Contextual Bandit Problems: A Brief Introduction

Presented by Bo Zhang

January 24, 2019

Motivation I

- In most sequential decision making scenarios, there is some additional information available.
- For instance, in a clinical trial, we have access to subjects' genetic and demographic information.
- *Contextual bandit problem:* how to map user features into one of the available actions.
- Recent applications include: mobile health, news article recommendation, etc.

Motivation II

In such a decision making problem, the fundamental pattern that repeats over time is the following:

1. **at** a given decision point **do**
2. mobile phone collects tailoring variables (the context)
3. a decision rule maps the variables into an intervention
4. mobile phone records the proximal outcome (the reward)
5. **done**

Stochastic Contextual Bandits: Notation I

1. For simplicity, first assume two actions: $a = 0$ or $a = 1$. The data

$$\{(X_t, R_t^0, R_t^1)\}_{t=1}^T$$

is I.I.D. from some underlying distribution, where X_t is the context, $R_t^a, a = 0, 1$ is the reward under action $a \in \{0, 1\}$.

2. A *policy* or *decision* rule π maps an element X_t to an action. The *value* of a policy π is defined as:

$$V(\pi) = E[R^{\pi(X)}]$$

where the expectation is taken w.r.t the data-generating process of (X, R^0, R^1).

Stochastic Contextual Bandits: Notation II

1. The *expected reward function* is defined as

$$\eta_a = E(R^a|X = x), a = 0, 1$$

and it is easily seen that

$$V(\pi) = E[\eta_{\pi(X)}(X)].$$

2. The *optimal policy* is

$$\pi^*(x) = argmax_a\ \eta_a(x).$$

Stochastic Contextual Bandits: Greedy Policy

1. The problem can be viewed as one of estimating the expected reward functions $\eta_a(x)$. Given estimated reward $\hat{\eta}_a(x), a \in \mathcal{A}$, the greedy policy can be used:

$$GREEDY(\hat{\eta}_a)(x) = argmax_{a \in \mathcal{A}}\ \hat{\eta}_a(x)$$

2. The key is to estimate the reward $\eta_a(x)$ in each arm.

Stochastic Contextual Bandits: Parametric Approach I

Consider the familiar model:

$$R_t^a = \beta_a^T X_t + \epsilon_t^a$$

so that the expected reward is $\eta_a(x) = \beta_a^T x$. The best policy is therefore

$$\pi^*(x) = argmax_{a \in \mathcal{A}} \ \beta_a^T x$$

and the *regret* over T time steps is

$$T \cdot V(\pi^*) - \sum_{t=1}^{T} E[R_t].$$

Note $T \cdot V(\pi^*)$ is the best expected cumulative reward if you know the true β_a, while $\sum_{t=1}^{T} E[R_t]$ is the accumulated reward by the learning algorithm.

Stochastic Contextual Bandits: Parametric Approach II

1. One intuitive algorithm is as follows (assuming two arms):
   - Explore both arms for a period of time.
   - Model the reward via a regression model using the data accumulated during exploration.
   - Exploit the current estimate of the expected reward after the exploration period and takes the optimal action accordingly.
2. Goldenshluger and Zeevi (2013) adopts this intuition (Algorithm 1 on the next slide). They established an $O(p^3 \log T)$ regret bound.
3. You can also assume sparsity, extend the algorithm to high dimensional X, and improve the regret bound.

Stochastic Contextual Bandits: Parametric Approach III

Algorithm 1 Linear Response Bandit Algorithm [22]

Inputs: n_0 (initial exploration length), $\mathcal{T}_a$ (exploration times for action a), h (localization parameter to decide which estimates to use)

for $t = 1$ to $2n_0$ **do**
  Take action $A_t = 0$ or $A_t = 1$ depending on whether t is odd or even
end for
for $t = 2n_0 + 1$ to T **do**
  if $t \in \mathcal{T}_a$ **then**
    /$\star$ Exploration round $\star$/
    Take action $A_t = a$
    Update $\widehat{\beta}_a$ using least squares on previous rounds when action a was taken
    Update $\widetilde{\beta}_a$ using least squares on previous exploration rounds when action a was taken
  else
    /$\star$ Exploitation round $\star$/
    if $|(\widetilde{\beta}_1 - \widetilde{\beta}_0)^\top X_t| > h/2$ **then**
      Take action $A_t = \text{argmax}_a (\widetilde{\beta}^a)^\top X_t$
    else
      Take action $A_t = \text{argmax}_a (\widehat{\beta}^a)^\top X_t$
    end if
  end if
end for

SCB: Nonparametric Approach

Parametric models can be restrictive. We may consider the following model instead:

$$R_t^a = f_a(X_t) + \epsilon_t^a.$$

Yang and Zhu (2002) combined nonparametric regression with the ϵ-greedy strategy:

Algorithm 2 Randomized Allocation with Nonparametric Estimation [14]

