Practical Contextual Bandits

Alberto Bietti[1] Alekh Agarwal[2] John Langford[2]

[1]Inria, Grenoble
[2]MSR, NY

Télécom ParisTech. December 20th, 2018.

A Contextual Bandit Bake-Off

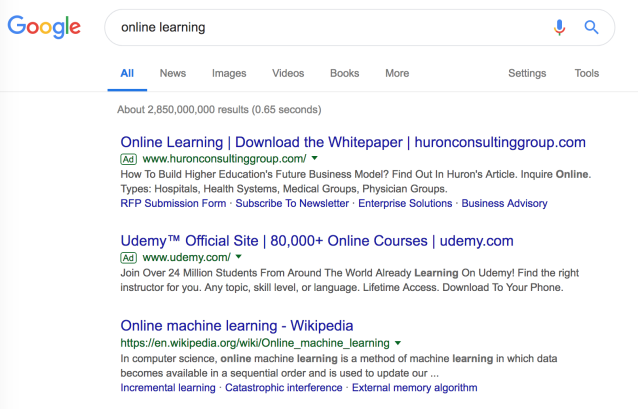
Alberto Bietti[1] Alekh Agarwal[2] John Langford[2]

[1]Inria, Grenoble
[2]MSR, NY

Télécom ParisTech. December 20th, 2018.

Contextual Bandits

Describes many real-world interactive machine learning problems



Ad placement, recommender systems, medical treatment assignment, . . .

Contextual Bandits

Repeat:

- Observe context $x_t \in \mathcal{X}$
  - search query, info about user/item
- Choose action $a_t \in \{1, \ldots, K\}$
  - advertisement, news story, medical treatment
- Observe loss $\ell_t(a_t) \in [0, 1]$ (or $\mathbb{R}$)
  - click/no click, revenue, treatment outcome

Goal: minimize cumulative loss $\sum_{t=1}^{T} \ell_t(a_t)$

Contextual Bandits

Repeat:

- Observe context $x_t \in \mathcal{X}$
  - search query, info about user/item
- Choose action $a_t \in \{1, \dots, K\}$
  - advertisement, news story, medical treatment
- Observe loss $\ell_t(a_t) \in [0, 1]$ (or $\mathbb{R}$)
  - click/no click, revenue, treatment outcome

Goal: minimize cumulative loss $\sum_{t=1}^{T} \ell_t(a_t)$
Need exploration!

Stochastic Contextual Bandits

- $(x_t, \ell_t) \in \mathcal{X} \times [0,1]^K$ sampled i.i.d. from $\mathcal{D}$
- Policy class Π of policies $\pi : \mathcal{X} \to \{1, \ldots, K\}$
  - *e.g.*, linear $\pi(x) = \arg\min_a \theta_a^\top x$
- Exploration algorithm: $a_t \sim p_t(\cdot)$

Stochastic Contextual Bandits

- $(x_t, \ell_t) \in \mathcal{X} \times [0, 1]^K$ sampled i.i.d. from $\mathcal{D}$
- Policy class Π of policies $\pi : \mathcal{X} \to \{1, \ldots, K\}$
  - *e.g.*, linear $\pi(x) = \arg\min_a \theta_a^\top x$
- Exploration algorithm: $a_t \sim p_t(\cdot)$
- Optimal policy: $\pi^* := \arg\min_{\pi \in \Pi} \mathbb{E}_{(x,\ell)\sim\mathcal{D}}[\ell(\pi(x))]$
- **Goal**: minimize regret against π^*:

$$R_T := \sum_{t=1}^{T} \ell_t(a_t) - \sum_{t=1}^{T} \ell_t(\pi^*(x_t))$$

Theory vs Practice

Theory: efficient exploration

- not always statistically efficient (e.g. only worst case)
- not always computationally efficient (e.g. covariance matrix in high dimensions)
- little empirical evaluation

