Multiarmed bandits

Thompson Sampling team

МФТИ x

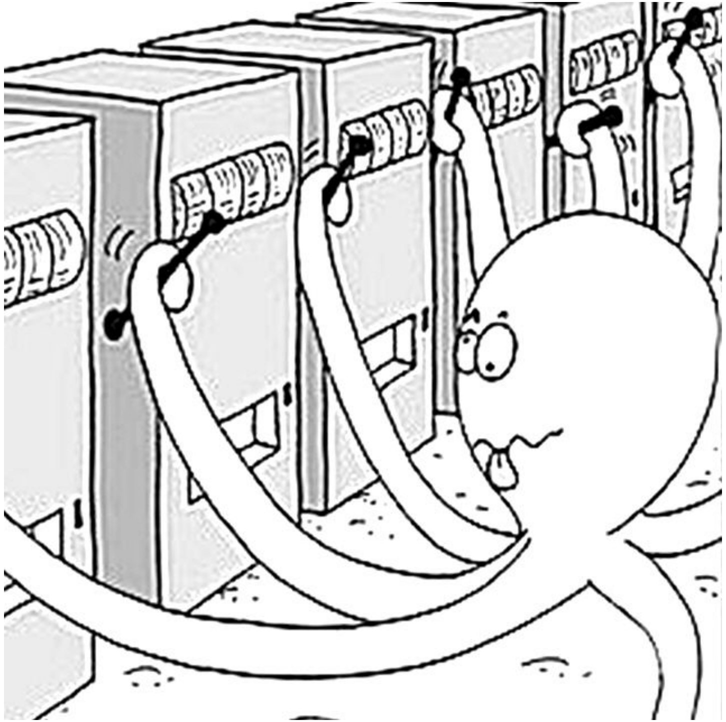
Общая постановка задачи многоруких бандитов

Агент взаимодействует с окружением, выбирая действия (руки) из множества доступных

Каждое действие приносит случайную награду с неизвестным распределением

Цель агента — максимизировать суммарную награду за определённое количество шагов, балансируя между исследованием (сбор информации о распределениях) и эксплуатацией (использование текущих знаний для максимизации награды)

Основная метрика — regret (разница между наградой оптимального действия и выбранного действия)



Неконтекстные бандиты

Особенности:

> Нет дополнительной информации (контекста) перед выбором действия

> Каждое действие имеет фиксированное, но неизвестное распределение наград

Примеры алгоритмов:

> ε-greedy

> UCB

> Thompson Sampling

Формально:

> На каждом шаге t агент выбирает действие a_t из N вариантов

> Награда r_t выбирается из распределения с математическим ожиданием μ_{a_t}

> Оптимальное действие: $a^* = argmax_a \mu_a$

> Regret на шаге t: $\Delta_t = \mu_{a^*} - \mu_{a_t}$

> Цель: минимизировать $R(T) = \sum\Delta_t,\ t \in [1, T]$

Контекстные бандиты

Особенности:

> Перед выбором действия агент получает контекст — вектор признаков, влияющий на награду

> Распределение наград зависит от контекста (например, линейно)

> Контекст может быть адаптивным (зависит от предыдущих действий агента)

Примеры алгоритмов:

> LinUCB

> LinTS

Формально:

> На шаге t:

Агент получает контекстные векторы $\{b_i(t)\} \in R^d$, $i \in [1, N]$ для каждого действия

Ожидаемая награда действия i: $E[r_i(t)] = b_i(t)^T\mu$, где $\mu \in R^d$ - неизвестный параметр

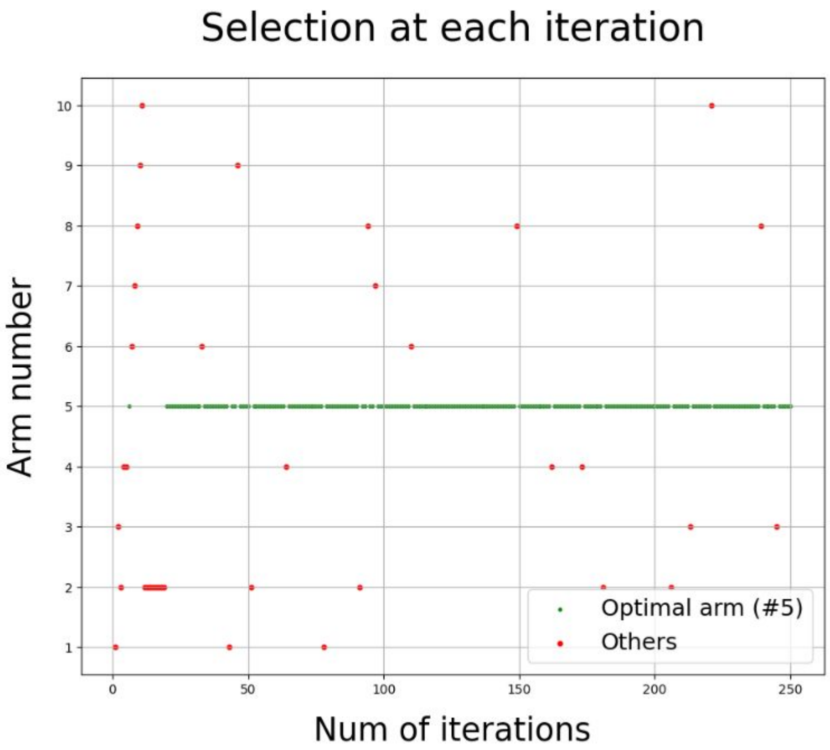
Агент выбирает действие a_t, получает награду $r_{a_t}(t)$

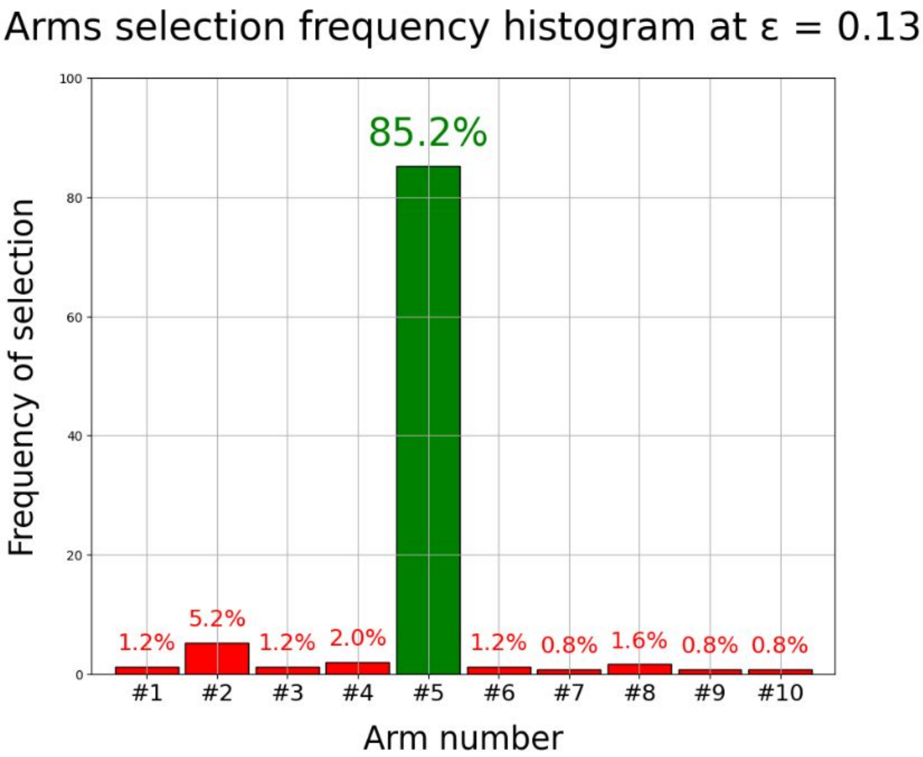
> Regret: $\Delta_t = max_i(b_i(t)^T\mu) - b_{a_t}(t)^T\mu$

