Multi-armed Bandits in Practice

Alexandr Vorobyev

Yandex

April 27, 2017

Outline

1. Classic Multi-Armed Bandit Problem
2. Contextual Multi-Armed Bandit Problem

Outline

1. Classic Multi-Armed Bandit Problem
2. Contextual Multi-Armed Bandit Problem

Multi-Armed Bandit Problem

Multi-Armed Bandit Problem: Formalization

Setting

- A finite set of **arms** $\{a_1, a_2, \ldots, a_k\}$
- We have T **steps (trials)**. At each step t:
  - We choose an arm $a_{j(t)}$.
  - We observe a **reward** $R_{j(t)}$ – **random** value, $ER_{j(t)} = r_{j(t)}$ ($R_{j(1)}, \ldots, R_{j(T)}$ are independent).

Multi-Armed Bandit Problem: Formalization

Setting

- A finite set of **arms** $\{a_1, a_2, \ldots, a_k\}$
- We have T **steps (trials)**. At each step t:
  - We choose an arm $a_{j(t)}$.
  - We observe a **reward** $R_{j(t)}$ – **random** value, $ER_{j(t)} = r_{j(t)}$ $(R_{j(1)}, \ldots, R_{j(T)}$ are independent).

Objective

Maximize the expectation of **cumulative reward** $\sum_{t=1}^{T} R_{j(t)}$.

Multi-Armed Bandit Problem: Intuition

Let's try to play

- At start: choose an arm arbitrarily.

Let's try to play

- At start: choose an arm arbitrarily.
- After several turns:
  - We need **to gain** reward, so we tend to choose **arms with higher estimates of expectation** r_j.

Multi-Armed Bandit Problem: Intuition

Let's try to play

- At start: choose an arm arbitrarily.
- After several turns:
    - We need **to gain** reward, so we tend to choose **arms with higher estimates of expectation** r_j.
    - We need **to explore** arms, so we tend to choose **arms with few statistics** on their observed rewards.

Let's try to play

- At start: choose an arm arbitrarily.
- After several turns:
  - We need **to gain** reward, so we tend to choose **arms with higher estimates of expectation** r_j.
  - We need **to explore** arms, so we tend to choose **arms with few statistics** on their observed rewards.

The problem of balancing between these two goals is known as **exploration–exploitation dilemma.**

Multi-Armed Bandit Problem

Most important case

which often arises in practice:

- Reward R_j is a **Bernoulli random variable**:

$$P(R_j = 1) = r_j \quad , P(R_j = 0) = 1 - r_j, ER_j = r_j.$$

- Parameter r_j has **uniform prior distribution** on $[0, 1]$.

Multi-Armed Bandit Problem: practical examples

Advertisement

- Problem setting: to **choose an ad** from the database **to show to a user in some known context**.
- Step = appearance of the context.
- Arm = ad.
- $R_{j(t)} = 1$, if the user clicked the ad $a_{j(t)}$, $= 0$ otherwise.
- r_j is known as CTR (click-through rate).

Multi-Armed Bandit Problem: practical examples

Information retrieval

- Problem setting: to **choose a document** from the database **to show to a user at the top position to some known query**.
- Step = an issue of the query.
- Arm = document.
- $R_{j(t)} = 1$, if the user clicked the document, $= 0$ otherwise.
- r_j is known as CTR at position 1.

Multi-Armed Bandit Problem: algorithms

Algorithms for Bernoulli R_j

Define an appropriate scoring $S_t(a)$, choose $j(t) = \mathrm{argmax}_j\, S_t(a_j)$.