Theory vs Practice

Theory: efficient exploration

- not always statistically efficient (e.g. only worst case)
- not always computationally efficient (e.g. covariance matrix in high dimensions)
- little empirical evaluation

Practice (this work)

- large-scale evaluation on 500+ datasets
- practical, efficient methods using *optimization oracles*
- improved, online implementations (Vowpal Wabbit)

Outline

1. Toolkit
2. Algorithms
3. The Bake-Off
4. Active ϵ-Greedy (bonus)

Optimization Oracles

- Leverage supervised learning algorithms for general policies
- **Cost-sensitive classification** (CSC) oracle:

$$\arg\min_{\pi \in \Pi} \sum_{t=1}^{T} c_t(\pi(x_t))$$

- **Regression** oracle (importance-weighted):

$$\arg\min_{f \in \mathcal{F}} \sum_{t=1}^{T} \omega_t (f(x_t, a_t) - y_t)^2$$

- Construct (x_t, c_t) or (x_t, a_t, y_t) from observed data

Reduction to Off-policy learning

- Common strategy:
  - find good "exploitation" policy π_t using past observed data
  - act according to this policy, but also explore to get useful data
- Interaction data: $(x_t, a_t, \ell_t(a_t), p_t(a_t)), t < T$

Reduction to Off-policy learning

- Common strategy:
  - find good "exploitation" policy π_t using past observed data
  - act according to this policy, but also explore to get useful data
- Interaction data: $(x_t, a_t, \ell_t(a_t), p_t(a_t)), t < T$
- **Off-policy**: data collected using different policies p_t
- Find π_T s.t. $L(\pi_T) \approx \min_\pi L(\pi)$, with $L(\pi) = \mathbb{E}_{\mathcal{D}}[\ell(\pi(x))]$
- Typically need uniform exploration: $p_t(a) > 0$ for all a
- **How?** loss estimation!

Reduction to Off-policy learning: loss estimation

- Construct $\hat{\ell}_t$ from observed data s.t. $\frac{1}{T}\sum_{t=1}^{T}\hat{\ell}_t(\pi(x_t)) \approx L(\pi)$
- Learn via CSC with examples $(x_t, \hat{\ell}_t)$

Reduction to Off-policy learning: loss estimation

- Construct $\hat{\ell}_t$ from observed data s.t. $\frac{1}{T}\sum_{t=1}^{T}\hat{\ell}_t(\pi(x_t)) \approx L(\pi)$
- Learn via CSC with examples $(x_t, \hat{\ell}_t)$
- **IPS** (inverse propensity scoring)

$$\hat{\ell}_t(a) := \frac{\ell_t(a_t)}{p_t(a_t)}\,\mathbb{1}\{a = a_t\}$$

Reduction to Off-policy learning: loss estimation

- Construct $\hat{\ell}_t$ from observed data s.t. $\frac{1}{T}\sum_{t=1}^{T}\hat{\ell}_t(\pi(x_t)) \approx L(\pi)$
- Learn via CSC with examples $(x_t, \hat{\ell}_t)$
- **IPS** (inverse propensity scoring)

$$\hat{\ell}_t(a) := \frac{\ell_t(a_t)}{p_t(a_t)}\,\mathbb{1}\{a = a_t\}$$

- **DR** (doubly robust, Dudik et al., 2011)

$$\hat{\ell}_t(a) := \frac{\ell_t(a_t) - \hat{\ell}(x_t, a_t)}{p_t(a_t)}\,\mathbb{1}\{a = a_t\} + \hat{\ell}(x_t, a)$$

  - $\hat{\ell}$ trained via regression on observed data
- Both **unbiased** when $p_t(\cdot) > 0$, DR has **lower variance**

Reduction to Off-policy learning: IWR

- **Importance-weighted regression** (IWR) reduction:

$$\hat{f} := \arg\min_{f \in \mathcal{F}} \sum_{t=1}^{T} \frac{1}{p_t(a_t)} (f(x_t, a_t) - \ell_t(a_t))^2,$$