> Цель: минимизировать $R(T) = \sum\Delta t$, $t \in [1, T]$

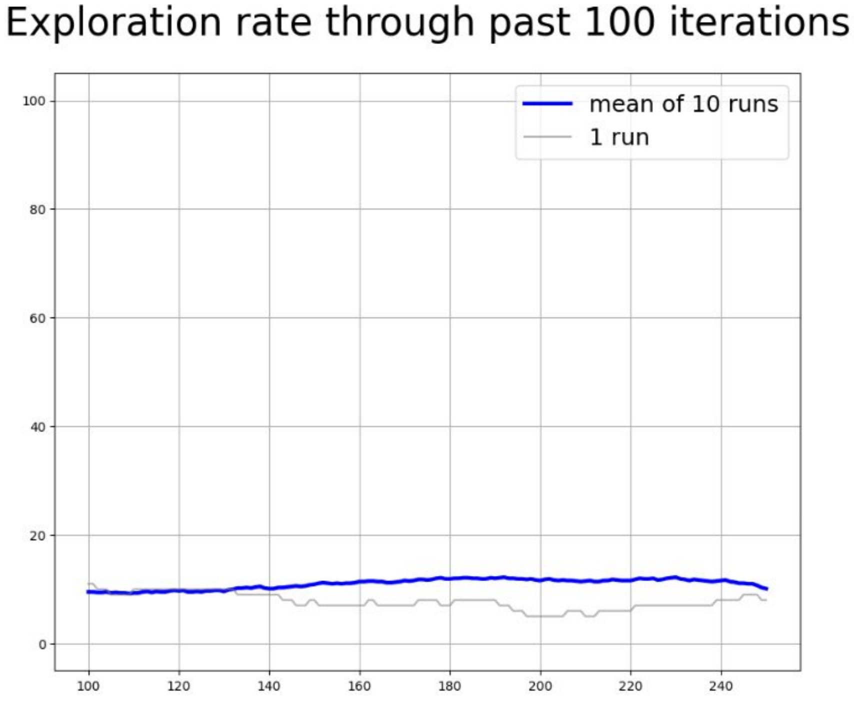
Сравнение алгоритмов

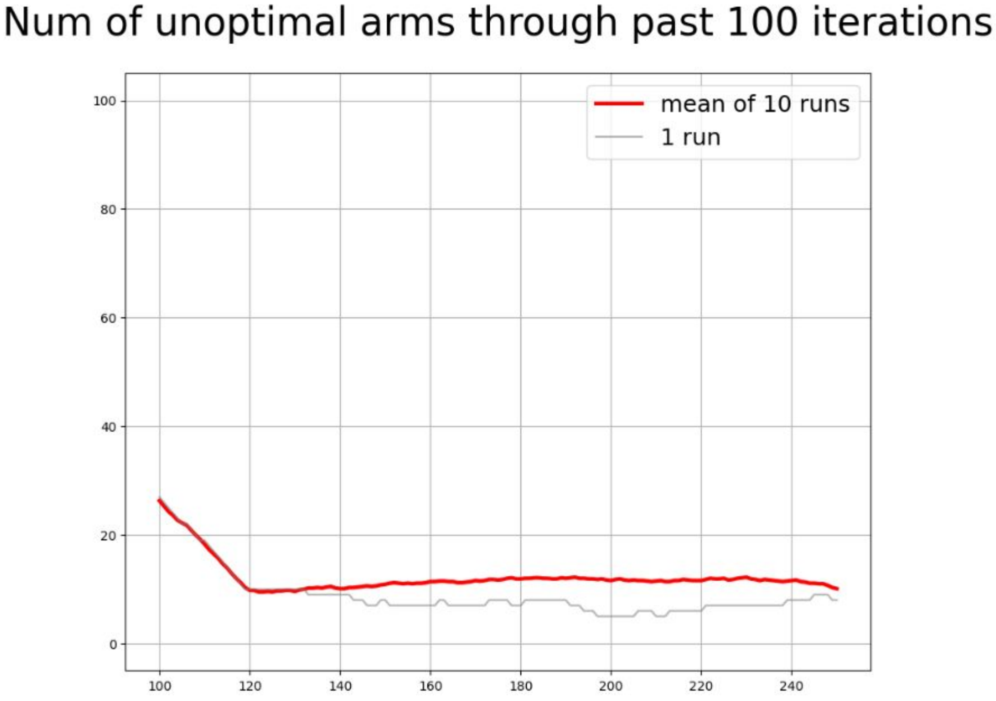
Epsilon-Greedy-Constant



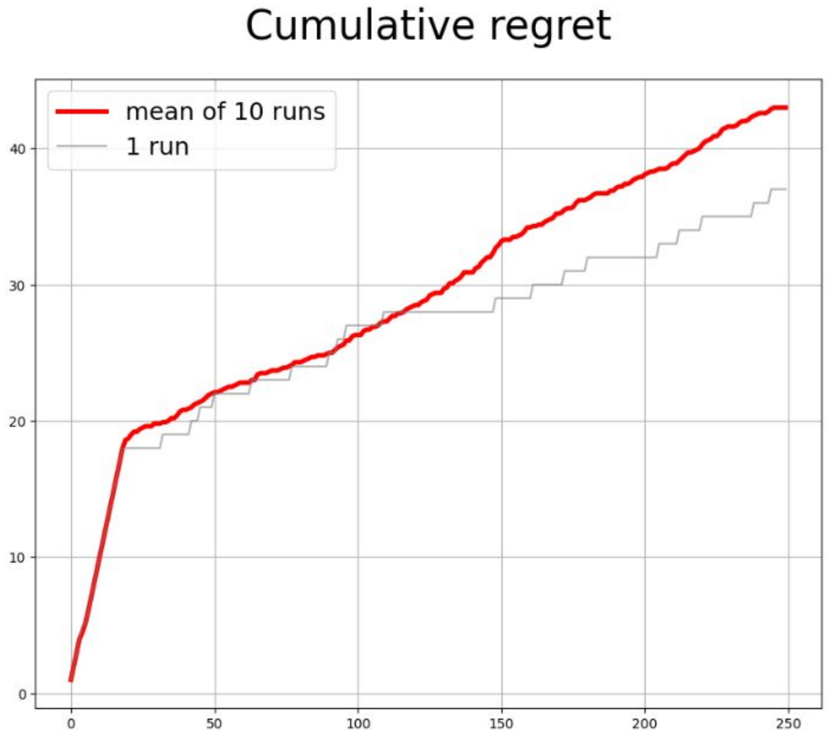


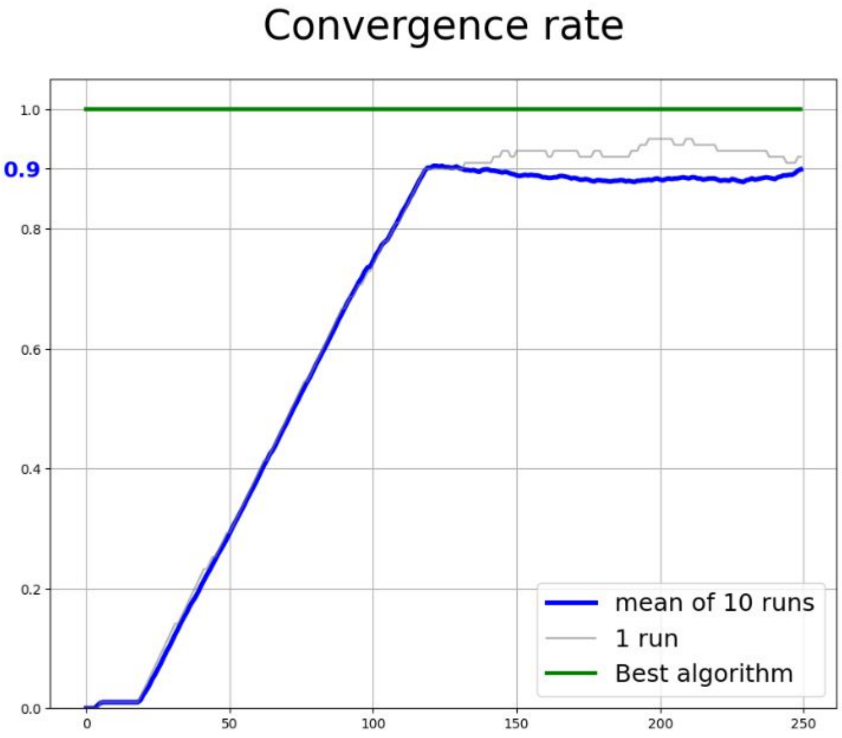
Epsilon-Greedy-Constant



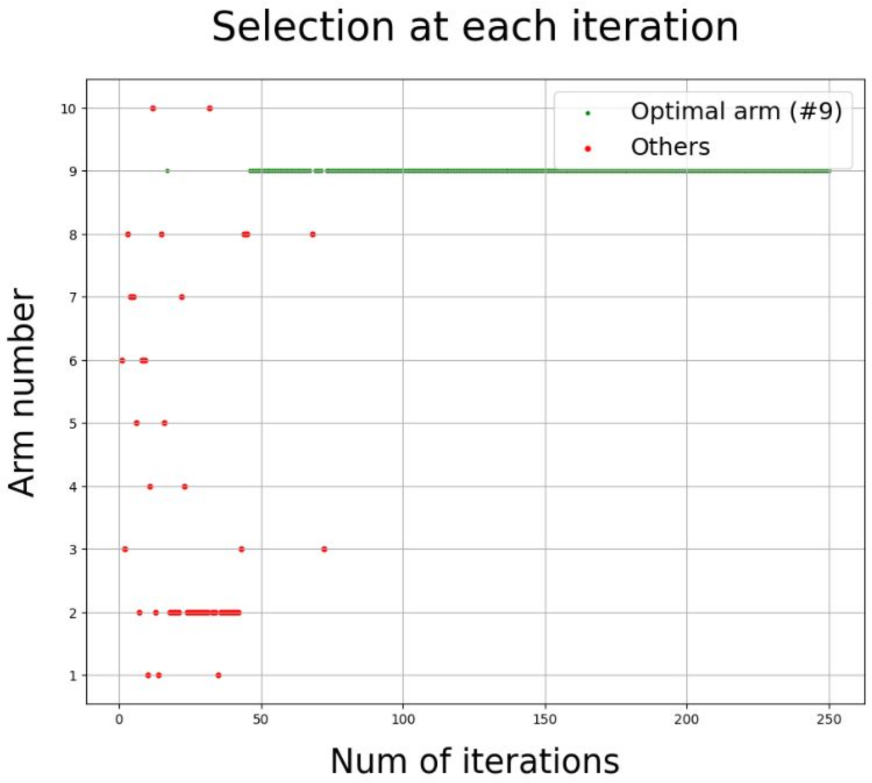


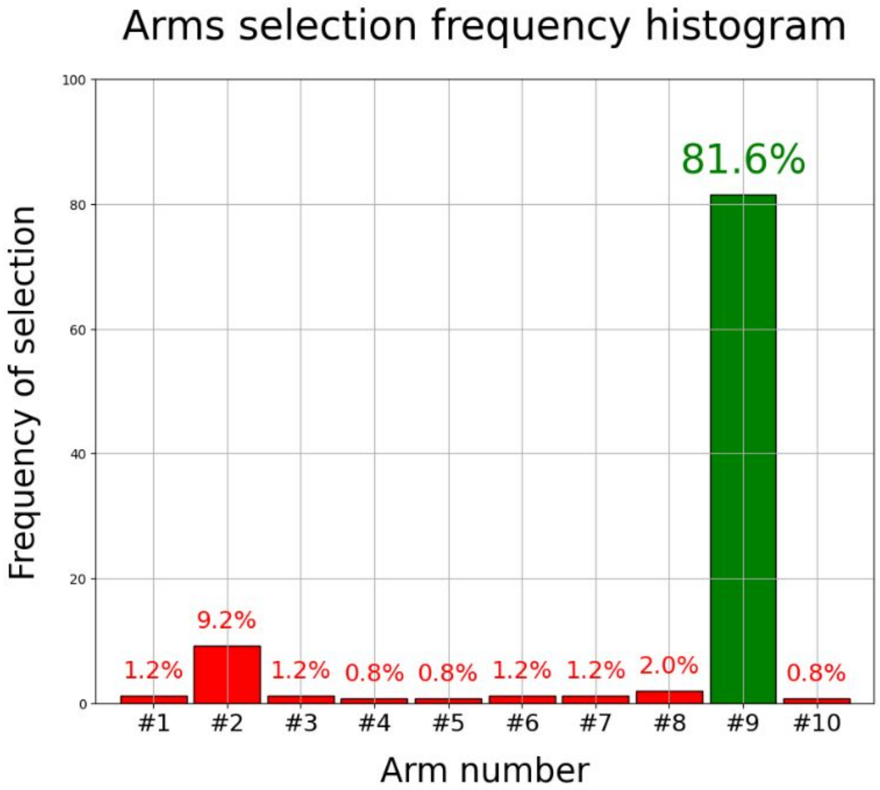
Epsilon-Greedy-Constant





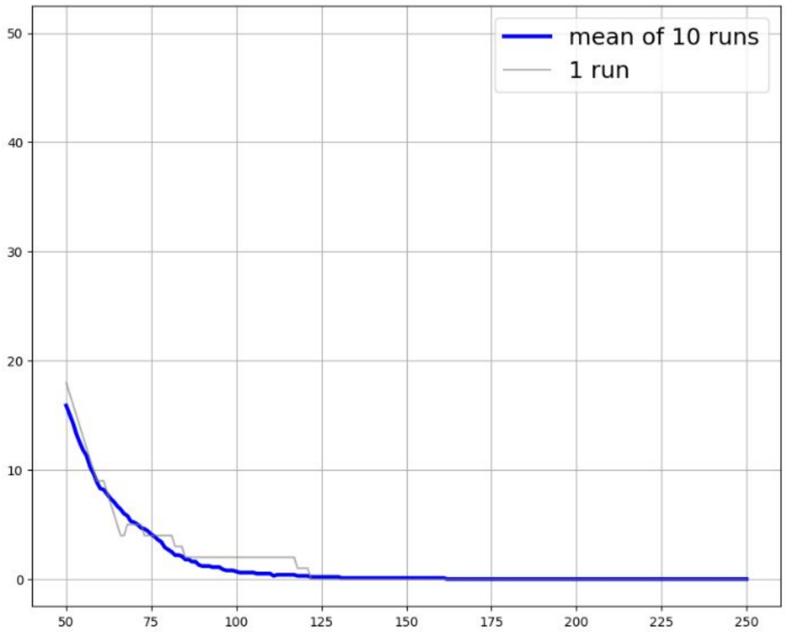
