Introduction to online learning and multi-armed bandits

Kwang-Sung Jun

U of Arizona

01/15/2020

Overview

- Introduce the key problems and their applications
- Course overview
- A toy online learning problem as a beginner (whiteboard only)

What is online learning?

Example: online linear regression

For time $t = 1 \dots T$

- The adversary sets the features $z_t \in \mathbb{R}^d$ and the label $y_t \in \mathbb{R}$, but keeps it secret.
- The player outputs linear coefficients $x_t \in \mathbb{R}^d$.
- The player receives z_t and y_t ; incurs loss $\ell_t(x_t) = (z_t^\top x_t - y_t)^2$.

No stochastic assumptions on how the data is generated

After T iterations, we measure the performance by

$$\text{Regret}_T = \sum_{t=1}^{T} \ell_t(x_t) - \min_x \sum_{t=1}^{T} \ell_t(x)$$

the oracle performance

What is online learning?

$$\mathrm{Regret}_T = \sum_{t=1}^{T} \ell_t(x_t) - \min_x \sum_{t=1}^{T} \ell_t(x)$$

- Ex 1. regression: $\ell_t(x) = (z_t^\top x - y_t)^2$
- Ex 2. classification: $\ell_t(x) = \max\{0, 1 - y_t z_t^\top x\}$ with $y_t \in \{-1,1\}$

	supervised learning	online learning
type	batch	sequential/incremental
data assumption	i.i.d.	adversarial (no stochastic assumption whatsoever)
performance metric	offline test error	online test error (regret)

- For regression, $\mathrm{Regret}_T = O(d \log T)$ is possible
- But, how is it possible??

Regret bounds

- The goal: sublinear regret bounds
  - e.g., $\mathrm{O}\left(\sqrt{T}\right), O\left(T^{\frac{3}{4}}\right), \mathrm{O}(\log(T))$, but not $\mathrm{O}(T)$

- Why is sublinear interesting?
  - The average regret = $\frac{\mathrm{Regret}_T}{T} = \frac{1}{T}\sum_{t=1}^{T} \ell_t(x_t) - \min_x \frac{1}{T}\sum_{t=1}^{T} \ell_t(x)$ goes to 0 as $T \to \infty$
  - Eventually, we perform as good as the oracle!
  - Regret determines the “rate” of convergence to the oracle.

Online learning (stylized)

For time $t = 1 \dots T$

- The adversary sets the loss function $\ell_t : \mathbb{R}^d \to \mathbb{R}$ but keeps it secret.
- The player outputs $x_t \in \mathbb{R}^d$
- The player receives ℓ_t and incurs loss $\ell_t(x_t)$

When the losses are convex, we call it online convex optimization (OCO)

Applications

- Online learning for the batch supervised learning
  - For extremely large scale datasets, online learning is the only viable option for training a model.
    - Vowpal Wabbit: online learning software.
  - Theoretical property: With the online-to-batch conversion technique, we get "generalization error" guarantee!

- Online learning for streaming data
  - In streaming data, we expect the concept to change over time. (e.g., twitter)
  - Algorithms with "shifting" regret guarantee => adapts to the environmental change in a provable way!
  - $\text{ShiftingRegret}_T = \sum_{t=1}^{T} \ell_t(x_t) - \min_{w_1,\ldots,w_T:\ \text{switches } K \text{ times}} \sum_{t=1}^{T} \ell_t(w_t)$

Application: online convex optimization (OCO)

- The loss function is any convex function.
- Stochastic gradient descent without tuning the step size*
  - SGD can be analyzed in the online learning framework + online-to-batch tech.
  - Convergence to the optimal model w^* : $\left(\frac{||\, w^* \,||^2}{\eta} + \eta\right)\frac{1}{\sqrt{T}}$
  - The optimal tuning $\eta = ||\, w^* \,||$ requires to know the unknown..!
  - In practice, people tune η with repeated runs.
  - COCOB algorithm by Orabona&Tommasi achieves the optimal convergence up to logarithmic factors!
    - There is no need to tune η.
  - It is quite effective in training deep networks.

*Orabona and Tommasi, Training Deep Networks without Learning Rates Through Coin Betting, NeurIPS'17.