- Computational/optimization benefits: only update a single action, importance-weighted updates

Practical Considerations

- CSC **harder** than regression (approximated with K regressors)
- **online learning**: online updates for policies/regressors
- **loss encodings**: $\ell_t(a_t) \in \{0, 1\}$ or $\{-1, 0\}$ for binary costs?
  - e.g. for click/no click outcomes
  - important design choice for better performance (lower variance)
- ***baseline***: learn global additive constant separately
- **learning dynamics**: random tie breaking, pessimistic initialization

Outline

1. Toolkit
2. Algorithms
3. The Bake-Off
4. Active ϵ-Greedy (bonus)

CB algorithm families

- ϵ-Greedy
- Greedy
- Thompson Sampling
- Optimism
- "(mini-)monster" (Agarwal et al., 2014)

ϵ-Greedy (Langford and Zhang, 2007)

- Always explore uniformly with prob. ϵ

$$p_t(a) = \epsilon/K + (1 - \epsilon)\, \mathbb{1}\{\pi_t(x_t) = a\}$$

- Learn by reduction to off-policy learning (IPS/DR/IWR)

$$\pi_{t+1} \leftarrow \mathit{oracle}(\pi_t, (x_t, a_t, \ell_t(a_t), p_t(a_t)))$$

ϵ-Greedy (Langford and Zhang, 2007)

- Always explore uniformly with prob. ϵ

$$p_t(a) = \epsilon/K + (1-\epsilon)\,\mathbb{1}\{\pi_t(x_t) = a\}$$

- Learn by reduction to off-policy learning (IPS/DR/IWR)

$$\pi_{t+1} \leftarrow \mathit{oracle}(\pi_t, (x_t, a_t, \ell_t(a_t), p_t(a_t)))$$

- $\sqrt{T/\epsilon}$ (exploit) + $T\epsilon$ (explore) $\to O(T^{2/3})$ regret
- Lots of wasted exploration
  - "active" ϵ-greedy variant can improve this (see below)

Greedy

- Take $\epsilon = 0$ in ϵ-Greedy with regression oracle:

$$a_t = \arg\min_a f_t(x_t, a)$$

- Can still explore enough!
- Leverage **diversity** in the contexts (Bastani et al., 2017; Kannan et al., 2018)
- Performs surprisingly well on many datasets

Bag (bootstrap Thompson Sampling)

- Thompson Sampling: maintain posterior over policies
- Approximate this posterior using (online) bootstrap (Agarwal et al., 2014; Eckles and Kaptein, 2014; Osband and Van Roy, 2015)

Bag (bootstrap Thompson Sampling)

- Thompson Sampling: maintain posterior over policies
- Approximate this posterior using (online) bootstrap (Agarwal et al., 2014; Eckles and Kaptein, 2014; Osband and Van Roy, 2015)
- Maintain N policies $\pi^1, \ldots, \pi^N$
- Explore uniformly over policies
- Update each using a bootstrap sample of exploration data (via reduction)
- **Bag-greedy**: π^1 uses regular sample instead of bootstrap

Bag (bootstrap Thompson Sampling)

Algorithm 3 Bag

$\pi_1^1, \ldots, \pi_1^N$.
`explore`(x_t):
  return $p_t(a) \propto |\{i : \pi_t^i(x_t) = a\}|$;[3]
`learn`$(x_t, a_t, \ell_t(a_t), p_t)$:
  for $i = 1, \ldots, N$ **do**
    $\tau^i \sim Poisson(1)$; $\{$with $\tau^1 = 1$ for bag-greedy$\}$
    $\pi_{t+1}^i = \texttt{oracle}^{\tau^i}(\pi_t^i, x_t, a_t, \ell_t(a_t), p_t(a_t))$;
  end for

Cover (Agarwal et al., 2014)

- "mini-monster": minimax optimal + computationally "efficient"
- Maintain a distribution over policies that are good for exploration and exploitation (low regret + low variance)