Value-Difference-Based-Epsilon



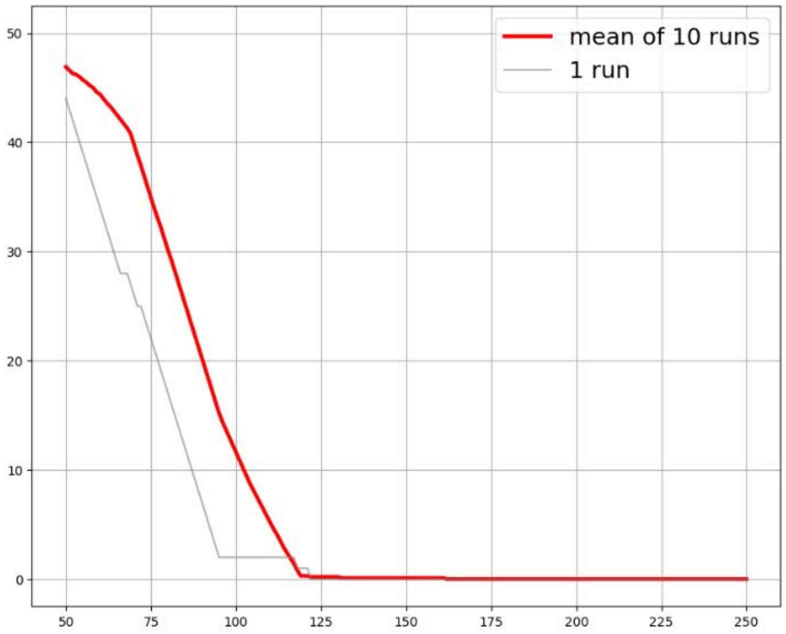


Value-Difference-Based-Epsilon

Exploration rate through past 50 iterations

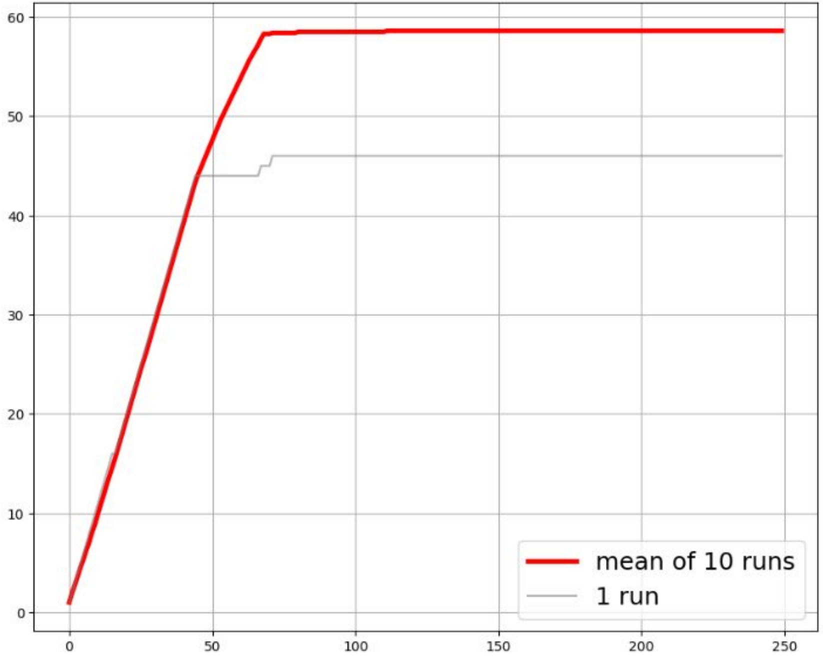


Num of unoptimal arms through past 50 iterations

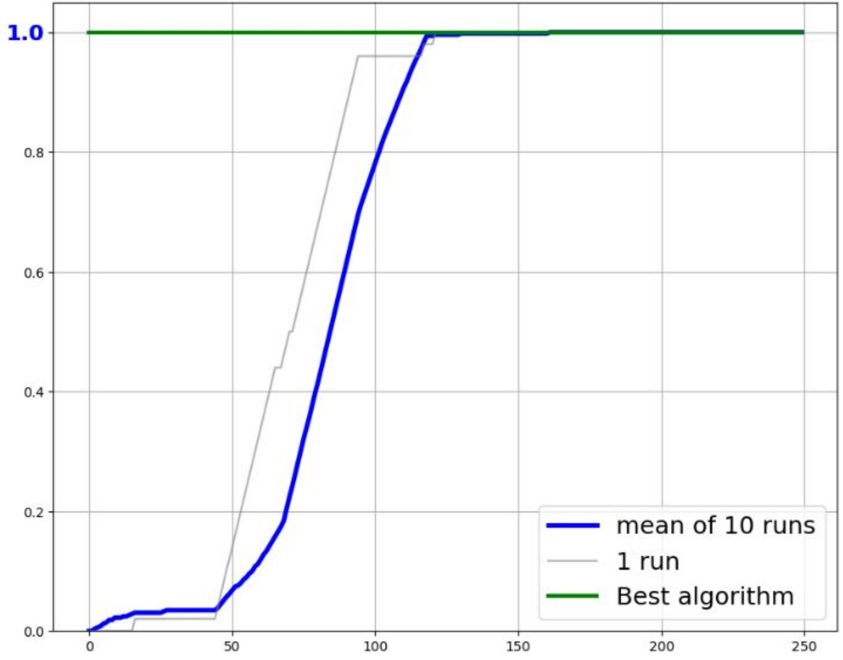


Value-Difference-Based-Epsilon

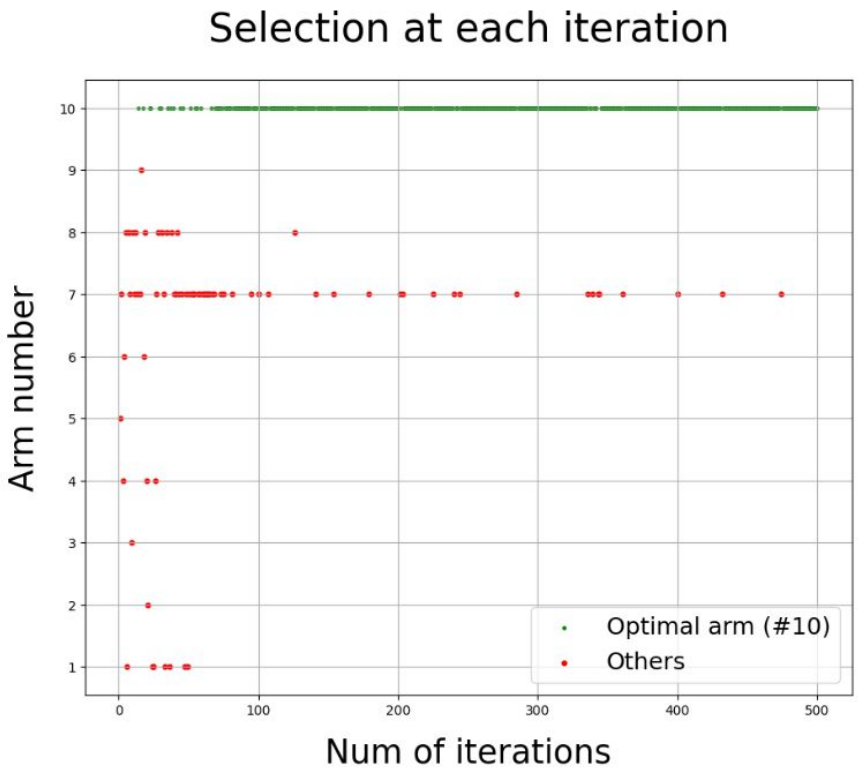
Cumulative regret

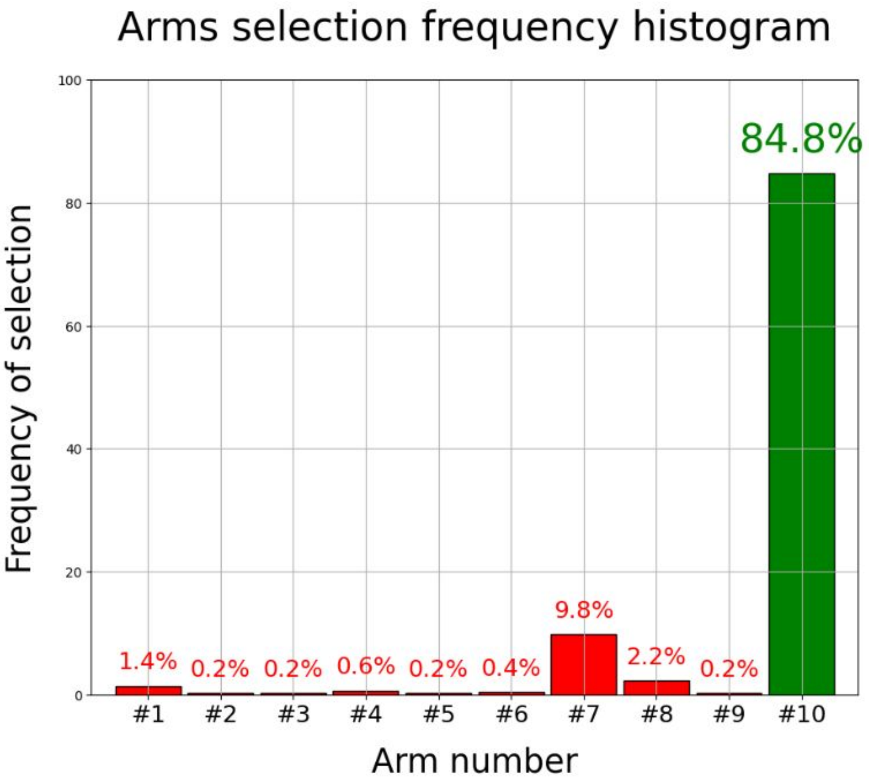


Convergence rate

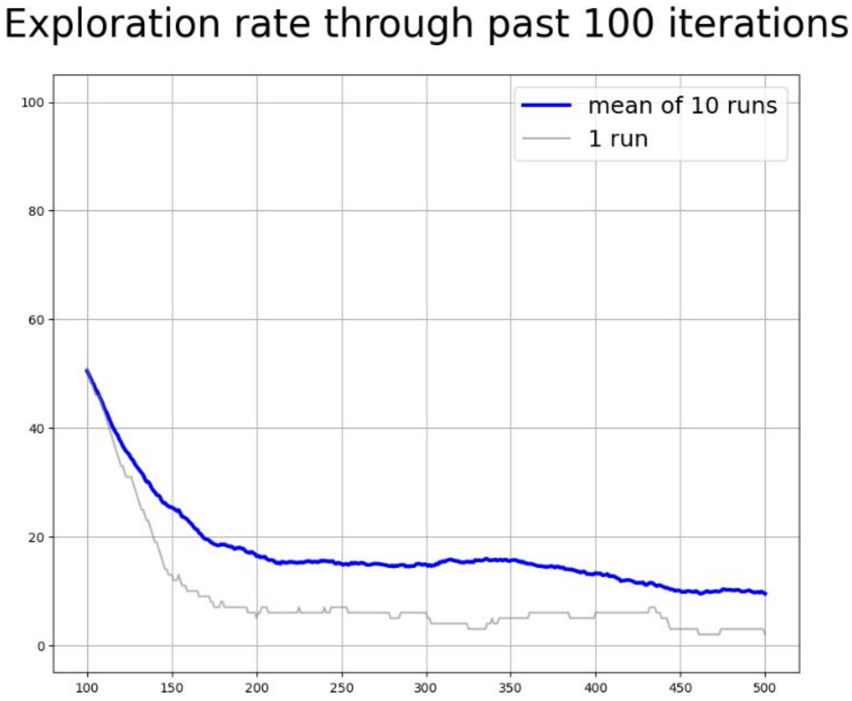


