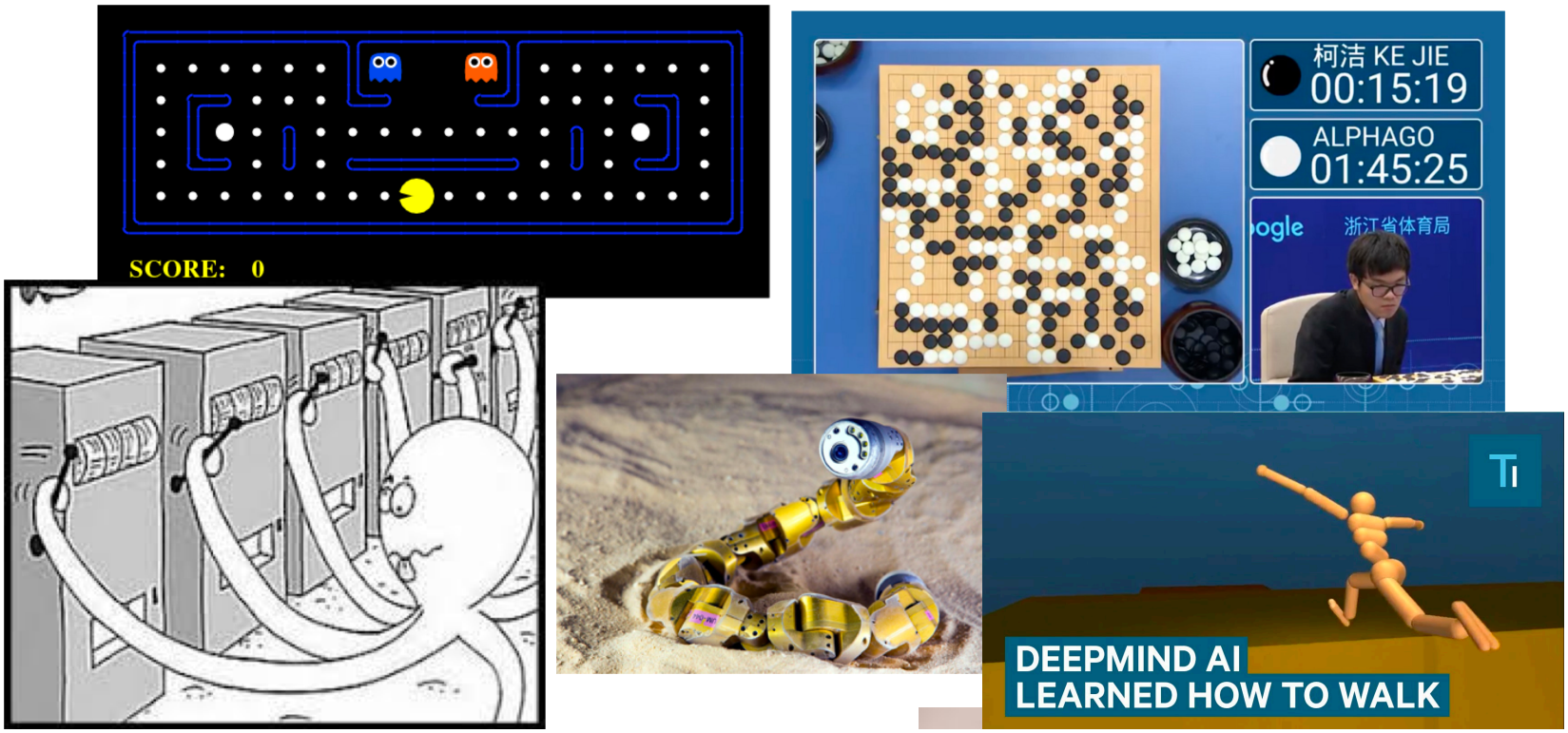
2. **Unknown** distribution shift happening in the background, due to unobserved and unpredictable factors
   - Tracking the changes, adjust models accordingly

Remainder of the talk

1. Controlling the Nonstationarity with offline reinforcement learning

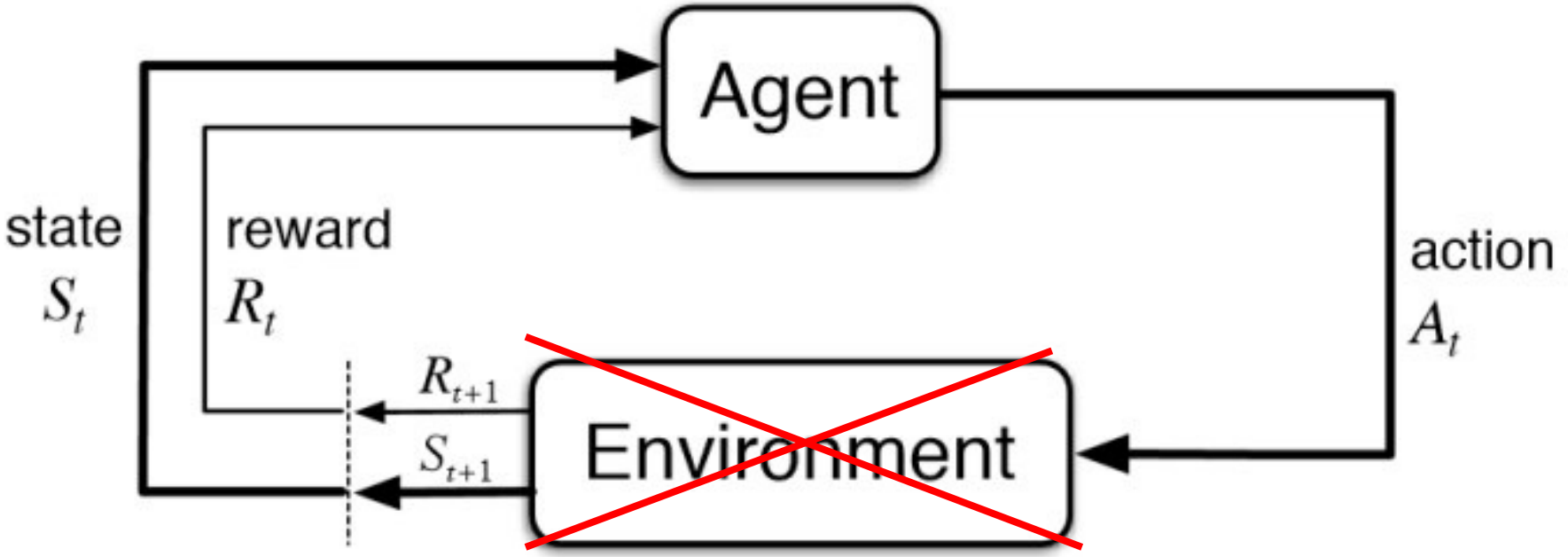
2. Tracking unknown nonstationarity with dynamic regret minimization

Reinforcement learning is among the hottest area of research in ML!