Cover (Agarwal et al., 2014)

- "mini-monster": minimax optimal + computationally "efficient"
- Maintain a distribution over policies that are good for exploration and exploitation (low regret + low variance)
- In practice: N policies $\pi^1, \dots, \pi^N$
- Explore uniformly over policies
- Update π^1 using IPS/DR
- Subsequent policies use CSC with additional cost to encourage **diverse** policies

Cover (Agarwal et al., 2014)

- “mini-monster”: minimax optimal + computationally “efficient”
- Maintain a distribution over policies that are good for exploration and exploitation (low regret + low variance)
- In practice: N policies $\pi^1, \ldots, \pi^N$
- Explore uniformly over policies
- Update π^1 using IPS/DR
- Subsequent policies use CSC with additional cost to encourage **diverse** policies
- **Cover-NU**: remove uniform exploration on all actions (required from theory)
- Still often too much exploration by design

Cover (Agarwal et al., 2014)

Algorithm 4 Cover

$\pi_1^1, \ldots, \pi_1^N$; $\epsilon_t = \min(1/K, 1/\sqrt{Kt})$; $\psi > 0$.

`explore`(x_t):

  $p_t(a) \propto |\{i : \pi_t^i(x_t) = a\}|$;

  return $\epsilon_t + (1 - \epsilon_t)p_t$; {for cover}

  return p_t; {for cover-nu}

`learn`$(x_t, a_t, \ell_t(a_t), p_t)$:

  $\pi_{t+1}^1 = \texttt{oracle}(\pi_t^1, x_t, a_t, \ell_t(a_t), p_t(a_t))$;

  $\hat{\ell}_t = \texttt{estimator}(x_t, a_t, \ell_t(a_t), p_t(a_t))$;

  for $i = 2, \ldots, N$ **do**

    $q_i(a) \propto |\{j \leq i - 1 : \pi_{t+1}^j(x_t) = a\}|$;

    $\hat{c}(a) = \hat{\ell}_t(a) - \frac{\psi \epsilon_t}{\epsilon_t + (1 - \epsilon_t) q_i(a)}$;

    $\pi_{t+1}^i = \texttt{csc\_oracle}(\pi_t^i, x_t, \hat{c})$;

  end for

RegCB (Foster et al., 2018)

- Construct **confidence bounds** on each action based on good regressors for loss estimation

$$\mathcal{F}_t = \{f \in \mathcal{F} : MSE(f, \mathcal{D}_t) - \min_{f \in \mathcal{F}} MSE(f, \mathcal{D}_t) \leq C/t\}$$

$$LCB(x_t, a) = \min_{f \in \mathcal{F}_{t-1}} f(x_t, a)$$

$$UCB(x_t, a) = \max_{f \in \mathcal{F}_{t-1}} f(x_t, a)$$

RegCB (Foster et al., 2018)

- Construct **confidence bounds** on each action based on good regressors for loss estimation

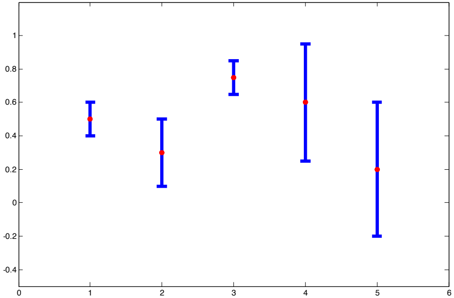
$$\mathcal{F}_t = \{f \in \mathcal{F} : MSE(f, \mathcal{D}_t) - \min_{f \in \mathcal{F}} MSE(f, \mathcal{D}_t) \leq C/t\}$$

$$LCB(x_t, a) = \min_{f \in \mathcal{F}_{t-1}} f(x_t, a)$$

$$UCB(x_t, a) = \max_{f \in \mathcal{F}_{t-1}} f(x_t, a)$$

- LCB and UCB can be computed efficiently using **regression oracles**
- Even with online learning, using importance weight sensitivity
- Explore using **optimism**, or uniform **sampling** over surviving actions

RegCB (Foster et al., 2018)



RegCB (Foster et al., 2018)

Algorithm 5 RegCB

f_1; $C_0 > 0$.