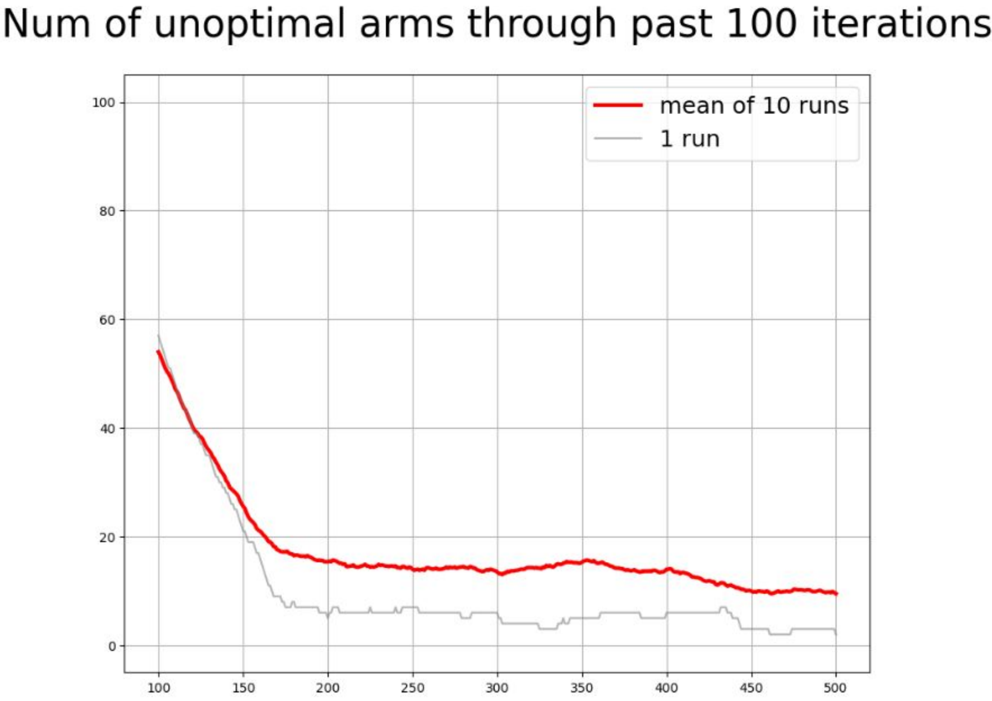
SoftMax


Selection at each iteration
Arm number
Num of iterations
10
9
8
7
6
5
4
3
2
1
0
100
200
300
400
500
Optimal arm (#10)
Others


Arms selection frequency histogram
Frequency of selection
Arm number
100
80
60
40
20
0
84.8%
9.8%
1.4%
0.2%
0.2%
0.6%
0.2%
0.4%
2.2%
0.2%
#1
#2
#3
#4
#5
#6
#7
#8
#9
#10

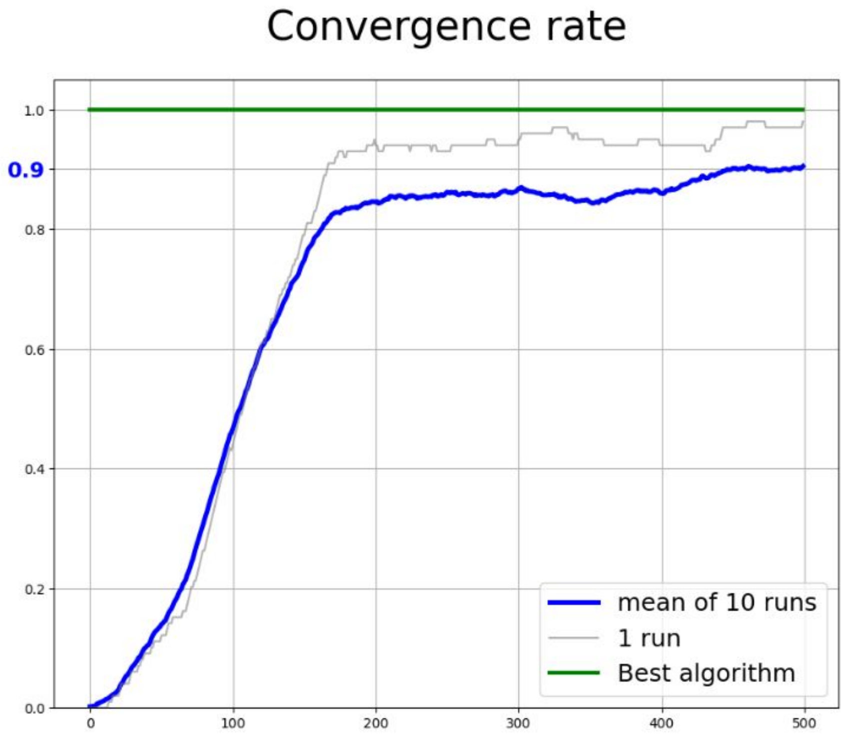
SoftMax


Exploration rate through past 100 iterations
mean of 10 runs
1 run
0
20