Application: portfolio management

- d given stocks, W_0 := initial wealth
- On t-th day, buy stocks with the current wealth W_{t-1} in the morning, sell them at the end of the day.
  - $p_t \in \Delta_d$: the distribution over the stocks; To stock i, you invest $W_{t-1}p_{t,i}$
  - Let $r_t \in (0,\infty)^d$ be the price change ratio.
- Then, $W_T = W_0 \prod_{t=1}^{T} p_t^\top r_t$
- Goal: Maximize "log" wealth, competing with "constant rebalanced portfolio" (CRP) strategy.

$$\text{Regret}_T = \max_{p \in \Delta_d} \sum_{t=1}^{T} \log(p^\top r_t) - \sum_{t=1}^{T} \log(p_t^\top r_t)$$

  - Exists a strategy with O(d logT) regret bound, under mild assumptions.

(Agarwal et al., Algorithms for Portfolio Management based on the Newton Method, 2006)

Other applications/extensions

- Caching policy for reducing server load
  (Paschos et al., Learning to Cache With No Regrets, 2019)
- Online learning under changing environments (compete with switching comparator)
  (Daniely et al., Strong adaptive online learning, 2015)
- Online meta-learning
  (Finn et al., online meta-learning, 2019)

Learning with expert advice

For time $t = 1 \ldots T$

- The adversary sets the loss function $\ell_t: \mathbb{R}^d \to \mathbb{R}$ but keeps it secret.
- The player predicts x_t
- The player receives ℓ_t and incurs loss $\ell_t(x_t)$

$$\ell_t(x) = g_t^\top x, \qquad g_t \in [0,1]^d$$
$$x_t \in \{e_1, \ldots, e_d\} \qquad \text{(one hot vectors)}$$

- E.g., predict raining or not.
  - d expert opinions everyday (0/1).
  - Everyday, choose expert $I_t \in \{1, \ldots, d\}$; follow her opinion.
  - After T days, how many mistakes do we make? (relative to the best expert in hindsight)

Learning with expert advice

$$\ell_t(x) = g_t^\top x, \qquad g_t \in [0,1]^d$$
$$x_t \in \{e_1, \dots, e_d\}$$

- I_t = the player's choice.
- $\text{Regret}_T = \mathbb{E}\left[\sum_{t=1}^T g_{t,I_t}\right] - \min_{i\in\{1,\dots,d\}} \sum_{t=1}^T g_{t,i}$
- The optimal rate: $O\left(\sqrt{T\log(d)}\right)$
- Application
  - Online model selection among multiple classifiers
  - Or, SGD with multiple learning rates => perform as well as the best learning rate in hindsight.

Bandit feedback

For time $t = 1 \dots T$

- The adversary sets the loss function $\ell_t: \mathbb{R}^d \to \mathbb{R}$ but keeps it secret.
- The player predicts x_t
- The player ~~receives ℓ_t~~ and incurs loss $\ell_t(x_t)$

receives g_{t,I_t} Learns the loss of the chosen expert, not the others!

$$\ell_t(x) = g_t^\top x, \qquad g_t \in [0,1]^d$$
$$x_t \in \{e_1, \dots, e_d\} \qquad \text{(one hot vectors)}$$

Optimal regret turns out to be $O\left(\sqrt{dT}\right)$ rather than $O\left(\sqrt{T\log(d)}\right)$

Learning with expert advice + bandit feedback = the (adversarial) multi-armed bandit problem

- A slot machine with d arms (=experts)
- Each arm gives you a different reward at each time: $r_{t,i} \in [0,1]$
  - these rewards are set ahead of time by an adversary
- You can play T times total
- Maximize rewards (w.r.t. any fixed decision throughout).

$$\text{Regret}_T = \max_{i \in \{1,\ldots,d\}} \sum_{t=1}^{T} r_{t,i} - \mathbb{E}\left[\sum_{t=1}^{T} r_{t,I_t}\right]$$

The (**stochastic**) multi-armed bandit problem

- A slot machine with d arms
- Each arm gives you a different reward at each time: $r_{t,i} \in [0,1]$
  - these rewards are **drawn from unknown distributions** $\nu_1, \ldots, \nu_d$
- Exploration vs exploitation dilemma.