"RL" is Top 1 Keyword at NeurIPS'2021, appearing 199 times
"Deep Learning" only 129 times [source]

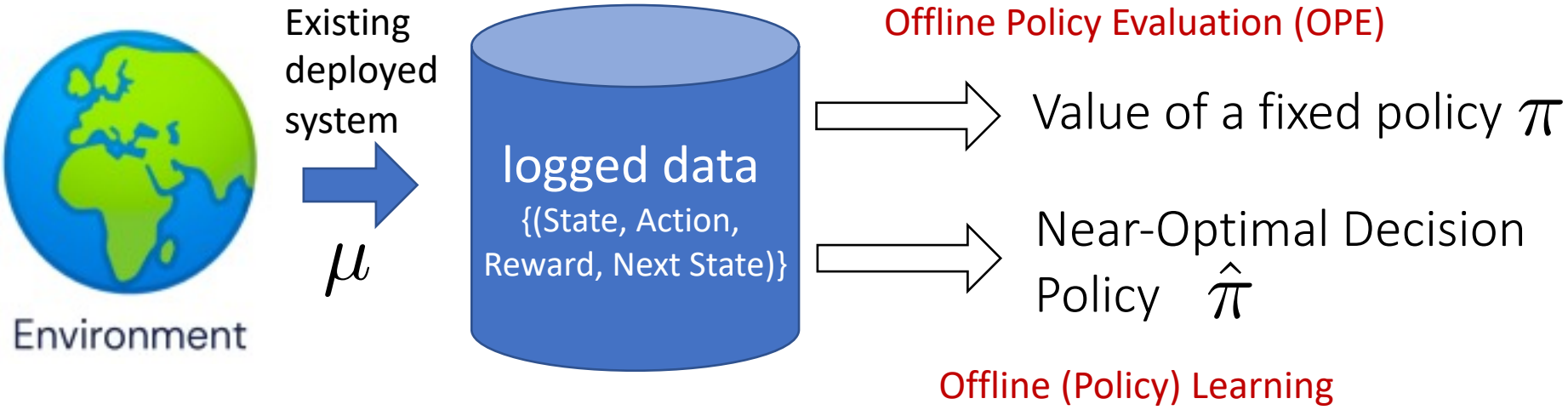
In real-life applications, we have limited access to the environment.



- Exploration is often costly, unsafe, illegal, …
- "Drive off road and crash the car to learn it's a bad idea"

RL in practice always starts with an existing dataset => Offline RL

- Two typical tasks are: OPE and Offline Learning



*Notation: $v^\pi := \mathbb{E}_\pi[\text{Total Reward}]$ $\quad$ Optimal policy $\pi^* := \arg\max_\pi v^\pi$

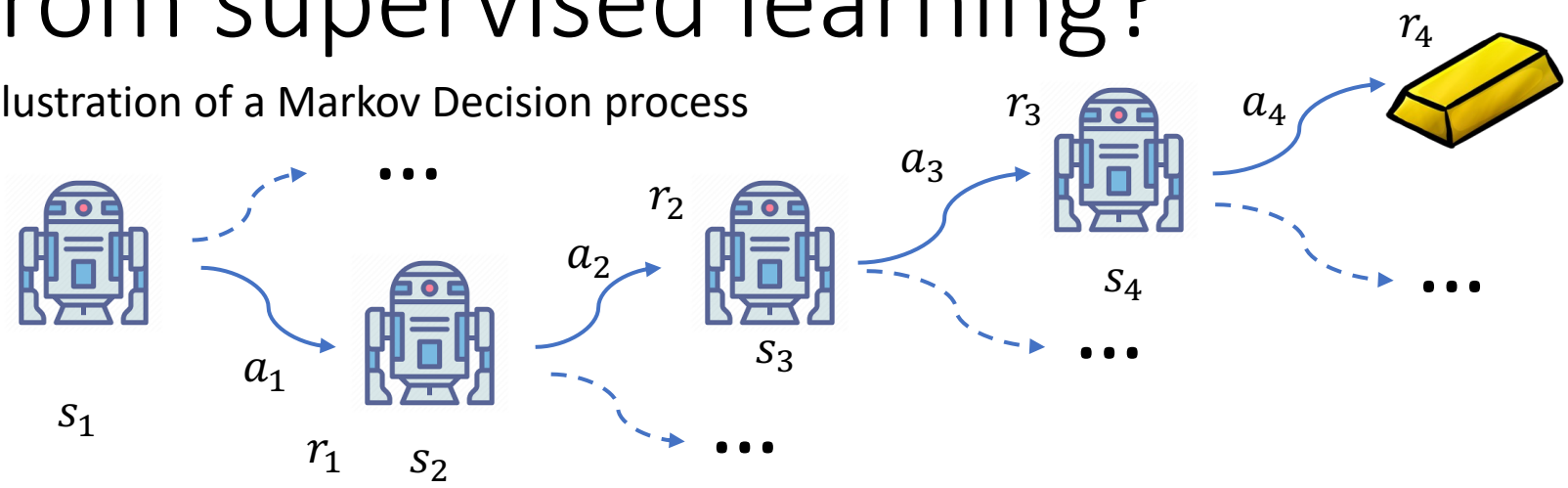
Applications in Healthcare

- Learning / Improving Personalized Treatment Plan
  - State: Measurements
  - Action: Medication
  - Reward: Surviving / reduced tumor size

- Health Monitoring with Wearable Devices Data
  - State: Heart rate, step count, sleep time, …
  - Action: Get a test or wait
  - Reward: How early is the disease detected

What makes offline RL different from supervised learning?