40
60
80
100
100
150
200
250
300
350
400
450
500

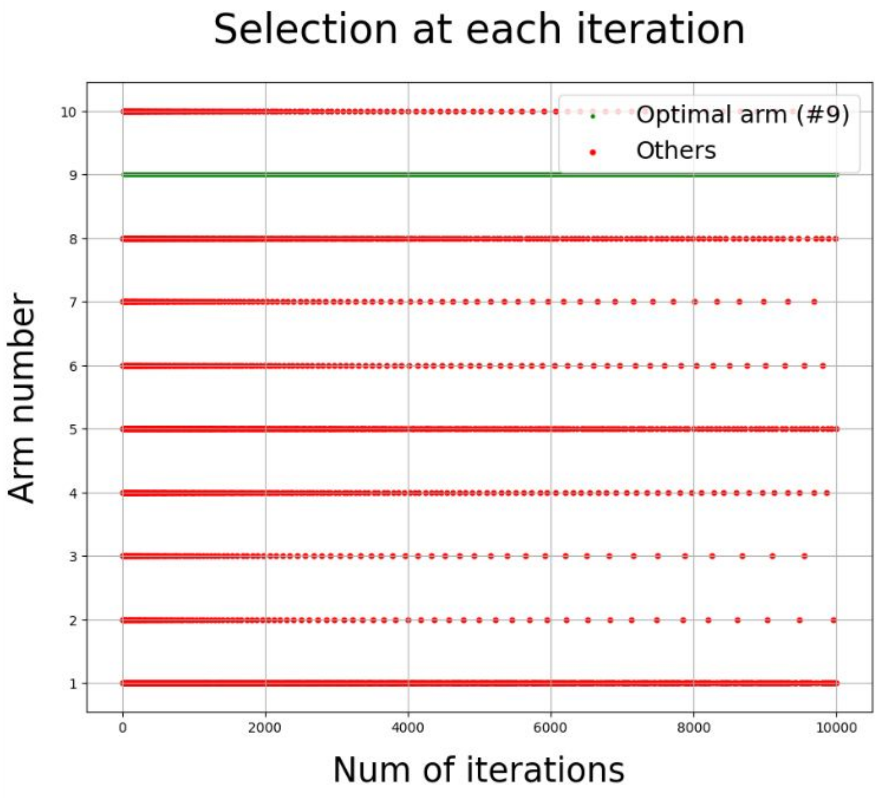

Num of unoptimal arms through past 100 iterations
mean of 10 runs
1 run
0
20
40
60
80
100
100
150
200
250
300
350
400
450
500

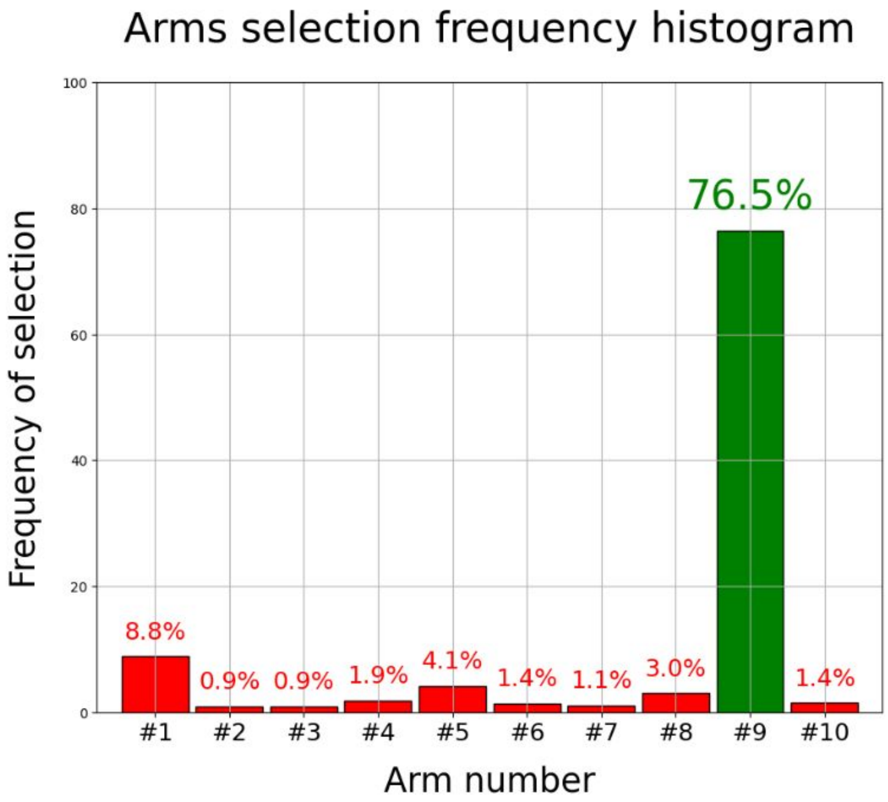
SoftMax


Cumulative regret
mean of 10 runs
1 run
100
80
60
40
20
0
0
100
200
300
400
500


Convergence rate
1.0
0.9
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500
mean of 10 runs
1 run
Best algorithm