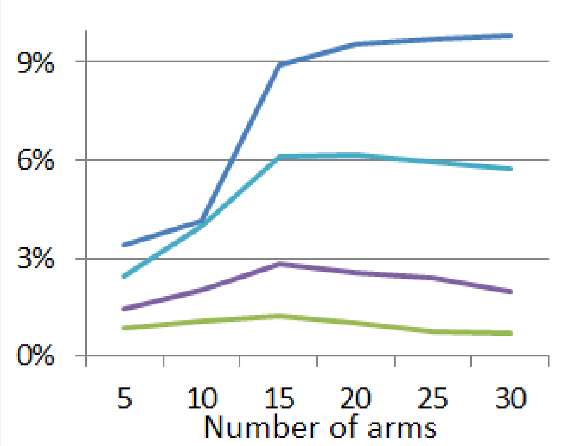
- **UCB-1**: $S_t(a_j) = \widehat{r_{t,j}} + \alpha\sqrt{\frac{2\ln t}{N_{t-1,j}}}$, where
  - $\widehat{r}_{a_j,t} = \frac{S_{t-1,j}}{N_{t-1,j}}$, $S_{t-1,j}$ ($N_{t-1,j}$) is the number of succesful (all) trials of a_j.
  - α is an exploration parameter (to be fitted to T).

Algorithms for Bernoulli R_j

Define an appropriate scoring $S_t(a)$, choose $j(t) = \mathrm{argmax}_j\, S_t(a_j)$.

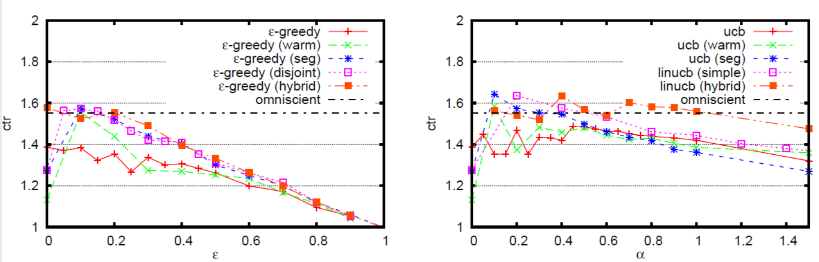
- **UCB-1**: $S_t(a_j) = \widehat{r_{t,j}} + \alpha\sqrt{\frac{2\ln t}{N_{t-1,j}}}$, where
  - $\widehat{r}_{a_j,t} = \frac{S_{t-1,j}}{N_{t-1,j}}$, $S_{t-1,j}$ ($N_{t-1,j}$) is the number of succesful (all) trials of a_j.
  - α is an exploration parameter (to be fitted to T).
- **Bayesian approach**
  - Calculate posterior distribution of r_j: $p_{t,j}(r) \propto r^{S_{t-1,j}}(1-r)^{N_{t-1,j}-S_{t-1,j}}p_{0,j}(r)$
  - **Thompson sampling** algorithm: $S_t(a_j)$ is a sample from $p_{t,j}(r)$.
  - **Bayesian-UCB** algorithm: $S_t(a_j)$ is a $\alpha(t)$-quantile of $p_{t,j}(r)$, $\alpha(t)$ is an exploration parameter (to be fitted to T).

Number of arms



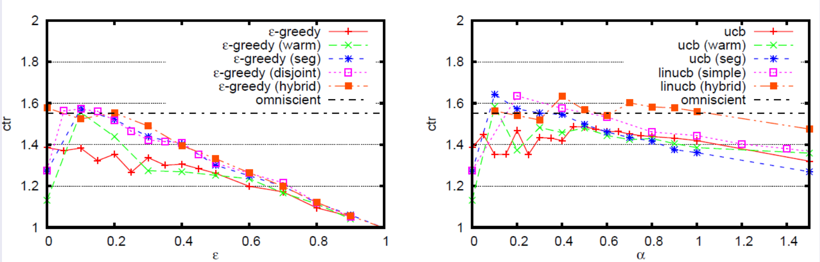
Practical Aspects

Exploration rate



Practical Aspects

Exploration rate



Bayes UCB: $\alpha(t) = 1 - \frac{1}{t \cdot (\log n)^c}$

Outline

1. Classic Multi-Armed Bandit Problem
2. Contextual Multi-Armed Bandit Problem

Contextual Multi-Armed Bandit Problem

Setting

- A set **A** of **arms**
  - finite: documents, objects for recommendation;
  - continuum: vectors of formula's coefficients.
- A set **C** of **contexts** (query, user, location, position, upper documents) with a distribution P_C on it.
- Arm-context pair $(a, c) \longleftrightarrow$ a feature vector $x_{a,c} \in \mathbb{R}^d$.