Inputs: n_0 (initial exploration length), NPR (nonparametric regression procedure such as nearest neighbor regression), ε_t (sequence of exploration probabilities)

for $t = 1$ to $2n_0$ **do**
  Take action $A_t = 0$ or $A_t = 1$ depending on whether t is odd or even
end for
Get initial estimates $\widehat{f}^a$ by feeding data from previous rounds to NPR
for $t = 2n_0 + 1$ to T **do**
  Let $G_t = \text{argmax}_a \widehat{f}^a(X_t)$ // greedy action
  Let E_t = action selected at random // random exploration
  With probability $(1 - \varepsilon_t)$ take action $A_t = G_t$, else $A_t = E_t$ // ε-greedy
  Collect reward R_t and feed into NPR to get updated estimate $\widehat{f}^a$ for $a = A_t$
end for

Adversarial Contexts with Stochastic Rewards: Notation

- The contexts are arbitrary, i.e., they are no longer I.I.D. random variables.
- Rewards are still drawn from $\mathcal{D}^a(\cdot|x) : x \in \mathcal{X}$.
- The optimal policy is still

$$\pi^*(x) = argmax_{a \in \mathcal{A}} \, \eta_a(x)$$

  and the regret

$$\sum_{t=1}^{T} \eta_{\pi^*(x_t)}(x_t) - \sum_{t=1}^{T} E[R_t].$$

- Regret bounds need to hold uniformly over all possible sequences $\{x_t\}_{t=1}^T$ of contexts.

Adversarial Contexts with Stochastic Rewards: Algorithm I

Algorithm 4 LinUCB Algorithm [2]

Inputs: α (tuning parameter used in computing upper confidence bounds)
$\mathbf{A}^a = \mathbf{I}_{p\times p}, \mathbf{b}^a = \mathbf{0}_{p\times 1}$ for all a

for $t = 1$ to T **do**
  Compute $\hat{\beta}^a = (\mathbf{A}^a)^{-1}\mathbf{b}^a$ for all a // ridge regression
  Compute $U^a = (\hat{\beta}^a)^\top x_t + \alpha\sqrt{x_t^\top(\mathbf{A}^a)^{-1}x_t}$ for all a // upper confidence bound
  Take action $A_t = \text{argmax}_a U^a$ and observe reward R_t
  For $a = A_t$, update $\mathbf{A}^a = \mathbf{A}^a + x_t x_t^\top$, $\mathbf{b}^a = \mathbf{b}^a + R_t x_t$
end for

- $A^a = D_a^T D_a + I_p$, where D_a is the design matrix in arm a.
- We have

$$|(\hat{\beta}^a)^T x_t - \beta_a^T x_t| \leq \alpha\sqrt{x_t(A^a)^{-1}x_t}$$

with probability at least $1-\delta$ for any δ and x_t, with $\alpha = 1 + \sqrt{ln(2/\delta)/2}$.

Adversarial Contexts with Stochastic Rewards: Algorithm II

A Bayesian perspective involves putting a prior on the parameters β^a and draw samples from posterior of β^a.

Algorithm 5 Thompson Sampling Algorithm [2]

Inputs: σ^2 (variance parameter used in the prior and in the reward linear model)
$\mathbf{A}^a = \mathbf{I}_{p\times p}, \mathbf{b}^a = \mathbf{0}_{p\times 1}$ for all a

for $t = 1$ to T **do**
  Compute $\hat{\beta}^a = (\mathbf{A}^a)^{-1}\mathbf{b}^a$ for all a
  Sample $\widetilde{\beta}^a$ from $\text{NORMAL}(\hat{\beta}^a, \sigma^2(\mathbf{A}^a)^{-1})$ for all a // Sample from the posterior
  Take action $A_t = \text{argmax}_a(\widetilde{\beta}^a)^\top x_t$ and observe reward R_t
  For $a = A_t$, update $\mathbf{A}^a = \mathbf{A}^a + x_t x_t^\top$, $\mathbf{b}^a = \mathbf{b}^a + R_t x_t$
end for

Fully Adversarial Contextual Bandits I

- One protocol is as follows:
  1. nature generates $\{(x_t, \mathcal{D}_t^0(\cdot|x), \mathcal{D}_t^1(\cdot|x))\}_{t=1}^T$ in advance
  2. **for** t = 1 to T **do**
  3. receive context x_t
  4. algorithm takes action A_t
  5. receive reward R_t from $\mathcal{D}_t^{A_t}(\cdot|x)$ with expectation $\eta_t^{A_t}(x_t)$
  6. **end**
- Slivkins (2014) gave an algorithm with regret bound of $O(T^{1-1/(2+d_{\mathcal{X}})}(\log T))$ where $d_{\mathcal{X}}$ is the covering dimension of $\mathcal{X}$, which satisfies $d_{\mathcal{X}} \leq p$ when $\mathcal{X} \subseteq \mathbb{R}^p$.

Fully Adversarial Contextual Bandits II

- Another protocol is as follows:
  1. nature generates $\{(x_t, r_t^0, r_t^1)\}$ in advance
  2. **for** t = 1 to T **do**
  3. receive context x_t
  4. algorithm takes action A_t
  5. receive reward $R_t = r_t^{A_t}$
  6. **end**
- Regret is now defined as

$$\max_{\pi \in \Pi} \sum_{t=1}^{T} r_t^{\pi(x_t)} - \sum_{t=1}^{T} E[R_t]$$

  where Π is a fixed class of policies.
- When $x_t = x$, this reduces to the multi-armed bandit problem with K arms and Exp family algorithms work.