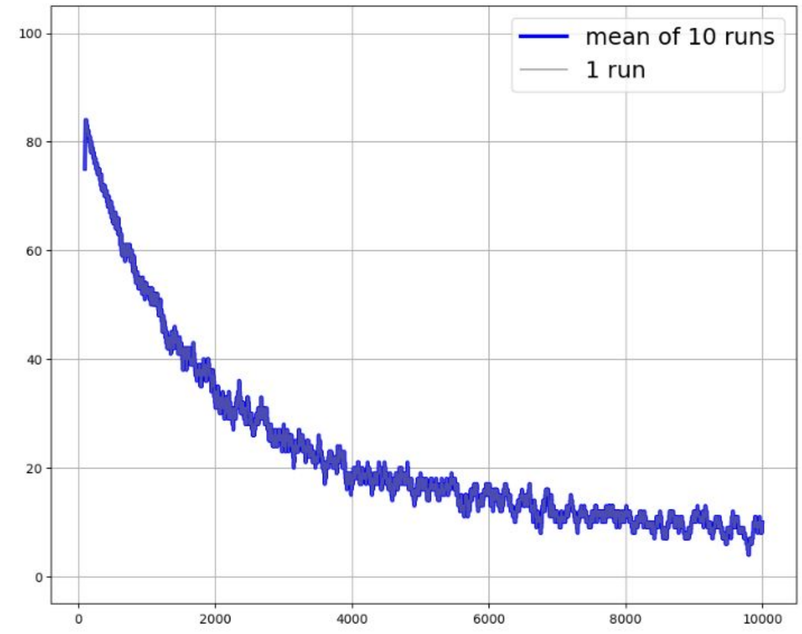
Upper Confidence Bound



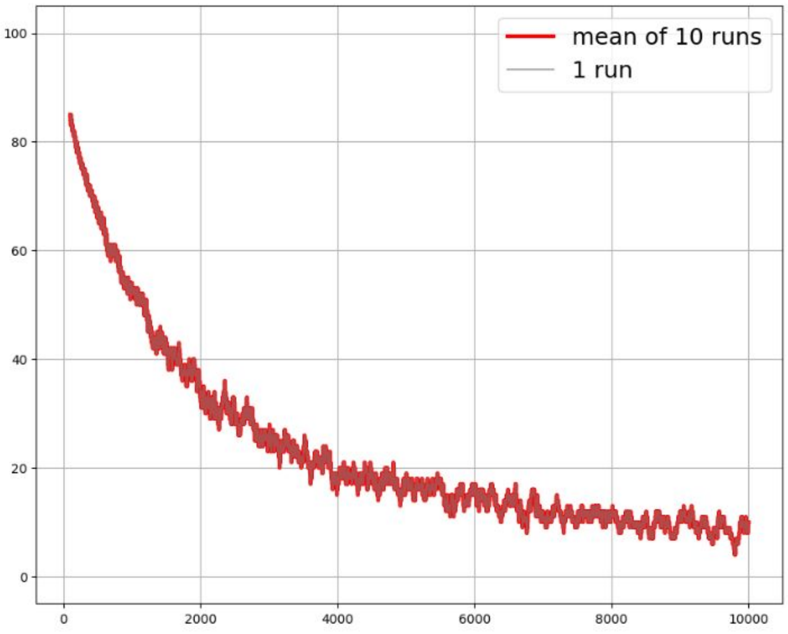


Upper Confidence Bound

Exploration rate through past 100 iterations

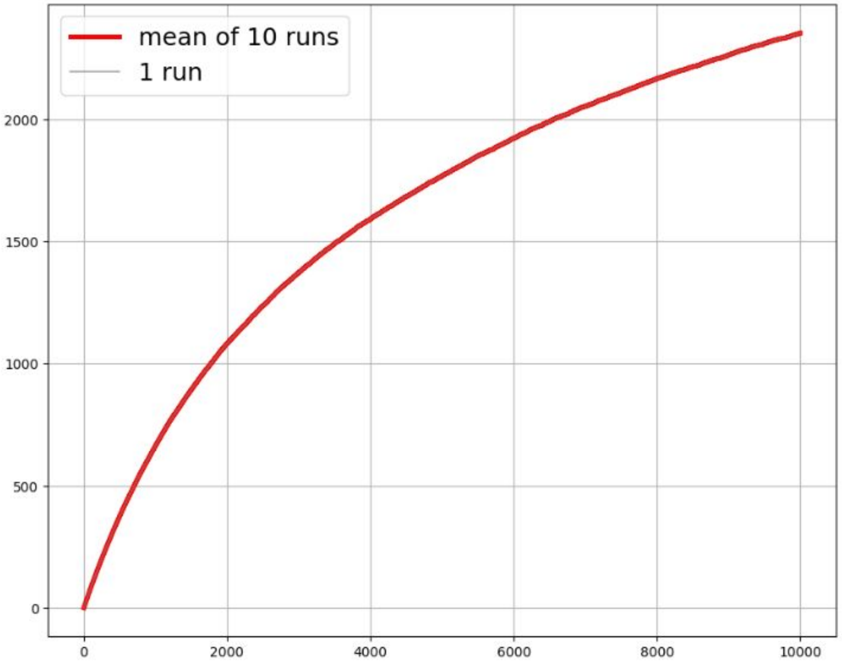


Num of unoptimal arms through past 100 iterations

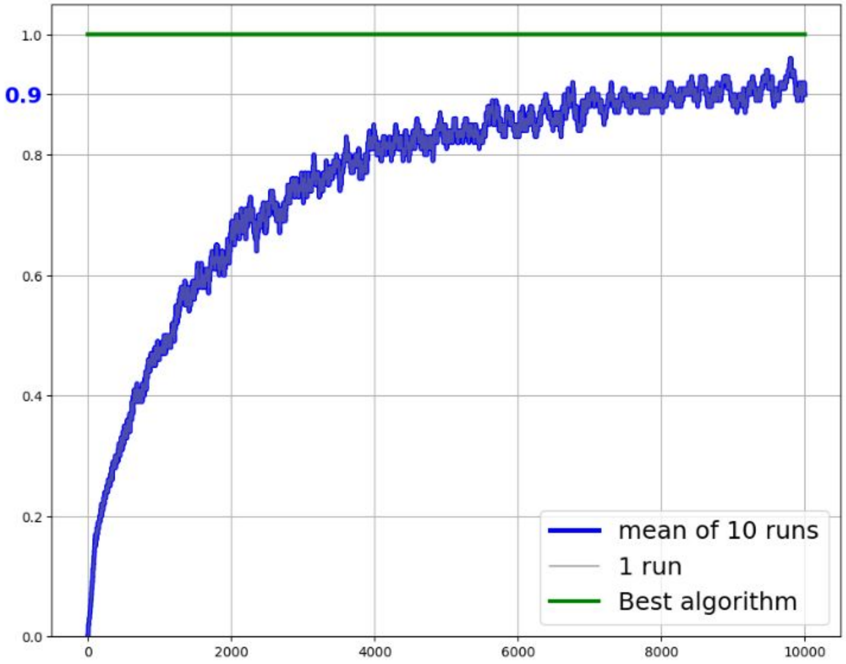


Upper Confidence Bound