Contextual Multi-Armed Bandit Problem

Setting

- A set **A** of **arms**
  - finite: documents, objects for recommendation;
  - continuum: vectors of formula's coefficients.
- A set **C** of **contexts** (query, user, location, position, upper documents) with a distribution P_C on it.
- Arm-context pair $(a, c) \longleftrightarrow$ a feature vector $x_{a,c} \in \mathbb{R}^d$.
- We have T **steps (trials)**. At each step t:
  - We observe a context $c(t) \in \mathbf{C}$ sampled from P_C.
  - We choose an arm $a(t) \in \mathbf{A}$.
  - We observe a **reward** R_t – a realization of a r.v. $R(x_{a(t),c(t)})$, $ER(a(t), c(t)) = r(x_{a(t),c(t)})$, $R_1, \ldots, R_T$ are independent.

Contextual Multi-Armed Bandit Problem

Setting

- A set **A** of **arms**
  - finite: documents, objects for recommendation;
  - continuum: vectors of formula's coefficients.
- A set **C** of **contexts** (query, user, location, position, upper documents) with a distribution P_C on it.
- Arm-context pair $(a, c) \longleftrightarrow$ a feature vector $x_{a,c} \in \mathbb{R}^d$.
- We have T **steps (trials)**. At each step t:
  - We observe a context $c(t) \in \mathbf{C}$ sampled from P_C.
  - We choose an arm $a(t) \in \mathbf{A}$.
  - We observe a **reward** R_t – a realization of a r.v. $R(x_{a(t),c(t)})$, $ER(a(t), c(t)) = r(x_{a(t),c(t)})$, $R_1, \ldots, R_T$ are independent.

Objective

Maximize the expectation of **cumulative reward** $\sum_{t=1}^{T} R_t$.

What is new?

- We believe that $r(x)$ is continuous, Lipschitzian or smooth function on $\mathbb{R}^d$.
- A challenge: aggregate information over features $x_{a,c}$.

Contextual Multi-Armed Bandit Problem

What is new?

- We believe that $r(x)$ is continuous, Lipschitzian or smooth function on $\mathbb{R}^d$.
- A challenge: aggregate information over features $x_{a,c}$.

General Idea

At step t, for each arm a_j, try to **estimate** $r(a(t), c(t))$ **and confidence** in this estimate (variance, confidence bounds) on the features $x_{a(t),c(t)}$ and the history of observations $\{x_{j(\tau),c(\tau)}, R_t\}_{\tau=1,\dots,T}$.

Adaptation of the classical MAB (for contexts)

[Hoffman; Radlinski; Sloan, Wang; our paper on WWW'15...]

- Divide **C** into regions:
  region=query (web search), region=user (recommendations).
- Run a separate bandit for each region.

Known approaches

Adaptation of the classical MAB (for contexts)

[Hoffman; Radlinski; Sloan, Wang; our paper on WWW'15...]

- Divide **C** into regions:
  region=query (web search), region=user (recommendations).
- Run a separate bandit for each region.

Analysis

+ The smaller region – the more specific feedback.
- The smaller region – more information ignored, the lower learning rate.

Effective for **small regions with a lot of feedback** (frequent queries, active users).

Known approaches

Context tree

[Slivkins, Radlinski, “zooming algorithm”]

- Fix tree structure of $\mathbf{A} \times \mathbf{C}$ (e.g., topical taxonomy of documents).

