Online Learning: Introduction to Multi-Armed Bandits

Sham A. Puthiya Parambath

July 2, 2021

Introduction

Offline vs Online Learning

- Standard learning paradigm using historical data is called offline learning.

Introduction

Offline vs Online Learning

- Standard learning paradigm using historical data is called offline learning.
- Many variants like supervised, semi-supervised, unsupervised and domain adaptation.

Introduction

Offline vs Online Learning

- Standard learning paradigm using historical data is called offline learning.
- Many variants like supervised, semi-supervised, unsupervised and domain adaptation.

- Online learning algorithms learn from incoming sequences of data.

Introduction

Offline vs Online Learning

- In offline learning, the hypothesis space is fixed during the training phase whereas in online learning it can change in every round.

Introduction

Offline vs Online Learning

- In offline learning, the hypothesis space is fixed during the training phase whereas in online learning it can change in every round.
- In online learning there is no distinction between training and test set

Introduction

Offline vs Online Learning

- In offline learning, the hypothesis space is fixed during the training phase whereas in online learning it can change in every round.
- In online learning there is no distinction between training and test set
- Statistical assumption regarding the generative process

Online Learning

As a Game

- Online learning can be considered as a repetitive game

As a Game

- Online learning can be considered as a repetitive game
- The game proceeds as follows *(i)* predictor chooses an instance *(ii)* environment chooses an instance *(iii)* predictor incurs a reward (or loss) and *(iv)* predictor uses the loss/reward to improve the next round selection. Consider spam detection as an example.

Online Learning

Performance Measure

- How do we measure the performance of online algorithms?

Online Learning

Performance Measure

- How do we measure the performance of online algorithms?
- To measure performance, we rely on regret.

Online Learning

Performance Measure

- How do we measure the performance of online algorithms?
- To measure performance, we rely on regret.
- Regret is defined as $\mathrm{R}(T) = \sum_{t=1}^{T} (\hat{y}_t - y_t)$

Online Learning

Performance Measure

- How do we measure the performance of online algorithms?
- To measure performance, we rely on regret.
- Regret is defined as $\mathrm{R}(T) = \sum_{t=1}^{T} (\hat{y}_t - y_t)$
- If regret is linear, the algorithm is not learning anything.

Online Learning

General Online Learning Problems

- Online classification, Online regression

Online Learning

General Online Learning Problems

- Online classification, Online regression
- For online classification and regression problems we can use algorithms like FTL, FTRL etc.

Online Learning

General Online Learning Problems

- Online classification, Online regression
- For online classification and regression problems we can use algorithms like FTL, FTRL etc.
- Our main focus is on online learning problems involving exploration-exploitation trade-off. One instantiation of such a problem is multi-armed bandits (MAB)

Online Learning

Multi-Armed Bandit Problem

- In MAB, we have $k \geq 2$ arms corresponding to k actions.

Multi-Armed Bandit Problem

- In MAB, we have $k \geq 2$ arms corresponding to k actions.
- At each round, our algorithm chooses one of the actions and receives feedback (reward or loss) associated with that specific action.

Online Learning

Multi-Armed Bandit Problem

- In MAB, we have $k \geq 2$ arms corresponding to k actions.
- At each round, our algorithm chooses one of the actions and receives feedback (reward or loss) associated with that specific action.
- Two reward distribution settings are studied within the MAB framework *(i)* stochastic and *(ii)* adversarial.

Multi-Armed Bandit Problem

- Stochastic MAB problems can be further classified into *(i)* classical and *(ii)* pure-exploration

Multi-Armed Bandit Problem

- Stochastic MAB problems can be further classified into *(i)* classical and *(ii)* pure-exploration
- The objective of the two problem settings are different.

Multi-Armed Bandit Problem

- Stochastic MAB problems can be further classified into *(i)* classical and *(ii)* pure-exploration
- The objective of the two problem settings are different.
- Classical algorithms fails in pure-exploration settings.

Stochastic Bandits

Upper Confidence Bound Algorithm (UCB)

- UCB is one of the most popular stochastic MAB algorithms

Stochastic Bandits

Upper Confidence Bound Algorithm (UCB)

- UCB is one of the most popular stochastic MAB algorithms
- Try all the actions once

Stochastic Bandits

Upper Confidence Bound Algorithm (UCB)

- UCB is one of the most popular stochastic MAB algorithms
- Try all the actions once
- At round t, pick the action a that maximizes $r_a(t) + \sqrt{\frac{2 \ln t}{t_a}}$

Stochastic Bandits

Upper Confidence Bound Algorithm (UCB)

- UCB is one of the most popular stochastic MAB algorithms
- Try all the actions once
- At round t, pick the action a that maximizes $r_a(t) + \sqrt{\frac{2 \ln t}{t_a}}$
- One of the simplest algorithm that attains logarithmic regret

