A New Algorithm for Non-stationary Contextual Bandits: Efficient, Optimal and Parameter-free

Yifang Chen, **Chung-Wei Lee**, Haipeng Luo, Chen-Yu Wei

University of Southern California

One Sentence Summary

We achieve similar guarantees for the harder **contextual bandit** setting, **efficiently**.

From MAB to Contextual Bandits [ACFS02,LZ08]

For $t = 1, \ldots, T$,

- The learner chooses $a_t \in \{1, \ldots, K\}$.
- The environment reveals $r_t(a_t)$.

From MAB to Contextual Bandits [ACFS02,LZ08]

For $t = 1, \ldots, T$,

- The learner sees context x_t, where $(x_t, r_t) \sim \mathcal{D}_t$.
- The learner chooses $a_t \in \{1, \ldots, K\}$.
- The environment reveals $r_t(a_t)$.

From MAB to Contextual Bandits [ACFS02,LZ08]

For $t = 1, \ldots, T$,

- The learner sees context x_t, where $(x_t, r_t) \sim \mathcal{D}_t$.
- The learner chooses $a_t \in \{1, \ldots, K\}$.
- The environment reveals $r_t(a_t)$.

Goal: minimize dynamic regret against the best policy at each time

$$\text{Reg} = \sum_{t=1}^{T} \max_{\pi \in \Pi} \mathbb{E}_{(x,r)\sim\mathcal{D}_t}[r(\pi(x))] - \sum_{t=1}^{T} r_t\left(a_t\right),$$

where Π is a policy class: mappings from contexts to actions.

Non-stationarity

- Sublinear dynamic regret is impossible in general.

Non-stationarity

- Sublinear dynamic regret is impossible in general.
- Non-stationarity is measured by

Non-stationarity

- Sublinear dynamic regret is impossible in general.
- Non-stationarity is measured by
  - $S \triangleq 1 + \sum_{t=2}^{T} 1\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\}$

Non-stationarity

- Sublinear dynamic regret is impossible in general.
- Non-stationarity is measured by
  - $S \triangleq 1 + \sum_{t=2}^{T} 1\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\}$
  - or $V \triangleq \sum_{t=2}^{T} \|\mathcal{D}_t - \mathcal{D}_{t-1}\|_{\mathrm{TV}}$

Non-stationarity

- Sublinear dynamic regret is impossible in general.
- Non-stationarity is measured by
  - $S \triangleq 1 + \sum_{t=2}^{T} 1\left\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\right\}$
  - or $V \triangleq \sum_{t=2}^{T} \left\|\mathcal{D}_t - \mathcal{D}_{t-1}\right\|_{\mathrm{TV}}$
- Optimal regret is ([ACFS02,BGZ14]): $\mathcal{O}\left(\min\left\{\sqrt{ST}, V^{\frac{1}{3}} T^{\frac{2}{3}}\right\}\right)$

Non-stationarity

- Sublinear dynamic regret is impossible in general.
- Non-stationarity is measured by
  - $S \triangleq 1 + \sum_{t=2}^{T} 1\left\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\right\}$
  - or $V \triangleq \sum_{t=2}^{T} \|\mathcal{D}_t - \mathcal{D}_{t-1}\|_{\mathrm{TV}}$
- Optimal regret is ([ACFS02,BGZ14]): $\mathcal{O}\left(\min\left\{\sqrt{ST}, V^{\frac{1}{3}}T^{\frac{2}{3}}\right\}\right)$
- **Our algorithm achieves this without knowing S or V efficiently.**

Non-stationarity

- Sublinear dynamic regret is impossible in general.
- Non-stationarity is measured by
  - $S \triangleq 1 + \sum_{t=2}^{T} 1\left\{\mathcal{D}_t \neq \mathcal{D}_{t-1}\right\}$
  - or $V \triangleq \sum_{t=2}^{T} \|\mathcal{D}_t - \mathcal{D}_{t-1}\|_{\mathrm{TV}}$
- Optimal regret is ([ACFS02,BGZ14]): $\mathcal{O}\left(\min\left\{\sqrt{ST}, V^{\frac{1}{3}}T^{\frac{2}{3}}\right\}\right)$
- **Our algorithm achieves this without knowing S or V efficiently.**
- Prior works: [LWAL18] achieves $\mathcal{O}\left(\min\left\{S^{\frac{1}{4}}T^{\frac{3}{4}}, V^{\frac{1}{5}}T^{\frac{4}{5}}\right\}\right)$