Illustration of a Markov Decision process



	Supervised Learning	Offline Reinforcement Learning
Evaluation	$\frac{1}{n}\sum_{i=1:n} \ell(y_i, \pi(x_i))$	Why is OPE challenging? Try it!
Learning	ERM + Uniform Convergence	"ERM = Max OPE" ?

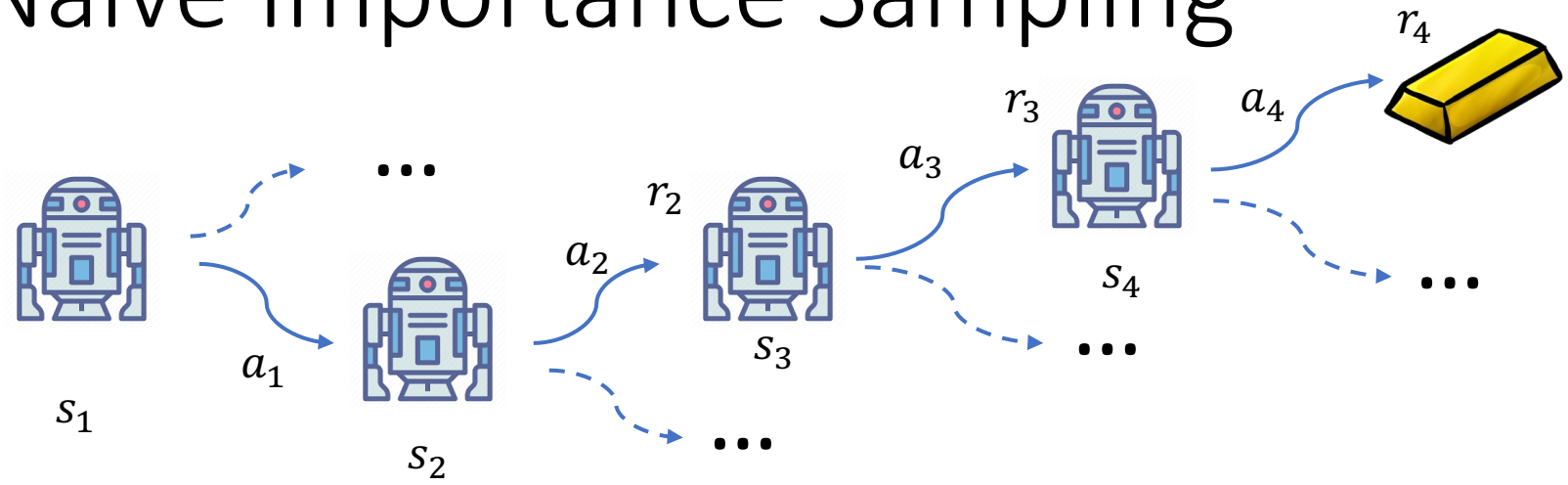
Even OPE is challenging because new policies change the states to visit!

Offline RL is fundamentally a causal inference problem with observational data.

- OPE ⇔ Avg Treatment Effect Estimation
- Offline Learning ⇔ Selecting the Best Treatment

- Main differences:
  - RL assumes a model "Markov Decision Process", which assumes away ignorability.
  - RL involves H rounds of decisions.

Naïve Importance Sampling



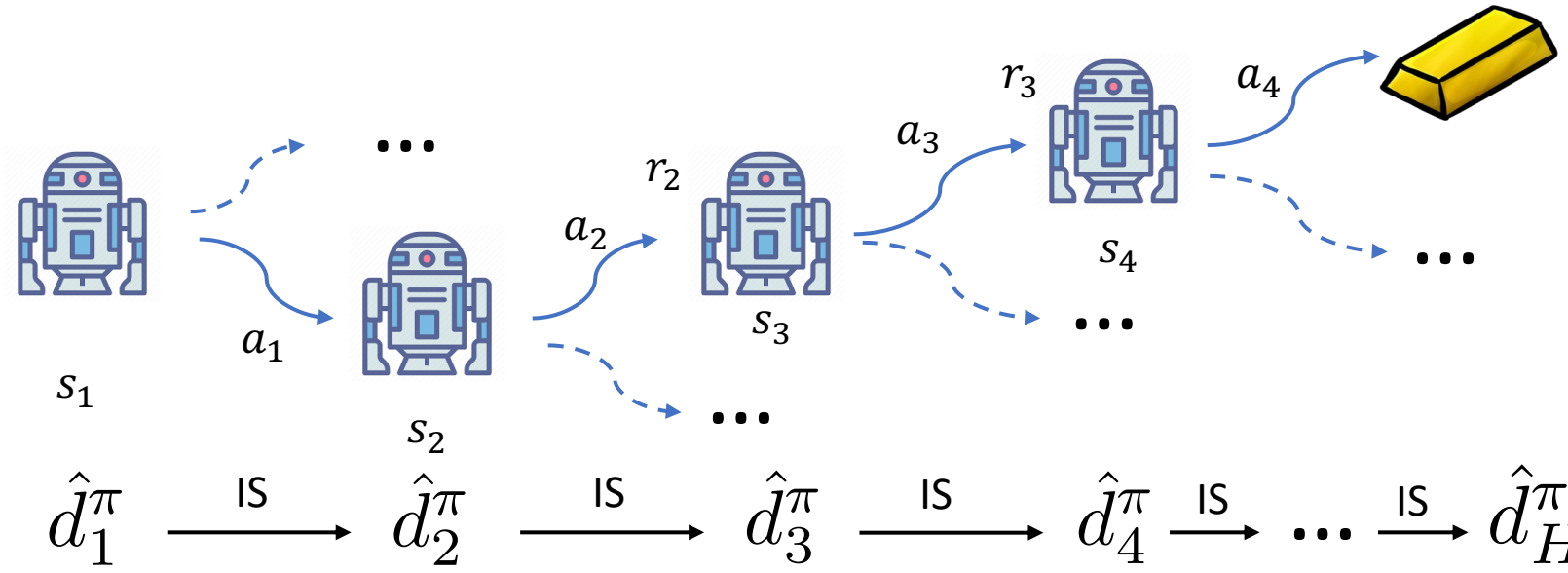
An S States, A actions, H Step Markov Decision process

$$\hat{v}^{\pi} = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{H} w_t^{(i)} r_t^{(i)}$$