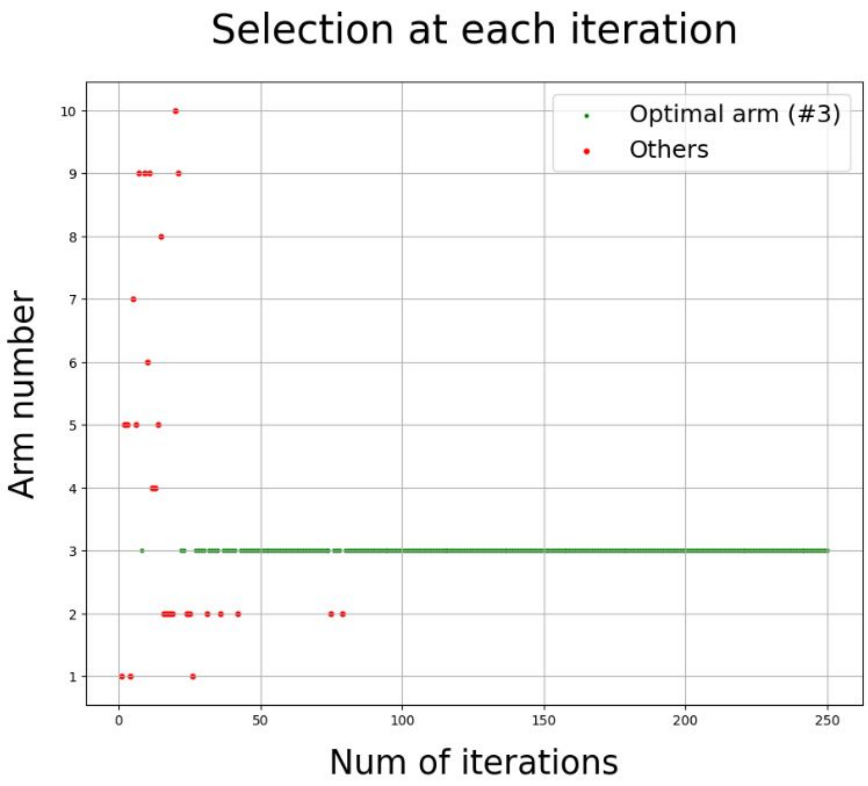
Cumulative regret

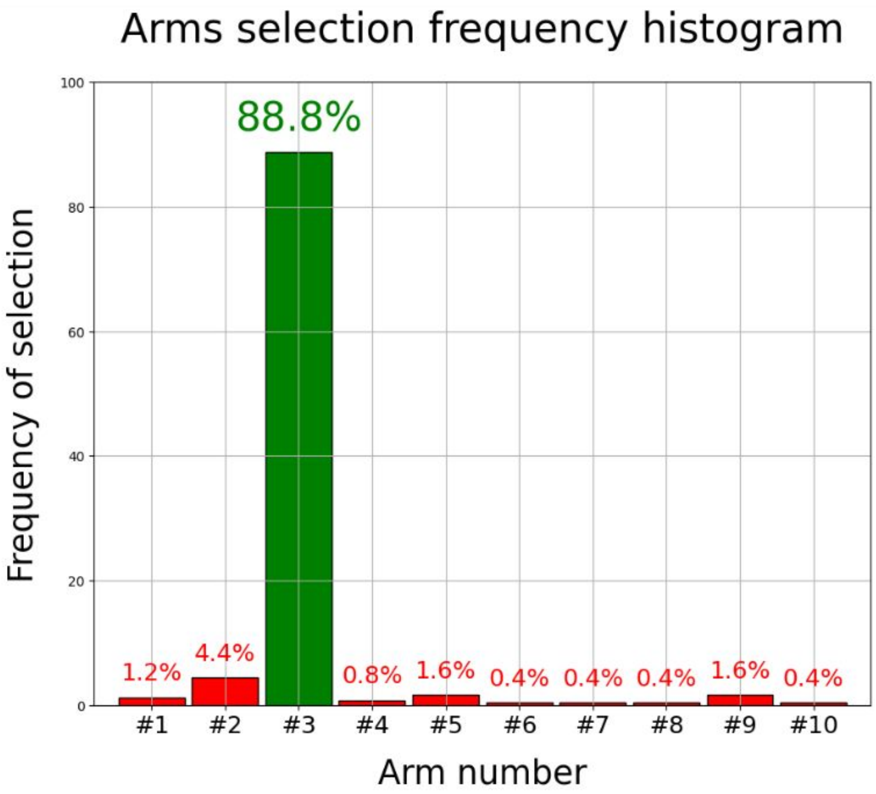


Convergence rate



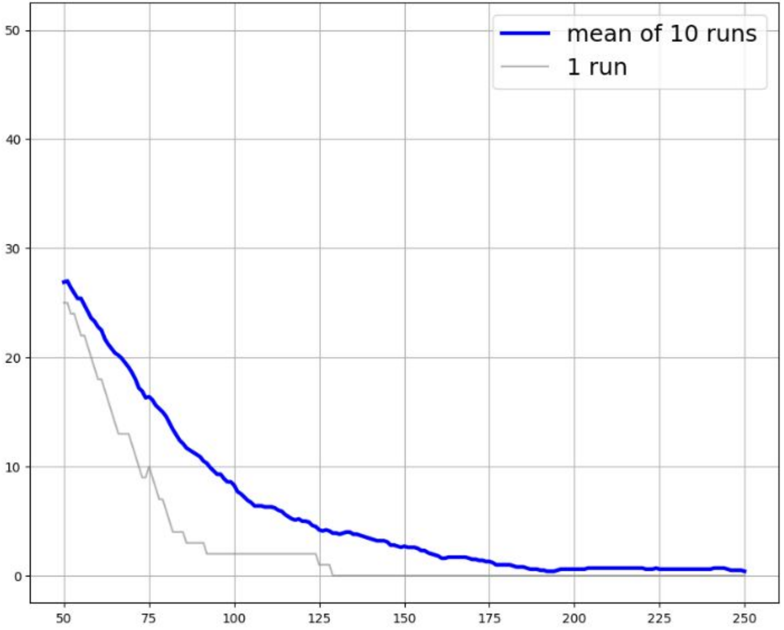
Thompson Sampling



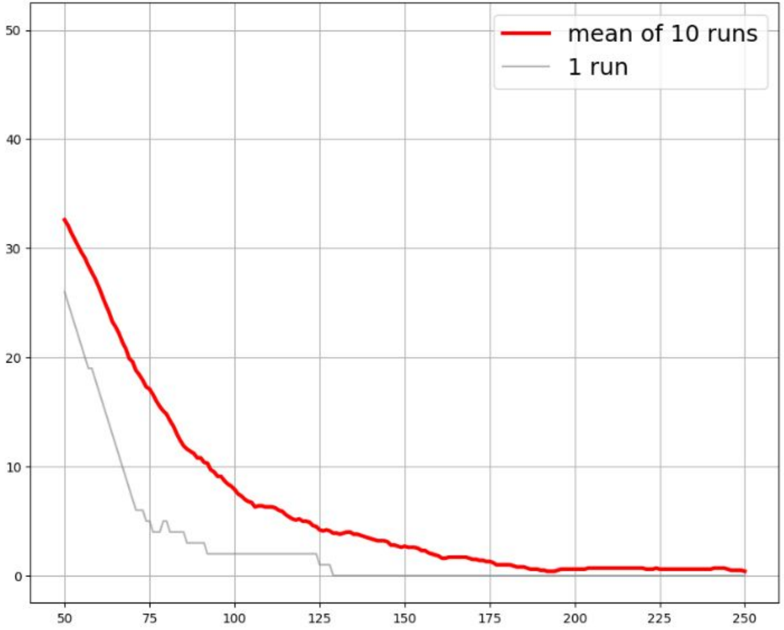


Thompson Sampling

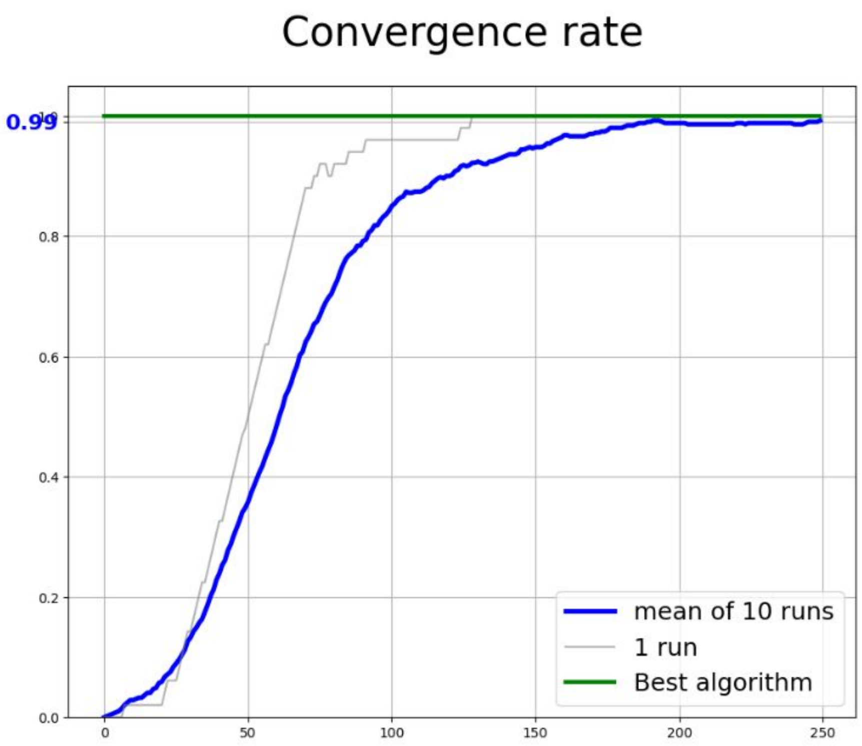
Exploration rate through past 50 iterations