`explore`(x_t):

$l_t(a) = \texttt{lcb}(f_t, x_t, a, \Delta_{t,C_0})$;

$u_t(a) = \texttt{ucb}(f_t, x_t, a, \Delta_{t,C_0})$;

$p_t(a) \propto \mathbb{1}\{a \in \arg\min_{a'} l_t(a')\}$; {RegCB-opt variant}

$p_t(a) \propto \mathbb{1}\{l_t(a) \leq \min_{a'} u_t(a')\}$; {RegCB-elim variant}

return p_t;

`learn`$(x_t, a_t, \ell_t(a_t), p_t)$:

$f_{t+1} = \texttt{reg\_oracle}(f_t, x_t, a_t, \ell_t(a_t))$;

Outline

1 Toolkit

2 Algorithms

3 The Bake-Off

4 Active ϵ-Greedy (bonus)

Evaluation Approach

- 500+ diverse datasets
  - 525 multi-class from `openml.org` (text, bio, medical, sensor, synthetic)
  - 5 multi-label
  - 3 cost-sensitive
- Supervised $(x_t, c_t) \rightarrow$ only reveal $c_t(a_t)$ in CB
- Online learning, linear models in Vowpal Wabbit (`hunch.net/~vw`)

Evaluation Approach

- 500+ diverse datasets
  - 525 multi-class from `openml.org` (text, bio, medical, sensor, synthetic)
  - 5 multi-label
  - 3 cost-sensitive
- Supervised $(x_t, c_t) \to$ only reveal $c_t(a_t)$ in CB
- Online learning, linear models in Vowpal Wabbit (`hunch.net/~vw`)
- **Progressive validation** loss (Blum et al., 1999)

$$PV = \frac{1}{T} \sum_{t=1}^{T} c_t(a_t)$$

- Compare A vs B using statistical test on PV
  - # of datasets where A significantly wins against B

Greed is Good, Optimism is Best

significant win-loss difference, fixed hyperparameters, -1/0 encoding

$\downarrow$ vs $\rightarrow$	G	RO	C-nu	B-g	ϵG
G	-	-7	10	50	54
RO	**7**	-	**26**	**49**	**68**
C-nu	-10	-26	-	22	57
B-g	-50	-49	-22	-	17
ϵG	-54	-68	-57	-17	-