Sutton/Barto uses $w_t = \dfrac{\pi(a_1|s_1)\pi(a_2|s_2)\ldots\pi(a_t|s_t)}{\mu(a_1|s_1)\mu(a_2|s_2)\ldots\mu(a_t|s_t)}$

Suffers "Curse of Horizon"

Marginalized Importance Sampling

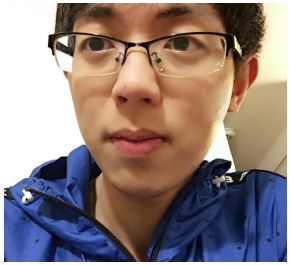


Sequentially estimating the induced marginal state-action visitation

- MIS uses weight $\frac{\hat{d}_t^\pi(s_t, a_t)}{\hat{d}_t^\mu(s_t, a_t)}$ to avoid the curse of horizon.

(Xie, Ma & W., NeurIPS'2019; Yin & W., AISTATS'2020)

Our results on offline RL



Ming Yin

- Optimal {OPE, uniform OPE, offline RL, offline reward-free RL, offline RL with linear function approx…}

Optimal bound for OPE

$$\mathbb{E}[(\hat{v}^{\pi}_{\mathrm{TMIS}} - v^{\pi})^2] \le \frac{1}{n}\sum_{h=0}^{H}\sum_{s_h,a_h}\frac{d_h^{\pi}(s_h)^2}{d_h^{\mu}(s_h)}\frac{\pi(a_h|s_h)^2}{\mu(a_h|s_h)}\cdot\mathrm{Var}\left[(V_{h+1}^{\pi}(s_{h+1}^{(1)}) + r_h^{(1)})\middle| s_h^{(1)} = s_h, a_h^{(1)} = a_h\right] + O(n^{-1.5})$$

Or if in a simplified expression: $\left|\hat{v}^{\pi}_{\mathrm{TMIS}} - v^{\pi}\right| \asymp \sqrt{\frac{H^2}{n\, d_m^{\mu}}}$

(Xie, Ma & W., NeurIPS'19)

(Yin & W., AISTATS-20)

Optimal bound for Offline Learning via local Uniform OPE

$$\hat{\pi} = \arg\max_{\pi\in\Pi} \hat{v}^{\pi}_{\mathrm{TMIS}}$$

$$v^{\pi^*} - v^{\hat{\pi}} \lesssim \sqrt{\frac{H^3}{n\, d_m^{\mu}}}$$

(Yin, Bai & W., AISTATS'21)

*Uniform coverage condition: $d_m^{\mu} := \min_{t,s,a} d_t^{\mu}(s,a)$

Per-instance optimal offline learning?

Ming Yin

Results under different exploration assumptions and special properties of the MDPs.

Uniform Visitation

$$\widetilde{O}\left(\sqrt{\frac{H^3}{n \cdot d_m}}\right)$$

(Yin, Bai & W., 2021)

Single Concentrability

$$\widetilde{O}\left(\sqrt{\frac{H^3 S C^*}{n}}\right)$$

(RZMJR 2021)

Adaptive Domain

$$\widetilde{O}\left(\sum_{h=1}^{H}\sqrt{\frac{\mathbb{Q}_h^*}{n \cdot d_m}}\right) + \widetilde{O}\left(\frac{H^3}{n \cdot d_m}\right)$$

(Zanette and Brunskill, 2019)

Per-instance optimal offline learning?

Ming Yin

"Pessimism is all you need" (Yin and W., NeurIPS-21)

Intrinsic Offline Learning Bound

$$\sum_{h=1}^{H}\sum_{s_h,a_h} d_h^{\pi^*}(s_h,a_h)\sqrt{\frac{\mathrm{Var}_{P_{s_h,a_h}}(V_{h+1}^*+r_h)}{d_h^{\mu}(s_h,a_h)}}\cdot\sqrt{\frac{1}{n}} \quad +\tilde{O}(1/n)$$

Uniform Visitation

$$\tilde{O}\left(\sqrt{\frac{H^3}{n\cdot d_m}}\right)$$

(Yin, Bai & W., 2021)

Single Concentrability

$$\tilde{O}\left(\sqrt{\frac{H^3SC^*}{n}}\right)$$

(RZMJR 2021)

Adaptive Domain

$$\tilde{O}\left(\sum_{h=1}^{H}\sqrt{\frac{\mathbb{Q}_h^*}{n\cdot d_m}}\right)+\tilde{O}\left(\frac{H^3}{n\cdot d_m}\right)$$

(Zanette and Brunskill, 2019)

Strongest (most adaptive) result in offline RL to date!

(Extension to linear function approximation, ICLR'22;
for representation learning, in the pipeline)

What if the optimal policy visits states never seen in the data?

- Lazy answer: "Optimal policy not measurable"

- "Maybe we could still learn something?"

$$v^{\hat{\pi}} \leq \min_{\pi} v^{\pi} + \tilde{O}\left(\sum_{h=1}^{H} \sum_{s_h, a_h} d_h^{\pi}(s_h, a_h) \sqrt{\frac{\mathrm{Var}_{P_{s_h,a_h}}(V_{h+1}^{\pi} + r_h)}{d_h^{\mu}(s_h, a_h)}} \cdot \sqrt{\frac{1}{n}} \right)$$

Pessimism VI

Arbitrary comparator policy

Regret / performance difference

" Learn as much as we can. Identify the best policy identifiable! "

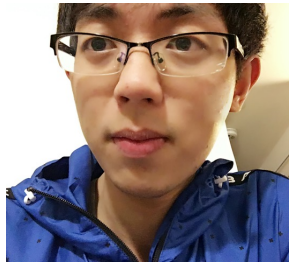
MIS and Pessimism are used in the empirical side of offline RL too!