- Maximize rewards (w.r.t. any fixed decision throughout).

$$\text{Regret}_T = \max_{i \in \{1,\ldots,d\}} \mathbb{E}\left[\sum_{t=1}^{T} r_{t,i}\right] - \mathbb{E}\left[\sum_{t=1}^{T} r_{t,I_t}\right]$$

- Studied much before online learning, dating back to 1933.
- Origin of this weird terminology “bandit”:
  - one-armed bandit = slot machine
  - generalized to **two**-armed bandit, then to **multi**-armed bandit.

The (**stochastic**) multi-armed bandit problem

- Faster rates available.
- Let $\mu_1, \dots, \mu_d$ be the expected reward of each arm.
- Define $i^* = \operatorname{argmax}_{i \in \{1,\dots,d\}} \mu_i$
- The optimal regret is

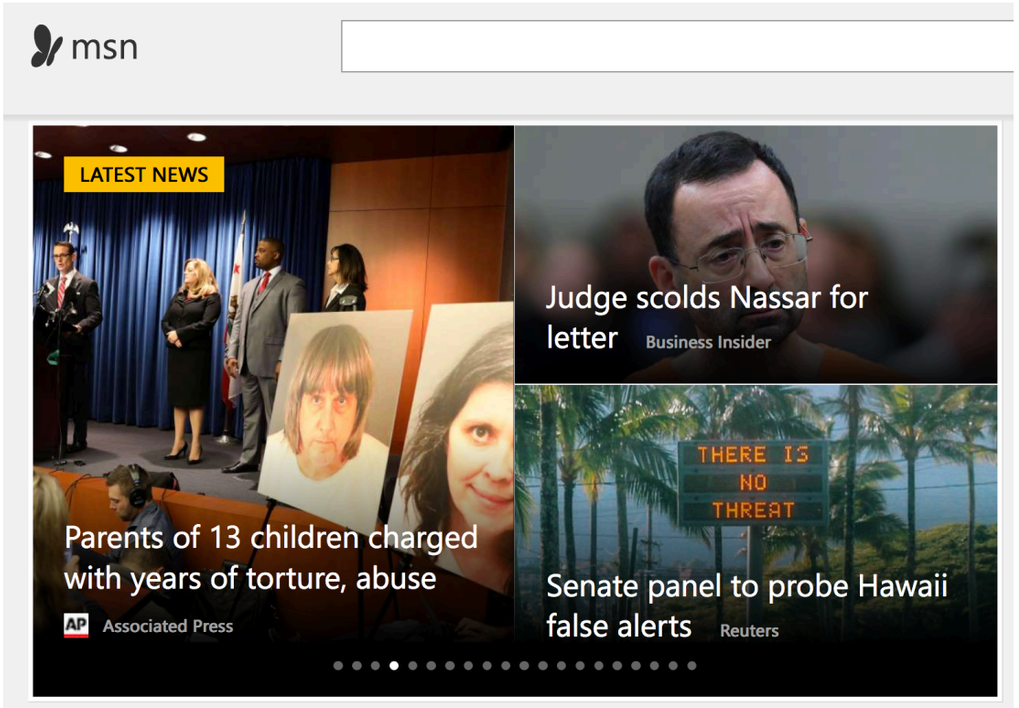
$$\mathrm{O}\left(\sum_{i \neq i^*} \frac{1}{(\mu_{i^*} - \mu_i)^2} \log T\right)$$

  - stark contrast with $O\left(\sqrt{dT}\right)$ regret from the adversarial version.

Applications: Online news recommendation

- Everyday, a set of news articles arrive
- ”top news” slot must be the most attractive one
- Every time a user arrives, pick one, and receive click feedback (=reward)
- Maximize the click through rate.

- More generally
  - product recommendation
  - online advertisement



Bandits with side-information

- Problem: The number of arms determines the difficulty of the problem.
- What if arms with the same colors give you has the same reward distribution?

- E.g., linear bandits
  - Features for i-th arm: $x^i \in \mathbb{R}^d$
  - Pulling arm i gives you reward $y = \theta^\top x^i + \eta$
    - θ : unknown parameter to learn.
    - η : zero-mean noise
- Recommendation system with a lot of items.