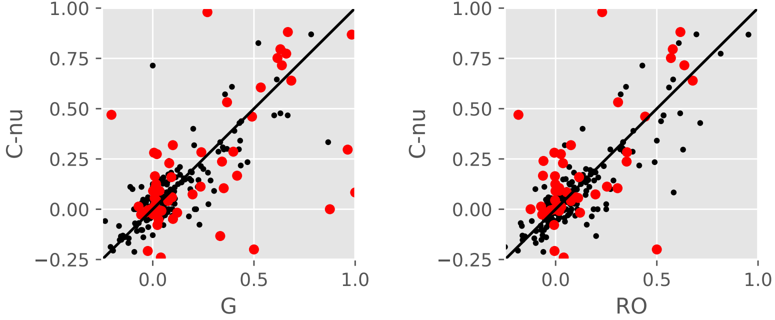
Greed is Good, Optimism is Best

significant win-loss difference, fixed hyperparameters, -1/0 encoding

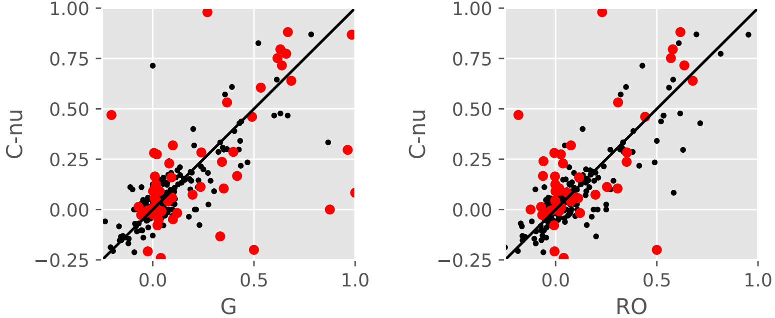
$\downarrow$ vs $\rightarrow$	G	RO	C-nu	B-g	ϵG
G	-	-7	10	50	54
RO	**7**	-	**26**	**49**	**68**
C-nu	-10	-26	-	22	57
B-g	-50	-49	-22	-	17
ϵG	-54	-68	-57	-17	-

Similar results for subsets with large T, large $\#$ features, large K

Cover can be preferred



Cover can be preferred



More robust to difficult datasets, but less efficient

Cover can be preferred

optimized hyperparams

$\downarrow$ vs $\rightarrow$	G	RO	C-nu	B-g	ϵG
G	-	-23	-50	16	92
RO	23	-	-3	42	112
C-nu	**50**	**3**	-	**64**	**142**
B-g	-16	-42	-64	-	90
ϵG	-92	-112	-142	-90	-

More adaptive variant would be desirable

Easy Data?

- Better exploration when supervised learning does well?
- "first-order" bounds (open problem for CBs: Agarwal et al., 2017)
- Exploration algorithms (especially Cover-NU) not so good

Easy Data?

$\downarrow$ vs $\rightarrow$	G	RO	C-nu	B-g	ϵG
G	-	**1**	**25**	**40**	**36**
RO	-1	-	26	36	43
C-nu	-25	-26	-	7	24
B-g	-40	-36	-7	-	4
ϵG	-36	-43	-24	-4	-

$PV_{OAA} \leq 0.2$ (135 datasets)

$\downarrow$ vs $\rightarrow$	G	RO	C-nu	B-g	ϵG
G	-	**1**	**14**	**8**	**15**
RO	-1	-	12	5	16
C-nu	-14	-12	-	-12	5
B-g	-8	-5	12	-	10
ϵG	-15	-16	-5	-10	-

$PV_{OAA} \leq 0.05$ (28 datasets)

Easy Data?

$\downarrow$ vs $\rightarrow$	G	RO	C-nu	B-g	ϵG
G	-	**2**	**8**	**5**	**12**
RO	-2	-	8	4	12
C-nu	-8	-8	-	-1	12
B-g	-5	-4	1	-	11
ϵG	-12	-12	-12	-11	-

$n \geq 10\,000$ and $PV_{OAA} \leq 0.1$ (13 datasets)

Reductions

- Doubly Robust always better than IPS
- When appropriate (ϵ-Greedy, Bagging), IWR is best
  - Better performance
  - Computationally more efficient

Reductions

ϵ-Greedy

$\downarrow$ vs $\rightarrow$	ips	dr	iwr
ips	-	-42	-59
dr	42	-	-28
iwr	**59**	**28**	-

Bag

$\downarrow$ vs $\rightarrow$	ips	dr	iwr
ips	-	63	-133
dr	-63	-	-155
iwr	**133**	**155**	-

Loss Encoding

- Binary outcomes: how to encode loss of success/failure?
- Provides initial bias, can lead to lower variance
- -1/0 is a good default choice
- Rule of thumb: -1/0 better if “failure” is common
  - e.g. no click more frequent than click
- 0/1 can be better once learner selects good actions
  - e.g. large, easy dataset

Loss Encoding

significant win/loss of -1/0 vs 0/1

datasets	G	RO	C-nu	B-g	ϵG
all	136 / 42	60 / 47	76 / 46	77 / 27	99 / 27
$\geq$ 10,000	19 / 12	10 / 18	14 / 20	15 / 11	14 / 5