- Marginalized importance sampling
  - Overcome the Curse of Horizon by explicitly estimating the state-distribution induced by each policy.
  - Correcting the distribution-shift by reweighting
  - DICE family of methods. (Nachum, Dai, et al.)

- Pessimism in Value iterations / Q-Learning
  - Very popular in deep RL
  - Various ways of implementing pessimism is the SOTA in applied offline RL

  (see Levine et al.)

Checkpoint: Learning to Act with Offline RL



Ming Yin

- A perspective that is largely missing from classical statistical learning theory

- We built a theoretical foundation for offline RL
  - OPE (ICML'19, AISTATS'20)
  - Uniform OPE (AISTATS'21, NeurIPS'21)
  - Offline Learning (NeurIPS'21 * 2, ICLR'22)

- Extensions to various settings: function approximation, representation learning, reward-free case, etc…

- Still a very young field.
  - A lot of opportunities of new theory / applications

Online RL vs Offline RL, revisited

	Online RL	Offline RL
Sample Complexity	$\tilde{O}\left(\frac{H^3SA}{\epsilon^2}\right)$	$\tilde{O}\left(\frac{H^3}{\epsilon^2 d_m}\right)$ or "Best effort learning" when d_m too small

Algorithmically enforce "Good Exploration"

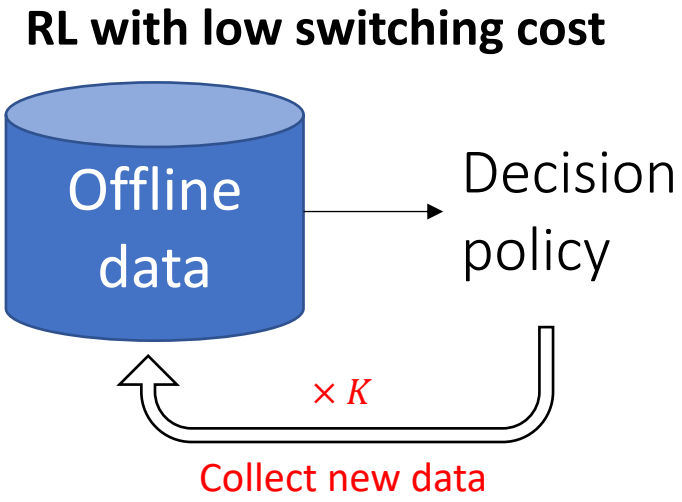
Assume "Good Exploration" or **weaken goal**

T rounds of adaptivity.
One per iteration!

1 round of adaptivity.

Anything in between?

Emerging new setting between online and offline RL



Can we solve exploration with a small number of policy changes?

Near optimal regret / sample complexity, but with:

K = O(log T) (Bai, Xie, Jiang, W., NeurIPS'19)

K = O(log log T) (Qiao, Yin, Yin and W., arxiv 2022)

Remainder of the talk

1. Controlling the Nonstationarity with offline reinforcement learning

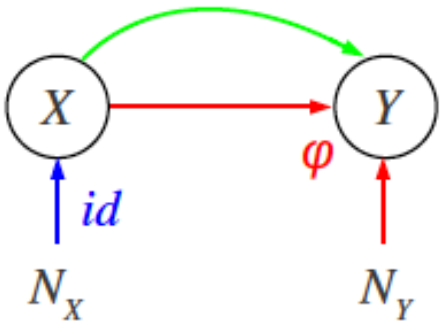
2. Tracking unknown nonstationarity with dynamic regret minimization

Existing methods for handling nonstationarity often make strong assumptions about the world.

Covariate Shift

$$q(\boldsymbol{x}, y) = q(\boldsymbol{x})p(y|\boldsymbol{x})$$

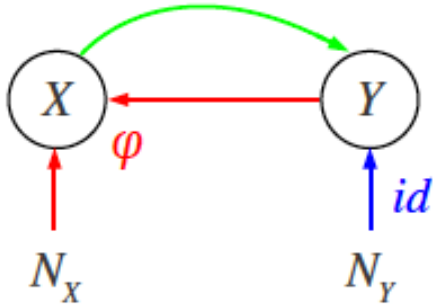
Causal Learning



Label Shift

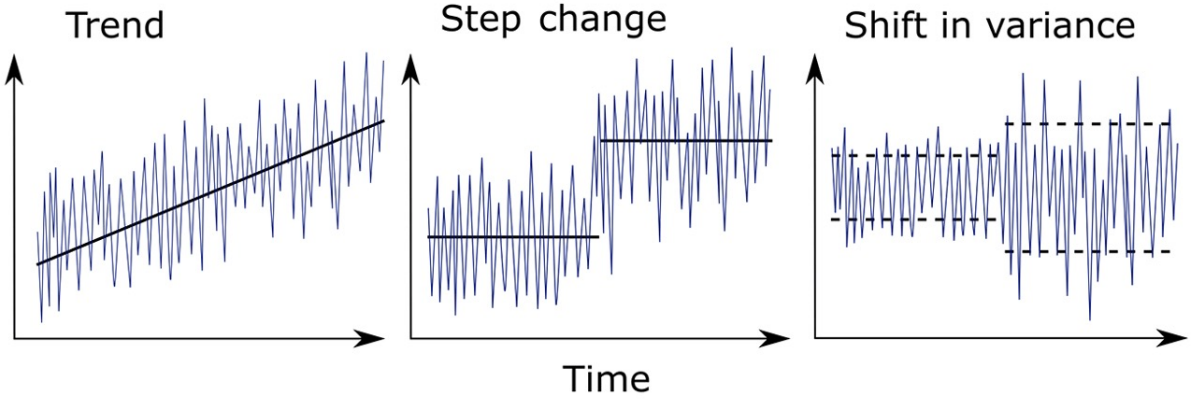
$$q(\boldsymbol{x}, y) = q(y)p(\boldsymbol{x}|y)$$

Anti-causal Learning



Concept Shift changes φ

Non-stationarity in practice is continuously happening and is a mix of all kinds of shifts at once.