Contextual bandits

- At time t, a context information $c_t \in \mathbb{R}^{d'}$ is available, and the reward is affected by it.
  - e.g., features of the user being served.

- Combining with context and side-information:
  - Features for i-th arm: $x^i \in \mathbb{R}^d$
  - At time t, pulling arm i gives you reward $y = f(x^i, c_t) + \eta$
    - E.g., $f(x^i, c_t) = c_t^\top \Theta x^i$ with $\Theta \in \mathbb{R}^{d' \times d}$

- Application: PERSONALIZED recommendations

The “batch” bandit problem

- Recommendation system: N items, contextual setting
- Suppose we have been running algorithm A for a week
- Gathered dataset: $\{(c_t, x_t, y_t)\}_{t=1}^{T}$ // (context, item, reward)
  - “bandit-logged” data

- The problem: counterfactual evaluation
  - What rewards could we have received had we used another algorithm A’?
  - If randomized algorithm is used, there exists an **unbiased estimator**, but comes with a very large variance.
  - **The industry is obsessed to to find solution.**

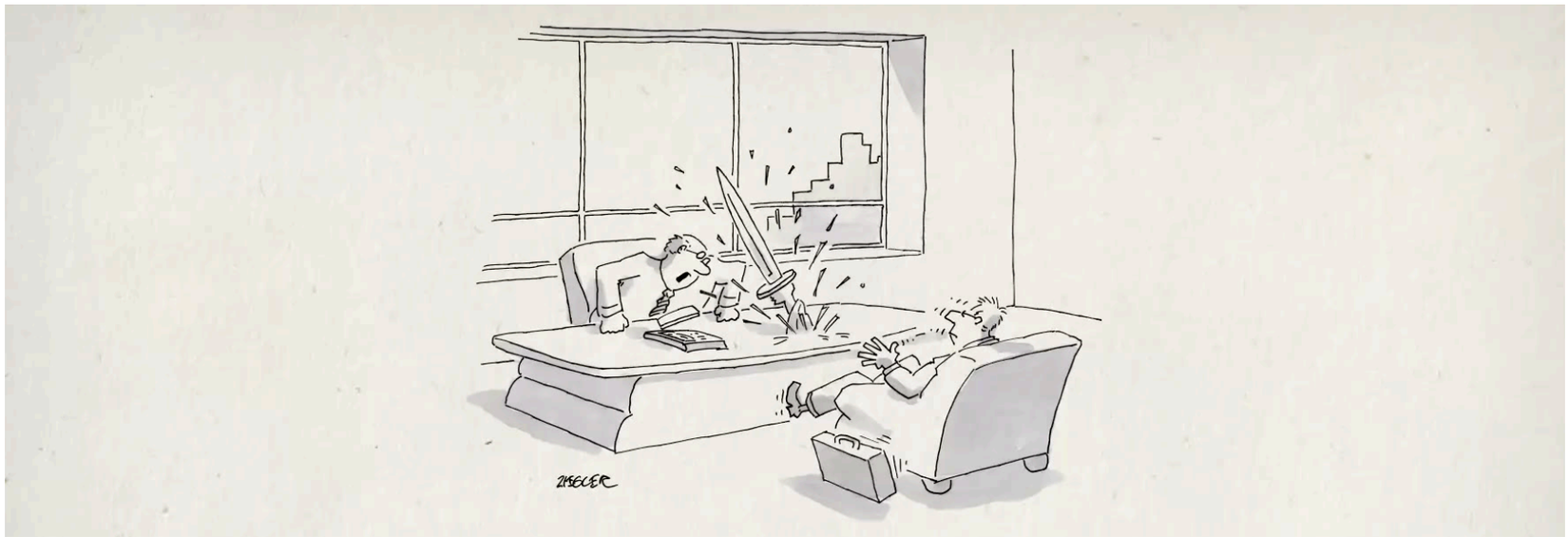
Pure exploration in multi-armed bandits

- What if we don't care about the rewards, but care about **identifying the best arm?**
- You go play the bandit game for T rounds, come tell me which one is the best arm.
- Performance measure: Let I be the claimed best arm