Baseline

- Global additive constant in loss estimator

$$\hat{\ell}(x, a) = c + \theta_a^\top x$$

- Learn with separate online update

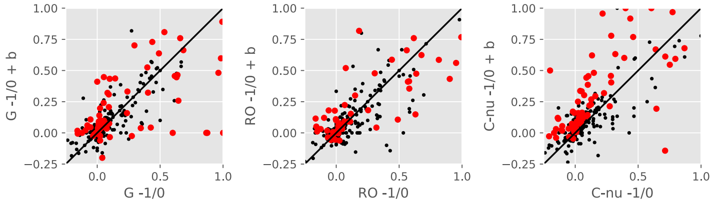
Baseline

- Global additive constant in loss estimator

$$\hat{\ell}(x, a) = c + \theta_a^\top x$$

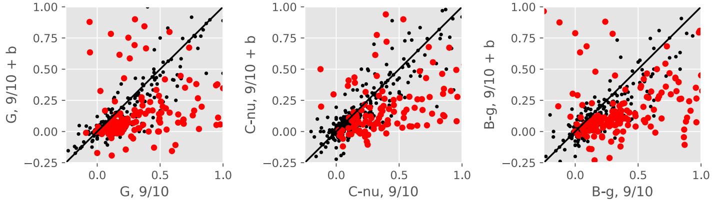
- Learn with separate online update
- Good to fight initial **pessimism** (e.g. -1/0) in Greedy/RegCB-optimistic
- Adapt to unknown loss range

Baseline



G -1/0 + b
G -1/0
RO -1/0 + b
RO -1/0
C-nu -1/0 + b
C-nu -1/0
1.00
0.75
0.50
0.25
0.00
−0.25
0.0
0.5
1.0

Baseline



1.00
0.75
0.50
0.25
0.00
−0.25
0.0
0.5
1.0
G, 9/10 + b
G, 9/10
C-nu, 9/10 + b
C-nu, 9/10
B-g, 9/10 + b
B-g, 9/10

Outline

1 Toolkit

2 Algorithms

3 The Bake-Off

4 Active ϵ-Greedy (bonus)

Active ϵ-Greedy: motivation

- ϵ-Greedy often a simple default method
- **But**: uniform exploration on all actions is too costly!

Active ϵ-Greedy: motivation

- ϵ-Greedy often a simple default method
- **But**: uniform exploration on all actions is too costly!
- Can we avoid exploring on actions that we know are not useful?
- Only explore if action is plausibly taken by optimal policy
  - Using techniques from disagreement-based active learning

Active ϵ-Greedy: algorithm

- After observing x_t, for any action a
  - try to find a **good** policy with $\pi(x_t) = a$
  - if found, there is disagreement $\implies$ explore
  - if not found, $\pi^*(x_t) \neq a$ w.h.p $\implies$ don't explore
- Good policy: small loss difference $\hat{L}_{t-1}(\pi_{t,\bar{a}}) - \hat{L}_{t-1}(\pi_t)$

$$\pi_t = \arg\min_{\pi} \hat{L}_{t-1}(\pi)$$
$$\pi_{t,\bar{a}} = \arg\min_{\pi:\pi(x_t)=\bar{a}} \hat{L}_{t-1}(\pi).$$

  - Can be computed using importance weight sensitivity analysis
- Explore with ϵ mass on each disagreeing actions, greedily otherwise

Active ϵ-Greedy: algorithm

Algorithm 6 Active ϵ-greedy

π_1; ϵ; $C_0 > 0$.