Non-Stochastic Bandits

EXPonential-weight for EXPloration and EXPloitation

- EXP3 is one of the simplest non-stochastic bandit algorithm

Non-Stochastic Bandits

EXPonential-weight for EXPloration and EXPloitation

- EXP3 is one of the simplest non-stochastic bandit algorithm
- EXP3 maintain a distribution vector P_t over the k actions initialized to $\frac{1}{k}$

Non-Stochastic Bandits

EXPonential-weight for EXPloration and EXPloitation

- EXP3 is one of the simplest non-stochastic bandit algorithm
- EXP3 maintain a distribution vector P_t over the k actions initialized to $\frac{1}{k}$
- At each round t, algorithm selects one action based on P_t and update the distribution based on the feedback

Non-Stochastic Bandits

EXPonential-weight for EXPloration and EXPloitation

- EXP3 is one of the simplest non-stochastic bandit algorithm
- EXP3 maintain a distribution vector P_t over the k actions initialized to $\frac{1}{k}$
- At each round t, algorithm selects one action based on P_t and update the distribution based on the feedback
- Updates are calculated following $\frac{\exp \eta \hat{r}_{i,t}}{\sum_i \exp \eta \hat{r}_{i,t}}$

Non-Stochastic Bandits

EXPonential-weight for EXPloration and EXPloitation

- EXP3 is one of the simplest non-stochastic bandit algorithm
- EXP3 maintain a distribution vector P_t over the k actions initialized to $\frac{1}{k}$
- At each round t, algorithm selects one action based on P_t and update the distribution based on the feedback
- Updates are calculated following $\frac{\exp \eta \hat{r}_{i,t}}{\sum_i \exp \eta \hat{r}_{i,t}}$
- Regret of the EXP3 is of the order $O(\sqrt{kT \ln k})$

Query Recommendations with Bandits

Contextual Bandits For Query Recommendations

- Consider the query recommendation problem in closed loop settings.

Query Recommendations with Bandits

Contextual Bandits For Query Recommendations

- Consider the query recommendation problem in closed loop settings.
- Given the initial query, represented as context vectors, how can we help the user to explore the query space efficiently

Query Recommendations with Bandits

Contextual Bandits For Query Recommendations

- Consider the query recommendation problem in closed loop settings.
- Given the initial query, represented as context vectors, how can we help the user to explore the query space efficiently
- Here the number of arms are countably many. So trying all the arms will not work. We need to select arms that might be of interest to the user.

Query Recommendations with Bandits

Contextual Bandits For Query Recommendations

- Consider the query recommendation problem in closed loop settings.
- Given the initial query, represented as context vectors, how can we help the user to explore the query space efficiently
- Here the number of arms are countably many. So trying all the arms will not work. We need to select arms that might be of interest to the user.
- We propose to select the queries based on max-utility.

Contextual Bandits For Query Recommendations

MAB algorithms work by randomly picking two arms and estimating statistical properties by running on-the-fly hypothesis testing. We propose to estimate the sampling probabilities using the similarity between context vectors.

Let $s_{j,i} = \text{sim}(a_j, a_i)$. $S^i_{\geq}(\varepsilon) = \{a_j : \text{sim}(a_j, a_i) \geq \varepsilon\}$ and $S^i_{<}(\varepsilon) = \{a_j : \text{sim}(a_j, a_i) < \varepsilon\}$ are the partition of the arms for a given similarity threshold value $\varepsilon > 0$.

We define:

$$s_{j,i}(\varepsilon) = p(a_j | a_j \in S^i_{\geq}(\varepsilon)) = \frac{s_{j,i}}{\sum_k s_{k,i} [\![a_k \in S^i_{\geq}(\varepsilon)]\!]}$$

$$\bar{s}_{j,i}(\varepsilon) = p(a_j | a_j \in S^i_{<}(\varepsilon)) = \frac{s_{j,i}}{\sum_k s_{k,i} [\![a_k \in S^i_{<}(\varepsilon)]\!]},$$

Contextual Bandits For Query Recommendations

Given the currently playing arm a_i, our hypothesis is that the arms are sampled according to the joint distribution $p(a_j, a_k | a_i, \varepsilon) = p(a_j, a_k | (s_{j,i}, s_{k,i}) \geq \varepsilon \vee (s_{j,i}, s_{k,i}) < \varepsilon)$.

The marginal conditional probability can be seen as the normalized preference score for a specific arm to be played next.

Our next assumption is that there exists a utility function which maps these preference scores to utilities.

This assumption is similar to the Lipschitz assumption.

A theorem due to [**Ortega2010**] states that only the ln function satisfies the above requirements.