$$\text{minimize } \mathbb{P}(I \text{ is not the best arm})$$

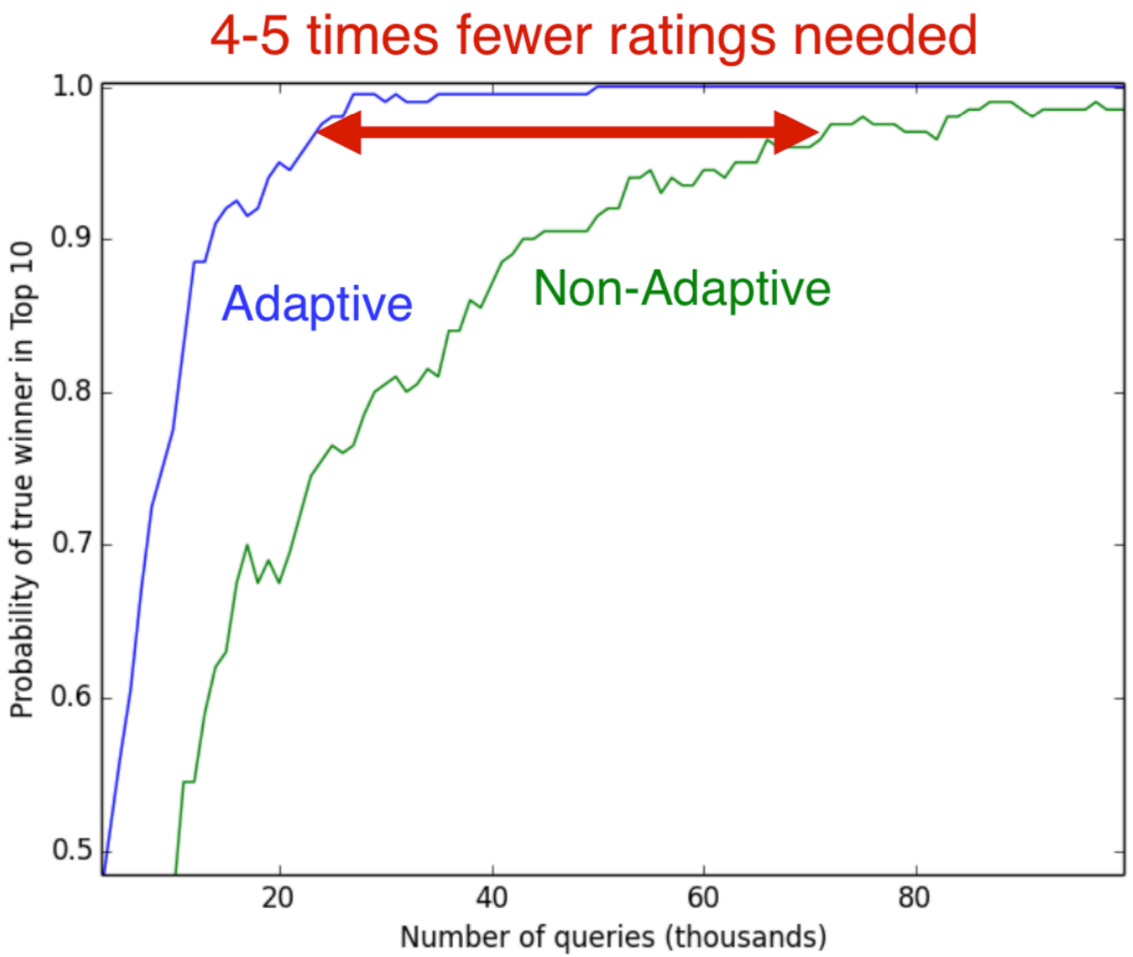
- Turns out, the strategy must be different from reward maximization.