Key Ideas

Adapting to S and V:

- inspired by [AGO18], but can't naively treat each policy as an arm

Key Ideas

Adapting to S and V:

- inspired by [AGO18], but can't naively treat each policy as an arm
- introduce the idea of **replay phases**: sample according to previous distributions occasionally,

Key Ideas

Adapting to S and V:

- inspired by [AGO18], but can't naively treat each policy as an arm
- introduce the idea of **replay phases**: sample according to previous distributions occasionally, in the same vein as sampling discarded arms [AGO19]

Key Ideas

Adapting to S and V:

- inspired by [AGO18], but can't naively treat each policy as an arm
- introduce the idea of **replay phases**: sample according to previous distributions occasionally, in the same vein as sampling discarded arms [AGO19]

Oracle-efficiency:

- want to avoid poly($|\Pi|$) time

Key Ideas

Adapting to S and V:

- inspired by [AGO18], but can't naively treat each policy as an arm
- introduce the idea of **replay phases**: sample according to previous distributions occasionally, in the same vein as sampling discarded arms [AGO19]

Oracle-efficiency:

- want to avoid poly($|\Pi|$) time
- as in prior works, assume access to ERM oracle

Key Ideas

Adapting to S and V:

- inspired by [AGO18], but can't naively treat each policy as an arm
- introduce the idea of **replay phases**: sample according to previous distributions occasionally, in the same vein as sampling discarded arms [AGO19]

Oracle-efficiency:

- want to avoid poly($|\Pi|$) time
- as in prior works, assume access to ERM oracle
- based on key ideas of ILOVETOCONBANDITS [AHKLLS14]

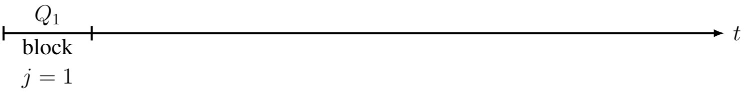
An Overview of ILOVETOCONBANDITS (i.i.d.)

for *block* $j = 1, 2, 3, \ldots$ **do**

   find a sparse distribution Q_j over Π using all previous data

   for *time* $t = 2^{j-1} \ldots 2^j - 1$ **do**

      play Q_j



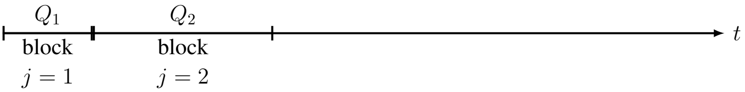
An Overview of ILOVETOCONBANDITS (i.i.d.)

for *block* $j = 1, 2, 3, \ldots$ **do**

  find a sparse distribution Q_j over Π using all previous data

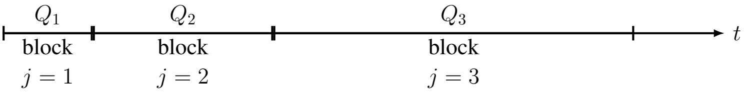
  for *time* $t = 2^{j-1} \ldots 2^j - 1$ **do**

    play Q_j



An Overview of ILOVETOCONBANDITS (i.i.d.)

for *block* $j = 1, 2, 3, \ldots$ **do**
   find a sparse distribution Q_j over Π using all previous data
   for *time* $t = 2^{j-1} \ldots 2^j - 1$ **do**
      play Q_j

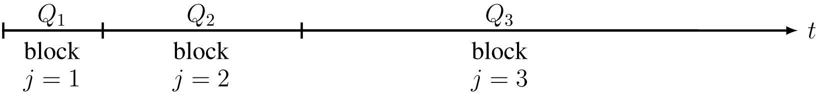


An Overview of Our Algorithm (non-stationary)