`explore`(x_t):

$A_t = \{a : \texttt{loss\_diff}(\pi_t, x_t, a) \leq \Delta_{t,C_0}\}$;

$p_t(a) = \frac{\epsilon}{K} \mathbb{1}\{a \in A_t\} + (1 - \frac{\epsilon |A_t|}{K}) \mathbb{1}\{\pi_t(x_t) = a\}$;

return p_t;

`learn`$(x_t, a_t, \ell_t(a_t), p_t)$:

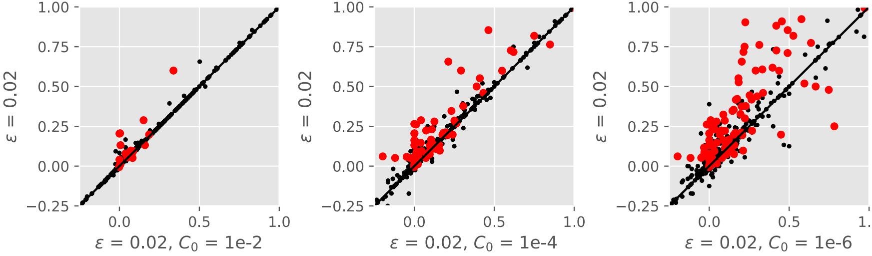
$\hat{\ell}_t = \texttt{estimator}(x_t, a_t, \ell_t(a_t), p_t(a_t))$;

$\hat{c}_t(a) = \begin{cases} \hat{\ell}_t(a), & \text{if } p_t(a) > 0 \\ 1, & \text{otherwise.} \end{cases}$

$\pi_{t+1} = \texttt{csc\_oracle}(\pi_t, x_t, \hat{c}_t)$;

$$\Delta_{t,C_0} = \sqrt{C_0 \frac{K \log t}{\epsilon t}} + C_0 \frac{K \log t}{\epsilon t}$$

Active ϵ-Greedy: algorithm



Active ϵ-Greedy: theory

- Worst-case regret is similar to ϵ-Greedy ($\tilde{O}(T^{2/3})$)

$$O(T^{2/3}(K \log(T|\Pi|/\delta))^{1/3})$$

- Under favorable conditions (disagreement + Massart noise), regret improves to $\tilde{O}(T^{1/3})$

$$O\left(\frac{1}{\tau}(\theta K \log(T|\Pi|/\delta))^{2/3}(T \log T)^{1/3}\right)$$

- Better than minimax rate of Mini-Monster: $O(\sqrt{KT \log(T|\Pi|/\delta)})$
- But, RegCB has **logarithmic** regret in similar conditions with realizability...

Conclusion

- RegCB and Greedy dominate, but need strong modeling assumptions
- Cover-NU more robust on difficult datasets, but too conservative otherwise
- $\implies$ need new robust + adaptive algorithms
- Simple practical design choices can matter a lot (reductions, encodings)
- Caveats/discussion:
  - Only i.i.d., what about non-stationary, or adversarial?
  - Non-linear policy classes?
  - Online vs Batch?

References I

A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. E. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. *arXiv preprint arXiv:1402.0555*, 2014.

A. Agarwal, A. Krishnamurthy, J. Langford, H. Luo, et al. Open problem: First-order regret bounds for contextual bandits. In *Conference on Learning Theory (COLT)*, 2017.

H. Bastani, M. Bayati, and K. Khosravi. Mostly exploration-free algorithms for contextual bandits. *arXiv preprint arXiv:1704.09011*, 2017.

A. Blum, A. Kalai, and J. Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Conference on Learning Theory (COLT)*, 1999.

M. Dudik, J. Langford, and L. Li. Doubly robust policy evaluation and learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2011.

D. Eckles and M. Kaptein. Thompson sampling with the online bootstrap. *arXiv preprint arXiv:1410.4009*, 2014.

References II

D. J. Foster, A. Agarwal, M. Dudík, H. Luo, and R. E. Schapire. Practical contextual bandits with regression oracles. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

S. Kannan, J. Morgenstern, A. Roth, B. Waggoner, and Z. S. Wu. A smoothed analysis of the greedy algorithm for the linear contextual bandit problem. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

J. Langford and T. Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

I. Osband and B. Van Roy. Bootstrapped thompson sampling and deep exploration. *arXiv preprint arXiv:1507.00300*, 2015.