Cartoon caption contest



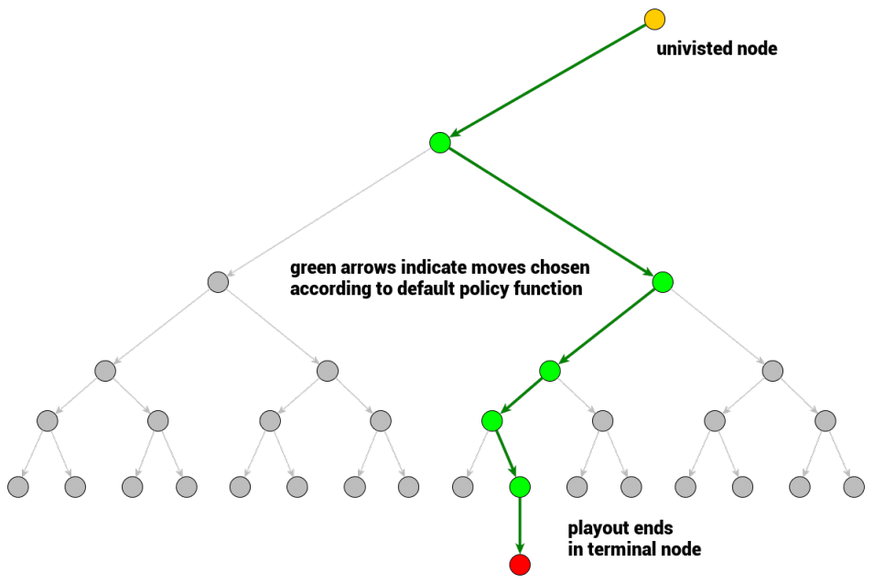
Cartoon caption contest

Cartoon caption contest



AlphaGo

Monte Carlo tree search
(main driver of AlphaGo besides deep learning)



arm = next move to make
reward = win(1) or lose(0) from that move

Bandit based Monte-Carlo Planning

Levente Kocsis and Csaba Szepesvári

Computer and Automation Research Institute of the
Hungarian Academy of Sciences, Kende u. 13-17, 1111 Budapest, Hungary
kocsis@sztaki.hu

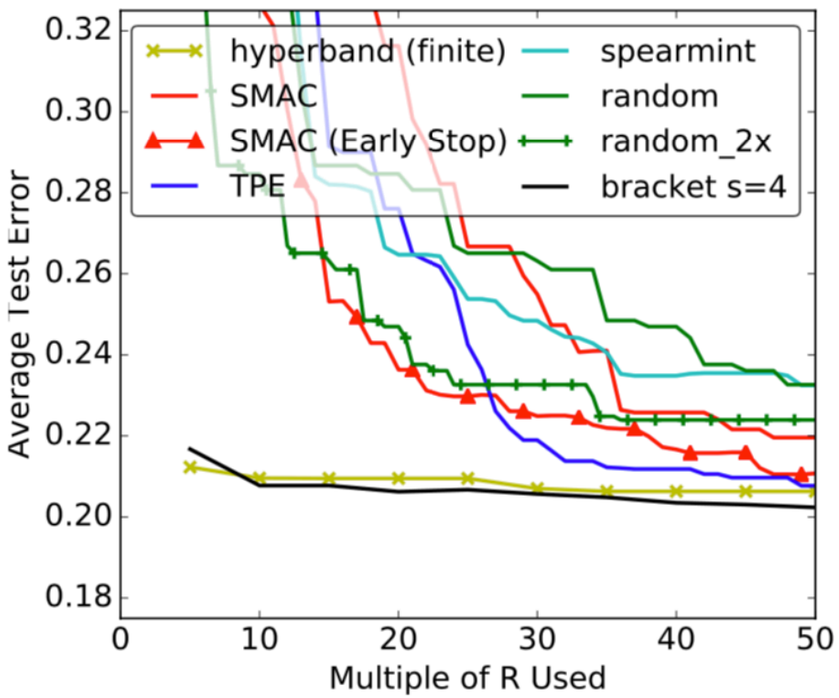
Abstract. For large state-space Markovian Decision Problems Monte-Carlo planning is one of the few viable approaches to find near-optimal solutions. In this paper we introduce a new algorithm, UCT, that applies bandit ideas to guide Monte-Carlo planning. In finite-horizon or discounted MDPs the algorithm is shown to be consistent and finite sample bounds are derived on the estimation error due to sampling. Experimental results show that in several domains, UCT is significantly more efficient than its alternatives.