Known approaches

Context tree

[Slivkins, Radlinski, “zooming algorithm”]

- Fix tree structure of $\mathbf{A} \times \mathbf{C}$ (e.g., topical taxonomy of documents).
- At each step:
  - Some set of nodes divides $\mathbf{A} \times \mathbf{C}$ into regions.
  - Choose one of nodes, choose an arm from it arbitrary, refer reward to the node.
  - Collected sufficient information for a node $\Longrightarrow$ substitute it by its children, use the information as prior for them.

Known approaches

Context tree

[Slivkins, Radlinski, "zooming algorithm"]

- Fix tree structure of $\mathbf{A} \times \mathbf{C}$ (e.g., topical taxonomy of documents).
- At each step:
  - Some set of nodes divides $\mathbf{A} \times \mathbf{C}$ into regions.
  - Choose one of nodes, choose an arm from it arbitrary, refer reward to the node.
  - Collected sufficient information for a node $\Longrightarrow$ substitute it by its children, use the information as prior for them.

Analysis

\+ Adaptive width of regions.

\- Threshold-based aggregation of feedback.

\- No approach to **construct a tree reflecting proximity of** $r(a, c)$ **over** (a, c).

Known approaches

LinUCB: linear regression

[Lihong Li, Langford, Schapire]

- Linear regression for $r_{a,c}$
  - Disjoint model — aggregate over contexts: $E(r_{a,c}|x_{a,c}) = x_{a,c}^T \theta_a$.

Known approaches

LinUCB: linear regression

[Lihong Li, Langford, Schapire]

- Linear regression for $r_{a,c}$
  - Disjoint model — aggregate over contexts: $E(r_{a,c}|x_{a,c}) = x_{a,c}^T \theta_a$.
  - Hybrid model — aggregate over contexts and arms: $E(r_{a,c}|x_{a,c}) = x_{a,c}^T \theta_a + x_{a,c}^T \xi$.

LinUCB: linear regression

[Lihong Li, Langford, Schapire]

- Linear regression for $r_{a,c}$
  - Disjoint model — aggregate over contexts: $E(r_{a,c}|x_{a,c}) = x_{a,c}^T \theta_a$.
  - Hybrid model — aggregate over contexts and arms: $E(r_{a,c}|x_{a,c}) = x_{a,c}^T \theta_a + x_{a,c}^T \xi$.
  - Add $x_{a,c}^T \eta_g$ for any division of $\mathbf{A} \times \mathbf{C}$ into regions g?
- $R_{a,c} = E(r_{a,c}|x_{a,c}) + \epsilon,\ E\epsilon = 0, |\epsilon| < b$.

Known approaches

LinUCB: linear regression

[Lihong Li, Langford, Schapire]

- Linear regression for $r_{a,c}$
    - Disjoint model — aggregate over contexts: $E(r_{a,c}|x_{a,c}) = x_{a,c}^T \theta_a$.
    - Hybrid model — aggregate over contexts and arms: $E(r_{a,c}|x_{a,c}) = x_{a,c}^T \theta_a + x_{a,c}^T \xi$.
    - Add $x_{a,c}^T \eta_g$ for any division of $\mathbf{A} \times \mathbf{C}$ into regions g?
- $R_{a,c} = E(r_{a,c}|x_{a,c}) + \epsilon,\ E\epsilon = 0, |\epsilon| < b$.
- At each step, obtain an upper confidence bound for $r_{a,c}$

Known approaches

LinUCB: linear regression

[Lihong Li, Langford, Schapire]

- Linear regression for $r_{a,c}$
  - Disjoint model — aggregate over contexts: $E(r_{a,c}|x_{a,c}) = x_{a,c}^T\theta_a$.
  - Hybrid model — aggregate over contexts and arms: $E(r_{a,c}|x_{a,c}) = x_{a,c}^T\theta_a + x_{a,c}^T\xi$.
  - Add $x_{a,c}^T\eta_g$ for any division of $\mathbf{A} \times \mathbf{C}$ into regions g?
- $R_{a,c} = E(r_{a,c}|x_{a,c}) + \epsilon,\ E\epsilon = 0, |\epsilon| < b$.
- At each step, obtain an upper confidence bound for $r_{a,c}$

Analysis

\+ Learning dependence between clicks and features.