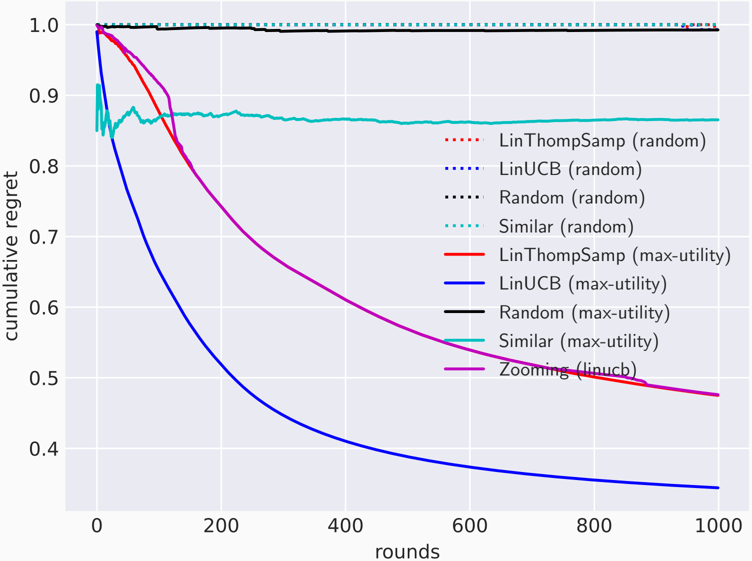
Contextual Bandits For Query Recommendations

This leads us to the following optimization problem for finding the candidate set.

$$\max_{\substack{\mathcal{C} \subseteq \mathcal{A}' \\ |\mathcal{C}| \leq k}} \log \left(g \left(\mathcal{C}, \{a_i\} \right) \right)$$

This optimization problem is submodular and it can be solved in lear time very efficiently using simple greedy algorithm

Contextual Bandits For Query Recommendations



Query Recommendations with Bandits

Adversarial Bandits with Self Supervised Experts

- We study the query recommendation problem in the adversarial setting.

Query Recommendations with Bandits

Adversarial Bandits with Self Supervised Experts

- We study the query recommendation problem in the adversarial setting.
- In the stochastic setting, we assume that there is a fixed query that the user will be interested in issuing with optimal utility.

Query Recommendations with Bandits

Adversarial Bandits with Self Supervised Experts

- We make use of the auto-regressive models as experts.

Query Recommendations with Bandits

Adversarial Bandits with Self Supervised Experts

- We make use of the auto-regressive models as experts.
- We train N auto-regressive models on the historical query logs.

Query Recommendations with Bandits

Parameters : parameter η

Initialization: $W_0 = 1, r_0 = 1, \mathcal{C}_0 = \emptyset, \hat{w}_i = 0$

for $t = 1, 2, \ldots T$ **do**

$q_t = get\_currently\_executing\_query()$

$\mathcal{C}^t = \bigcup_{e \in \mathcal{E}} query\_by\_score\_threshold(\mathcal{Q}, q_t, \epsilon)$

$\mathcal{C}_t = \mathcal{C}^t \cup \mathcal{C}_{t-1}$

$w_{i,t} = \frac{\eta \hat{r}_{t-1}[i \in \mathcal{C}_t \setminus \mathcal{C}_{t-1}]}{(1-\eta)|\mathcal{C}_{t-1}||\mathcal{C}_t \setminus \mathcal{C}_{t-1}|} + \hat{w}_{i,t} 1[i \in \mathcal{C}_{t-1}]$

$p_{i,t} = \frac{(1-\eta) w_{i,t} 1[i \in \mathcal{C}_t \setminus \mathcal{C}_{t-1}] + (1-\eta) w_{i,t} 1[i \in \mathcal{C}_{t-1}]}{\sum_i w_{i,t} 1[i \in \mathcal{C}_t]} + \frac{\eta}{|C_t|}$

Draw I_t from $\mathcal{C}_t$ according to p_t

Play I_t and observe gain $r_{I_t,t}$

$r_t = r_{t-1} + r_{I_t,t}$

Calculate the pseudo-reward values $\hat{r}_{i,t} = \frac{r_{i,t}[[I_{i,t}=1]]}{p_{i,t}}$

$\hat{w}_{i,t+1} = w_{i,t} \exp\left(\eta \hat{r}_{i,t}\right)$ for $j \in \mathcal{C}_t$

end

Query Recommendations with Bandits

Regret Analysis
Taking $\eta = \frac{1}{\sqrt{T|\mathcal{C}_T|}}$, and considering only the leading terms, per-round regret is

$$L_T - L^\star \leq O\left(\sqrt{\frac{|\mathcal{C}_T|}{T}}\right) \tag{1}$$

This regret matches with the regret that can be achieved if the size of $\mathcal{C}$ is fixed and known in advance.

Thank You

Questions..........?