for *block* $j = 1, 2, 3, \ldots$ **do**

  Find a sparse distribution Q_j over Π using all data since last restart

  for *time* $t = 2^{j-1} \ldots 2^j - 1$ **do**



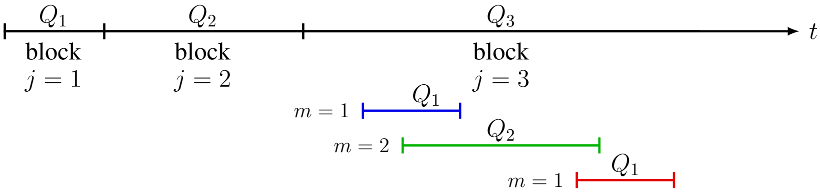
An Overview of Our Algorithm (non-stationary)

for *block* $j = 1, 2, 3, \ldots$ **do**

  Find a sparse distribution Q_j over Π using all data since last restart

  for *time* $t = 2^{j-1} \ldots 2^j - 1$ **do**

    Randomly start a replay phase of length 2^m, add it to $\mathcal{S}$



An Overview of Our Algorithm (non-stationary)

for *block* $j = 1, 2, 3, \ldots$ **do**

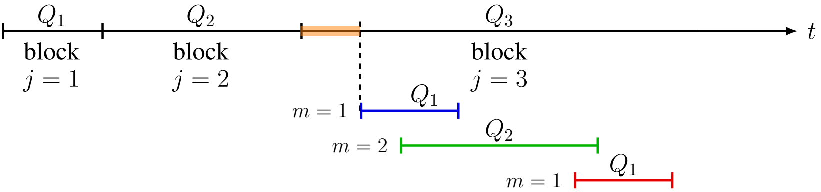
Find a sparse distribution Q_j over Π using all data since last restart

for *time* $t = 2^{j-1} \ldots 2^j - 1$ **do**

Randomly start a replay phase of length 2^m, add it to $\mathcal{S}$

if $\mathcal{S}$ *is empty* **then** Play Q_j;

else Sample u.a.r an "alive" replay phase from $\mathcal{S}$, play Q_m;



An Overview of Our Algorithm (non-stationary)

for *block* $j = 1, 2, 3, \ldots$ **do**

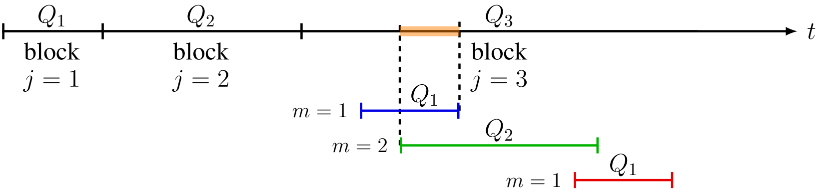
  Find a sparse distribution Q_j over Π using all data since last restart

  for *time* $t = 2^{j-1} \ldots 2^j - 1$ **do**

    Randomly start a replay phase of length 2^m, add it to $\mathcal{S}$

    if $\mathcal{S}$ *is empty* **then** Play Q_j;

    else Sample u.a.r an "alive" replay phase from $\mathcal{S}$, play Q_m;



An Overview of Our Algorithm (non-stationary)

for *block* $j = 1, 2, 3, \ldots$ **do**

  Find a sparse distribution Q_j over Π using all data since last restart

  for *time* $t = 2^{j-1} \ldots 2^j - 1$ **do**

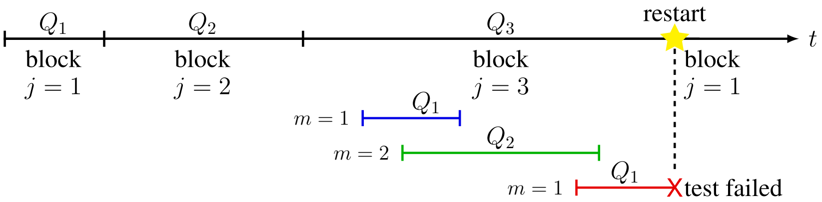
    Randomly start a replay phase of length 2^m, add it to $\mathcal{S}$

    if $\mathcal{S}$ *is empty* **then** Play Q_j;

    else Sample u.a.r an "alive" replay phase from $\mathcal{S}$, play Q_m;

    if *Non-stationarity tests fail* **then**

      Restart from scratch



Summary

Our algorithm achieves dynamic regret $\mathcal{O}\left(\min\left\{\sqrt{ST}, V^{\frac{1}{3}}T^{\frac{2}{3}}\right\}\right)$

- optimal
- oracle-efficient
- without knowing S and V.

Poster #186