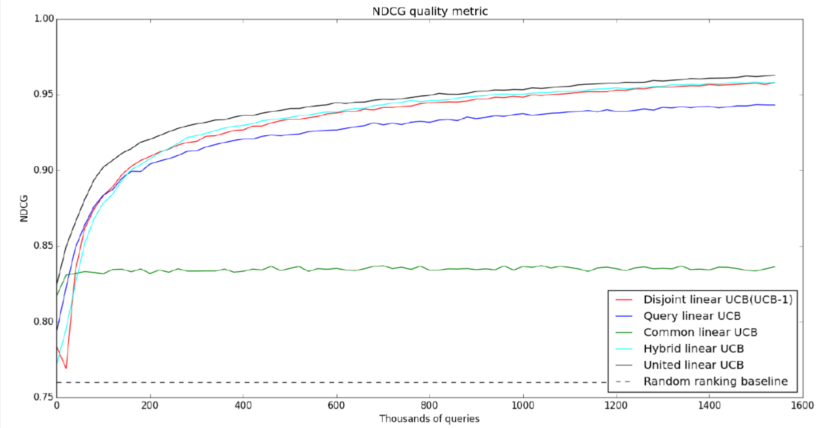
\- Linearity.

\- An upper bound is not in $[0, 1]$, logistic model is more preferable: $E(r_{a,c}|x_{a,c}) = \frac{1}{1+e^{-x_{a,c}^T\theta_a}}$

Disjoint LinUCB algorithm

Inputs: $\alpha \in \mathbb{R}_+$
for $t = 1, 2, 3, \ldots, T$ **do**
  Observe features of all arms $a \in \mathcal{A}_t$: $\mathbf{x}_{t,a} \in \mathbb{R}^d$
  for all $a \in \mathcal{A}_t$ **do**
    if a is new **then**
      $\mathbf{A}_a \leftarrow \mathbf{I}_d$ (d-dimensional identity matrix)
      $\mathbf{b}_a \leftarrow \mathbf{0}_{d \times 1}$ (d-dimensional zero vector)
    end if
    $\hat{\boldsymbol{\theta}}_a \leftarrow \mathbf{A}_a^{-1} \mathbf{b}_a$
    $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^\top \mathbf{x}_{t,a} + \alpha \sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}}$
  end for
  Choose arm $a_t = \arg\max_{a \in \mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_t
  $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t} \mathbf{x}_{t,a_t}^\top$
  $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t \mathbf{x}_{t,a_t}$
end for

Experiments in web search



Gaussian process

[Vanchinathan, Nicolic, De Bona]

- $E(r_{a,c}|x_{a,c}) = f(x_{a,c})$, $f(x)$ is a Gaussian process.

Known approaches

Gaussian process

[Vanchinathan, Nicolic, De Bona]

- $E(r_{a,c}|x_{a,c}) = f(x_{a,c})$, $f(x)$ is a Gaussian process.
- Set prior $Ef(x) = \mu(x), Cov(f(x), f(x')) = k(x, x')$.

Gaussian process

[Vanchinathan, Nicolic, De Bona]

- $E(r_{a,c}|x_{a,c}) = f(x_{a,c})$, $f(x)$ is a Gaussian process.
- Set prior $Ef(x) = \mu(x), Cov(f(x), f(x')) = k(x, x')$.
- $k(x, x') = x^T \Sigma x'$ provides the linear regression.

Gaussian process

[Vanchinathan, Nicolic, De Bona]

- $E(r_{a,c}|x_{a,c}) = f(x_{a,c})$, $f(x)$ is a Gaussian process.
- Set prior $Ef(x) = \mu(x)$, $Cov(f(x), f(x')) = k(x, x')$.
- $k(x, x') = x^T \Sigma x'$ provides the linear regression.
- $R_{a,c} = E(r_{a,c}|x_{a,c}) + \epsilon$, ϵ is a standard error.