Can we handle nonstationarity without modeling the world? Yes, by Dynamic Regret Minimization

The Online Learning setting

- For each $t \in [n] := \{1, \ldots, n\}$, learner predicts $\boldsymbol{x}_t \in \mathcal{D} \subset \mathbb{R}^d$.
- Adversary reveals a loss function $f_t : \mathbb{R}^d \to \mathbb{R}$

Example: $f_t(x) = (\mathrm{StockPrice}_t - \mathrm{Feature}_t^T x)^2$

Goal: Learner aims to control its dynamic regret against any sequence of comparators $\boldsymbol{w}_1, \ldots \boldsymbol{w}_n$ where $\boldsymbol{w}_t \in \mathcal{W} \subseteq \mathcal{D}$ for all t.

$$R_n(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n) := \sum_{t=1}^{n} f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{w}_t),$$

Dynamic regret is parameterized by the total variation of the comparator sequence

$$C_n(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n) = \sum_{t=1}^{n} \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|_1$$

Dheeraj Baby

Theorem (Baby and W., 2021)

For exp-concave losses, there is an efficient online algorithm, s.t.

$$\underbrace{\sum_{t=1}^{n} f_t(x_t)}_{\text{Our performance}} \leq \min_{w_1,\ldots,w_n} \underbrace{\sum_{t=1}^{n} f_t(w_t)}_{\text{Comparator performance}} + \underbrace{O(n^{1/3} C_n(w_1, \ldots, w_n)^{2/3})}_{\text{Dynamic regret}}$$

- Solves a problem opened for 17 years since Zinkevich (2003)
- COLT'21 Best Student Paper award
- Technically interesting and novel.

This is a change of paradigm in how we handle non-stationarity

- Covariate shift / label shift / Concept shift
  - Measure the differences in how much the distribution has changed
  - Need data points from target distributions

- We measure non-stationarity in **how much our model need to change** to predict well.

Why is this new paradigm powerful?

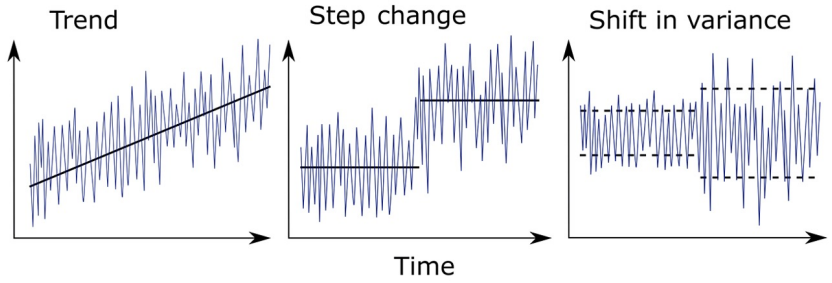
- It is fully agnostic and it does not make assumptions about the type of non-stationarity

Covariate Shift

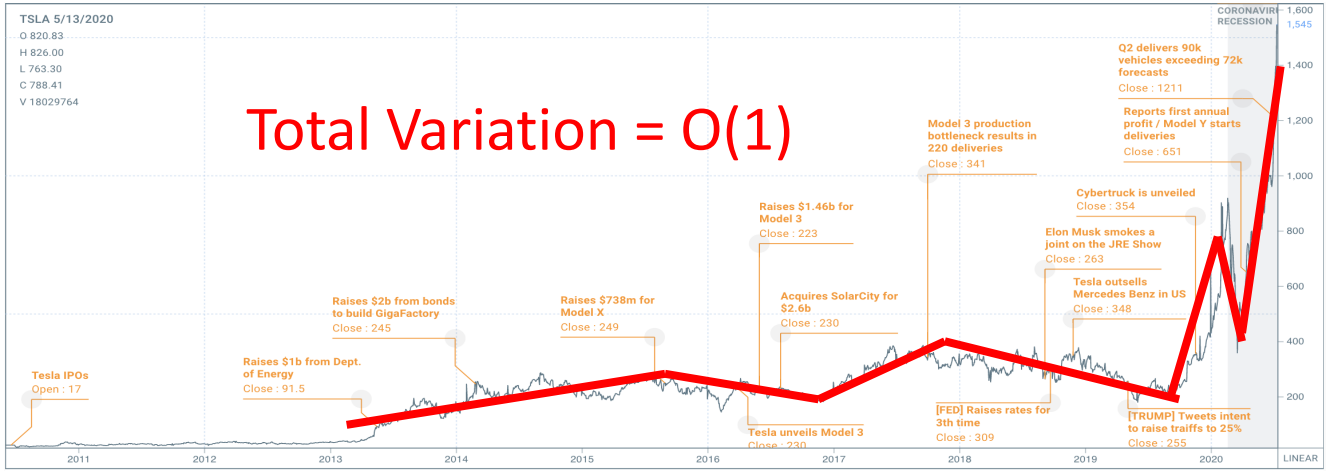
$$q(\boldsymbol{x}, y) = q(\boldsymbol{x})p(y|\boldsymbol{x})$$