- => bandits for reducing computational complexity!
- similar vein

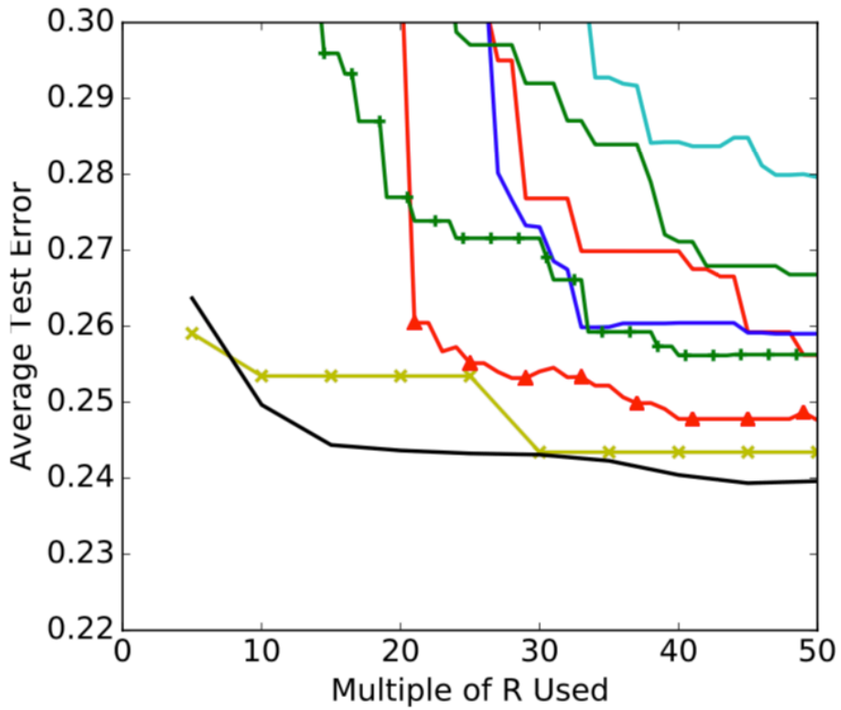
Bagaria, Vivek, et al. "Medoids in Almost-Linear Time via Multi-Armed Bandits." International Conference on Artificial Intelligence and Statistics. 2018.

Hyper-parameter optimization

Hyperband: Bandit-Based Configuration Evaluation for Hyperparameter Optimization,
Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, Ameet Talwalkar, *ICLR*, 2017.



(a) CIFAR-10



(b) MRBI

Course overview

Design choices

- Coherent topics
- Focus on ‘understanding’ the key concepts/techniques
- But far from a wide coverage
  - hopefully covered by student presentations

8 core OCO lectures

4 core stochastic bandits

4 more practical extensions of bandits

11 lecture slots for presentations

1: 01/15	Introduction to online learning and multi-armed bandits 1
2: 01/22	Introduction to online learning and multi-armed bandits 2
3: 01/27	Online gradient descent
4: 01/29	Subgradients and online-to-batch conversion
5: 02/03	Strong convexity
6: 02/05	Lower bounds for online linear optimization
7: 02/10	Online mirror descent 1
8: 02/12	Online mirror descent 2
9: 02/17	Follow-The-Regularized-Leader 1
10: 02/19	Follow-The-Regularized-Leader 2
11: 02/24	Adversarial multi-armed bandits (EXP3)
12: 02/26	Stochastic multi-armed bandits 1 (ETC, elimination)
13: 03/02	Stochastic multi-armed bandits 2 (UCB)
14: 03/04	Stochastic multi-armed bandits 3 (asymptotically optimal UCB)
15: 03/16	Lower bound on multi-armed bandits
16: 03/18	Contextual bandits
17: 03/23	Linear bandits
18: 03/25	Pure exploration
19: 03/30	Off-policy evaluation

Course overview

- What to expect
  - 2 homework assignments
  - One paper presentation
  - No final exam, no projects.
  - Every week, there will be a review quiz=> easy, but requires you to review the material
  - Paper critiques for student presentations.

- Grading
  - Attendance/participation: 10%
  - Quiz: 15%
  - Paper critique: 15%
  - Assignments: 30%
  - Paper presentation: 30%