Gaussian process

[Vanchinathan, Nicolic, De Bona]

- $E(r_{a,c}|x_{a,c}) = f(x_{a,c})$, $f(x)$ is a Gaussian process.
- Set prior $Ef(x) = \mu(x)$, $Cov(f(x), f(x')) = k(x, x')$.
- $k(x, x') = x^T \Sigma x'$ provides the linear regression.
- $R_{a,c} = E(r_{a,c}|x_{a,c}) + \epsilon$, ϵ is a standard error.
- At each step, obtain normal posterior distribution for $f(x_{a,c})$.

$$\mu_t(\mathbf{s}, \mathbf{z}) = \mathbf{k}_t(\mathbf{s}, \mathbf{z})^T (\mathbf{K}_t + \mathbb{I})^{-1} \bar{\mathbf{y}}_t, \tag{3}$$

$$\sigma_t^2(\mathbf{s}, \mathbf{z}) = \kappa((\mathbf{s}, \mathbf{z}), (\mathbf{s}, \mathbf{z})) - \mathbf{k}_t(\mathbf{s}, \mathbf{z})^T (\mathbf{K}_t + \mathbb{I})^{-1} \mathbf{k}_t(\mathbf{s}, \mathbf{z}), \tag{4}$$

where $\mathbf{k}_t(\mathbf{s}, \mathbf{z}) = [\kappa((\mathbf{s}_1, \mathbf{z}_1), (\mathbf{s}, \mathbf{z})), \ldots, \kappa((\mathbf{s}_t, \mathbf{z}_t), (\mathbf{s}, \mathbf{z}))]^T$ and $\mathbf{K}_t$ is the positive semi-definite kernel matrix such that $\mathbf{K}_{t,i,j} = [\kappa((\mathbf{s}_i, \mathbf{z}_i), (\mathbf{s}_j, \mathbf{z}_j))]$.

Known approaches

Gaussian process

[Vanchinathan, Nicolic, De Bona]

- $E(r_{a,c}|x_{a,c}) = f(x_{a,c})$, $f(x)$ is a Gaussian process.
- Set prior $Ef(x) = \mu(x)$, $Cov(f(x), f(x')) = k(x, x')$.
- $k(x, x') = x^T \Sigma x'$ provides the linear regression.
- $R_{a,c} = E(r_{a,c}|x_{a,c}) + \epsilon$, ϵ is a standard error.
- At each step, obtain normal posterior distribution for $f(x_{a,c})$.

$$\mu_t(\mathbf{s}, \mathbf{z}) = \mathbf{k}_t(\mathbf{s}, \mathbf{z})^T (\mathbf{K}_t + \mathbb{I})^{-1} \bar{\mathbf{y}}_t, \quad (3)$$

$$\sigma_t^2(\mathbf{s}, \mathbf{z}) = \kappa((\mathbf{s}, \mathbf{z}), (\mathbf{s}, \mathbf{z})) - \mathbf{k}_t(\mathbf{s}, \mathbf{z})^T (\mathbf{K}_t + \mathbb{I})^{-1} \mathbf{k}_t(\mathbf{s}, \mathbf{z}), \quad (4)$$

where $\mathbf{k}_t(\mathbf{s}, \mathbf{z}) = [\kappa((\mathbf{s}_1, \mathbf{z}_1), (\mathbf{s}, \mathbf{z})), \ldots, \kappa((\mathbf{s}_t, \mathbf{z}_t), (\mathbf{s}, \mathbf{z}))]^T$ and $\mathbf{K}_t$ is the positive semi-definite kernel matrix such that $\mathbf{K}_{t,i,j} = [\kappa((\mathbf{s}_i, \mathbf{z}_i), (\mathbf{s}_j, \mathbf{z}_j))]$.

Analysis

+ General model (despite two normality assumptions).
- No approach to set $k(x, x')$.

Interesting? Have ideas how to do better?

You are welcome:

`yandex.ru/jobs/vacancies/interns/intern_researcher`