Label Shift

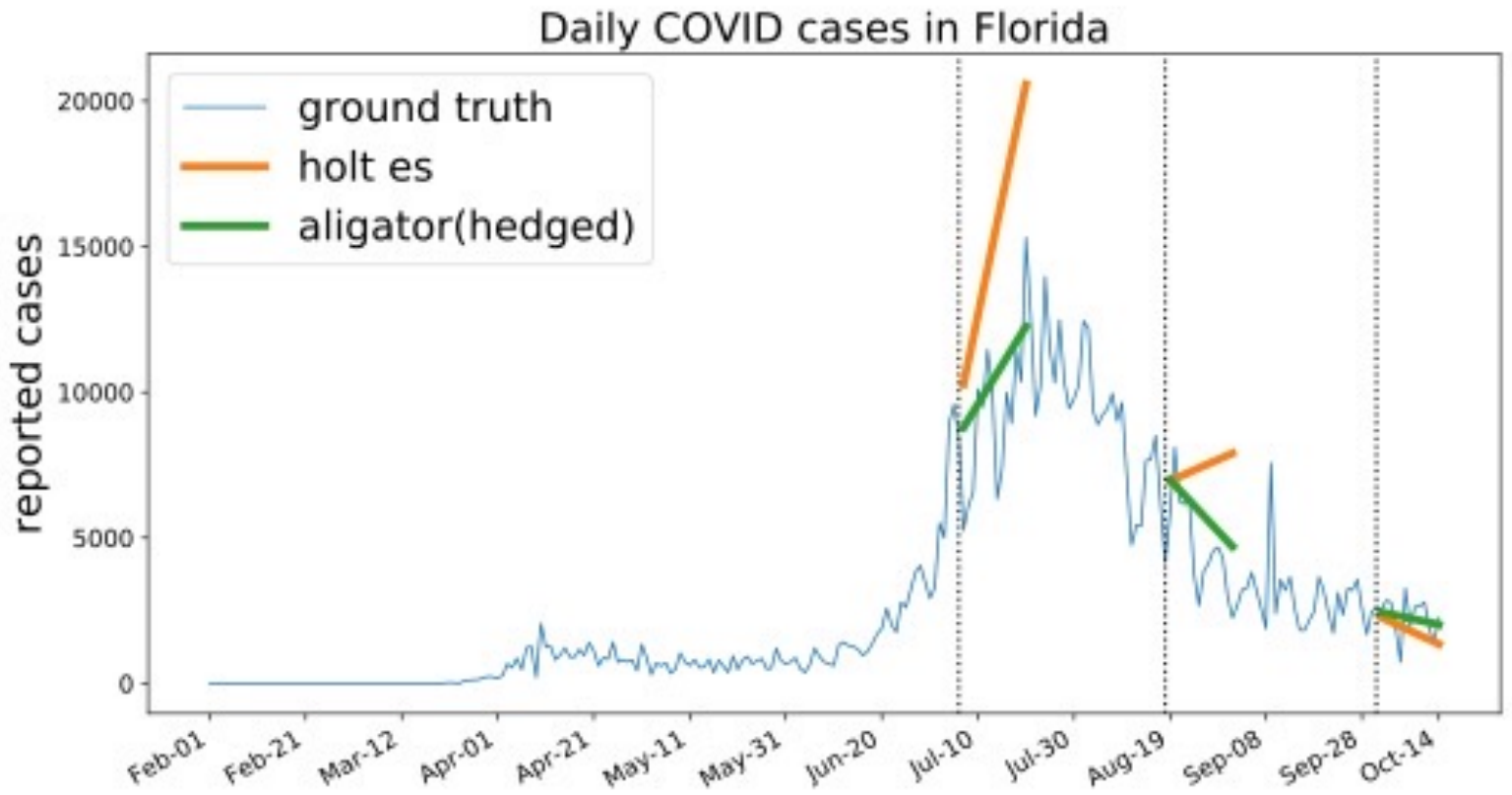
$$q(\boldsymbol{x}, y) = q(y)p(\boldsymbol{x}|y)$$



- Optimally compete with your favorite sequence chosen **in hindsight**



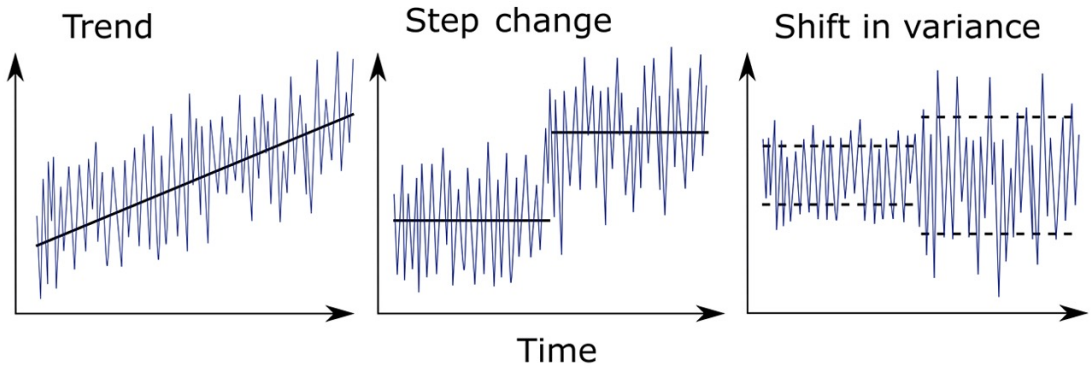
Application to “Online Trend Removal” in COVID hospitalization forecasting



(Baby, Zhao and W., AISTATS’21)

So what is the algorithm?

- Key intuition: How much past data to use?



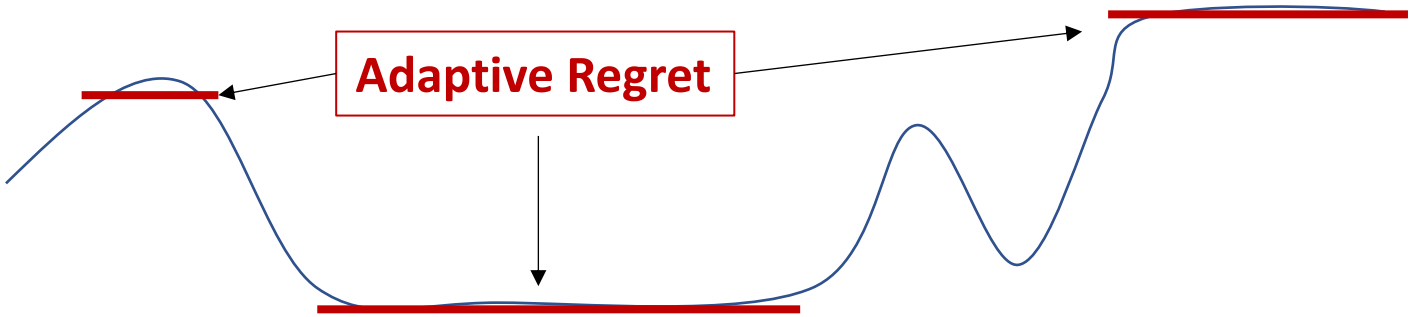
- Why not use all window sizes? **Start a new learner every day** and "Hedge" over them with an ensemble meta-learner.

- Computational / memory constraint? Use a geometric cover: O(n^2) ➔ O(n log n) time, O(n) ➔ O(log n) space

Proof highlights: Adaptive Regret and Strongly Adaptive Online Learner

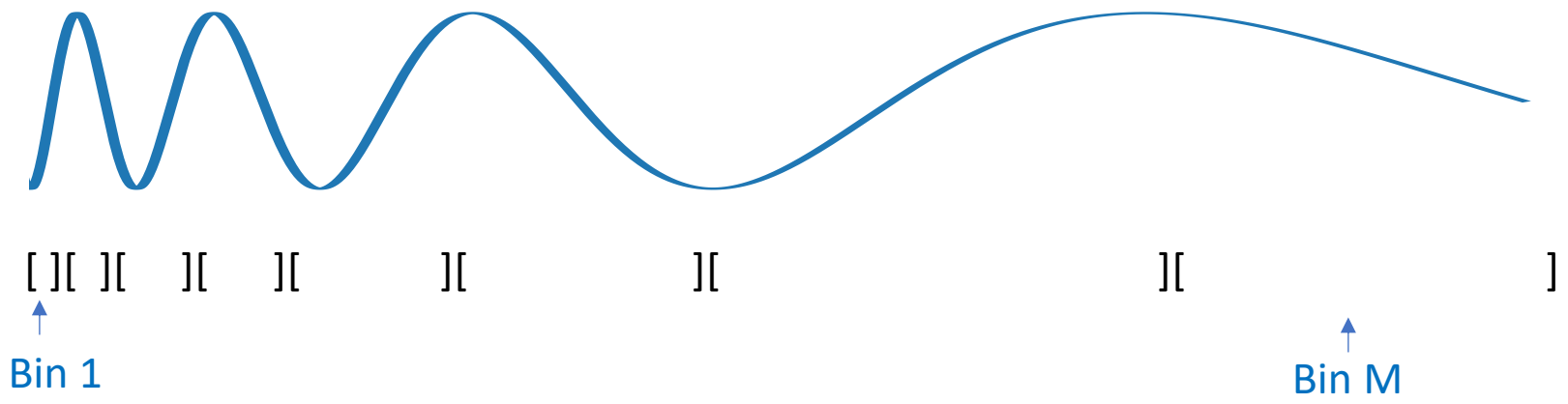
- Adaptive Regret Minimization (Hazan and Seshadhri, 2009) (Daniely, Gonen, Shalev-Shwartz, 2015)
  - Follow the Leading History (FLH)

- Our algorithm: FLH with Online-Newton-Step
- For exp-concave losses, FLH-ONS achieves an $\tilde{O}(1)$ **static regret** of on **all intervals** at the same time!



Proof highlights: Adaptive Partition

- Let the following be the offline optimal comparator



We construct a partitioning of $[n]$ into M bins as follows $\{[1_s, 1_t], \ldots, [i_s, i_t], \ldots, [M_s, M_t]\}$ satisfying:

- $C_i := \sum_{j=i_s}^{i_t-1} |u_{j+1} - u_j| \leq B/\sqrt{n_i}$ where $n_i := i_t - i_s + 1$, $i \in [M]$.
- Number of bins obeys $M = O(n^{1/3} C_n^{2/3} B^{-2/3} \vee 1)$.

Suffices to prove the **dynamic regret** in each bin is $\tilde{O}(1)$.

Proof highlights: Regret Decomposition

One-step Gradient Descent

$$R_n(C_n) \le \sum_{i=1}^{M}\sum_{t=i_s}^{i_t} \underbrace{f_t(x_t) - f_t\Big(\bar{u}_i - \eta\nabla \sum_{t'=i_s}^{i_t} f_{t'}(\bar{u}_i)\Big)}_{T_{1,i}}$$

By Strong Adaptivity $T_{1,i} = O(B^2 \log n)$.

$$+\sum_{i=1}^{M}\sum_{t=i_s}^{i_t} \underbrace{f_t\Big(\bar{u}_i - \eta\nabla \sum_{t'=i_s}^{i_t} f_{t'}(\bar{u}_i)\Big) - f_t(\bar{u}_i)}_{T_{2,i}}$$

By **Descent Lemma** $T_{2,i} \le -\frac{\eta}{2}\|\nabla\|^2$

$$+\sum_{i=1}^{M}\sum_{t=i_s}^{i_t} \underbrace{f_t(\bar{u}_i) - f_t(u_t)}_{T_{3,i}}$$

By KKT conditions

$$T_{3,i} \le n_i C_i^2 + 3\lambda C_i$$
$$\le B^2 + 3\lambda C_i,$$

* $T_{2,i}$ is not always strictly negative. $T_{3,i}$ is often very large. Turns out that there is a **magical refinement of the partition** such that $T_{2,i}$ is sufficiently negative when we need it be.

** The first time KKT conditions across time-steps are exploited in online learning.

Checkpoint: Harnessing Nonstationarity by Dynamic Regret Minimization

Dheeraj Baby

- Timeline of the research
  - NeurIPS'19 First ever O(n^1/3) dynamic regret for square loss in stochastic setting
  - NeurIPS'20 O(n^{1/(2k+3)}) higher-order case "Online Trend Filtering"
  - COLT'21: O(n^1/3) *universal* dynamic regret for exp-concave losses in full adversarial setting **(COLT Best Student Paper)**
  - AISTATS'22: From Improper Learning to Proper learning
  - In the pipeline: O(n^1/5) universal dynamic regret for TV1 for exp-concave losses in full adversarial setting

- Intersections with time series forecasting, nonstochastic control, reinforcement learning, pricing and so on…

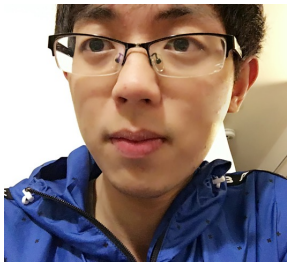
Take home messages

- Two types of nonstationarity
- Explicitly modeling how my new policy will change the distribution using offline RL
  - Staying “pessimistic” is the key
- New paradigm in handling unknown nonstationarity over time.
- Promising applications in healthcare.

Thank you for your attention!

Talk based on the work of UCSB PhD students:

Dheeraj Baby



Ming Yin

and contributions from many other collaborators!

UCSB Machine Learning Lab

CRML Center for Responsible Machine Learning

Our research is partially supported by:

NSF aws Adobe Google evidation NEC appfolio