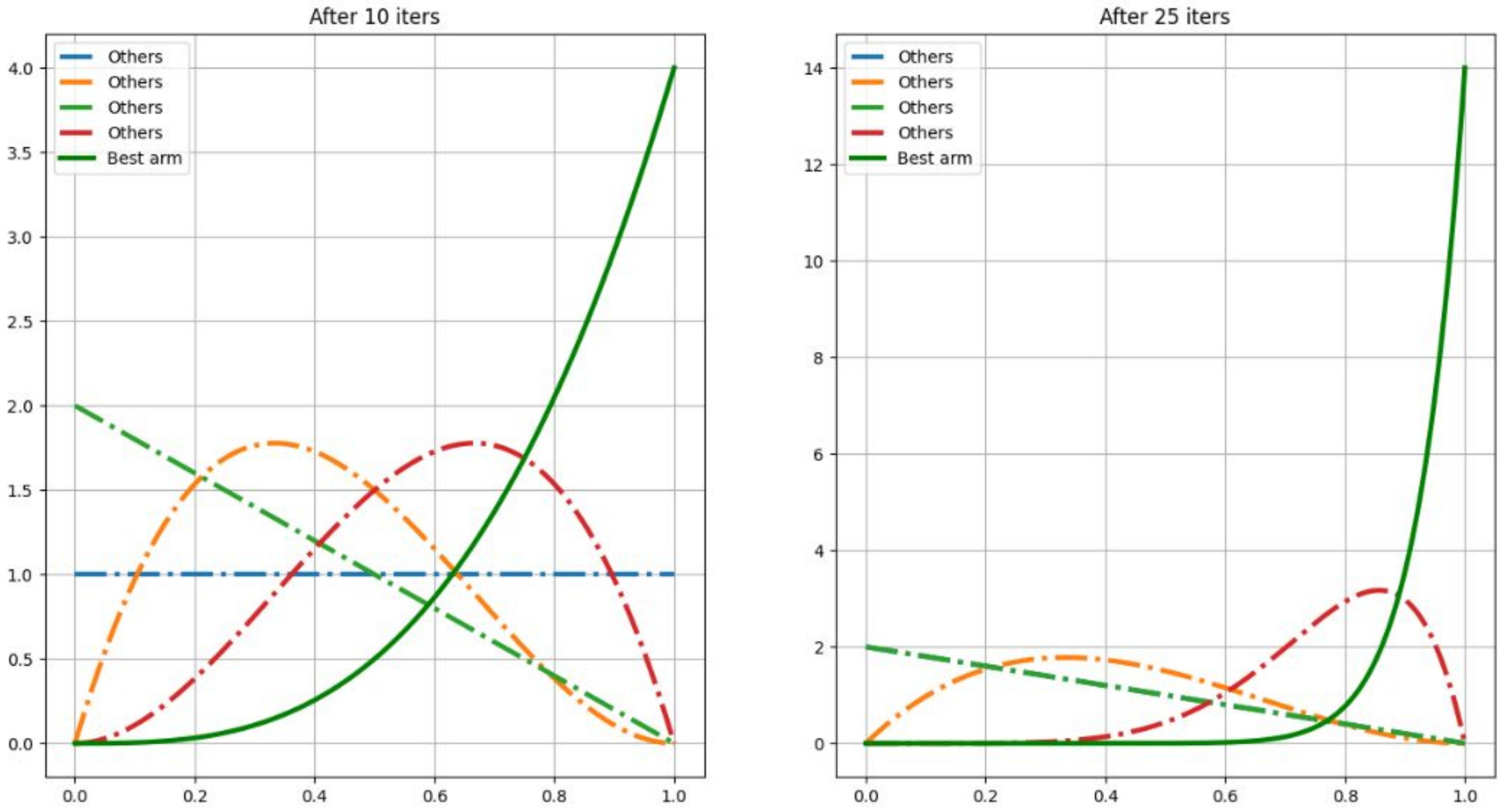
Num of unoptimal arms through past 50 iterations



Thompson Sampling



Thompson Sampling



Модификации TS

Algorithm 1 Online stochastic gradient descent with Thompson Sampling (SGD-TS)

Input: T, K, τ, α.

Randomly choose $a_t \in [K]$ and record X_t, Y_t for $t \in [\tau]$.
Calculate the maximum-likelihood estimator $\hat{\theta}_\tau$ by solving $\sum_{t=1}^{\tau}(Y_t - \mu(X_t^T\theta))X_t = 0$.
Maintain convex set $\mathcal{C} = \{\theta : \|\theta - \hat{\theta}_\tau\| \leq 2\}$.
$\tilde{\theta}_0 \leftarrow \hat{\theta}_\tau$.
for $t = \tau + 1$ **to** T **do**
  if $t\%\tau = 1$ **then**
    $j \leftarrow \lfloor (t-1)/\tau \rfloor$ and $\eta_j = \frac{1}{\alpha j}$.
    Calculate $\nabla l_{j,\tau}$ defined in Equation 3
    Update $\tilde{\theta}_j \leftarrow \prod_{\mathcal{C}} \left(\tilde{\theta}_{j-1} - \eta_j \nabla l_{j,\tau}(\tilde{\theta}_{j-1})\right)$.
    Compute $\bar{\theta}_j = \frac{1}{j}\sum_{q=1}^{j}\tilde{\theta}_q$.
    Compute A_j defined in Equation 5.
    Draw $\theta_j^{\mathrm{TS}} \sim \mathcal{N}\left(\bar{\theta}_j, A_j\right)$.
  end if
  Pull arm $a_t \leftarrow \operatorname{argmax}_{a \in [K]} \mu(x_{t,a}^T \theta_j^{\mathrm{TS}})$ and observe reward Y_t.
end for

Online Stochastic Gradient Descent and Thompson Sampling

Algorithm 1 BootstrapLinTS for partially observable delayed feedback

Input: $n_{\text{prior}}, D_{\max}, T, d, K$.

Data $D_0 = ()$
for $n = 1, \ldots, T$ **do**
  Update data D_n with observed conversions
  for $j = 1, \ldots, n_{\text{prior}}$ **do**
    Sample prior ϑ_j and x_j uniformly over $[0,1]^d$
    Normalise sampled ϑ_j and x_j
    Sample prior reward from Bernoulli$(\vartheta_j \cdot x_j)$
    Sample delays uniformly over $[0, D_{\max}]$
  end for
  Concatenate n_{prior} times and rewards with D_n
  Sample with replacement $n + n_{\text{prior}}$ data points
  Estimate $\hat{S}(t, x)$ and $\hat{p}_1(x)$ via EM
  Observe current contexts x_A, $A = 1, \ldots, K$
  for $A = 1, \ldots, K$ **do**
    Calculate probability $(1 - \hat{S}(T, x_A))\hat{p}_1(x_A)$
  end for
  Select arm $\operatorname{argmax}_i (1 - \hat{S}(T, x_A))\hat{p}_1(x_A)$
end for

Bootstrapped Thompson Sampling

Linear Thompson Sampling (LinTS)

Algorithm 1 Thompson Sampling for Contextual bandits

Set $B = I_d, \hat{\mu} = 0_d,\ f = 0_d$.

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

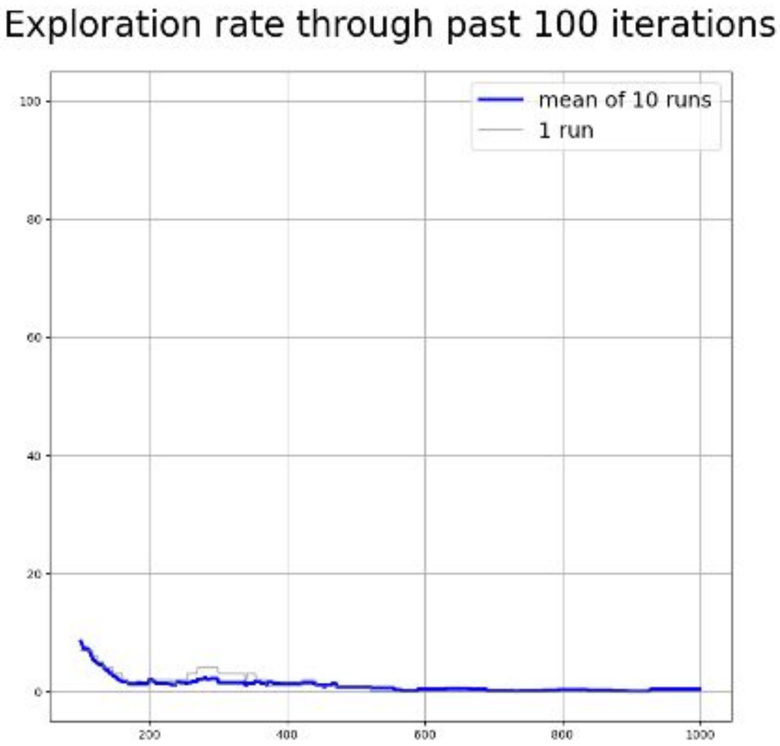
Sample $\tilde{\mu}(t)$ from distribution $\mathcal{N}(\hat{\mu}, v^2 B^{-1})$.

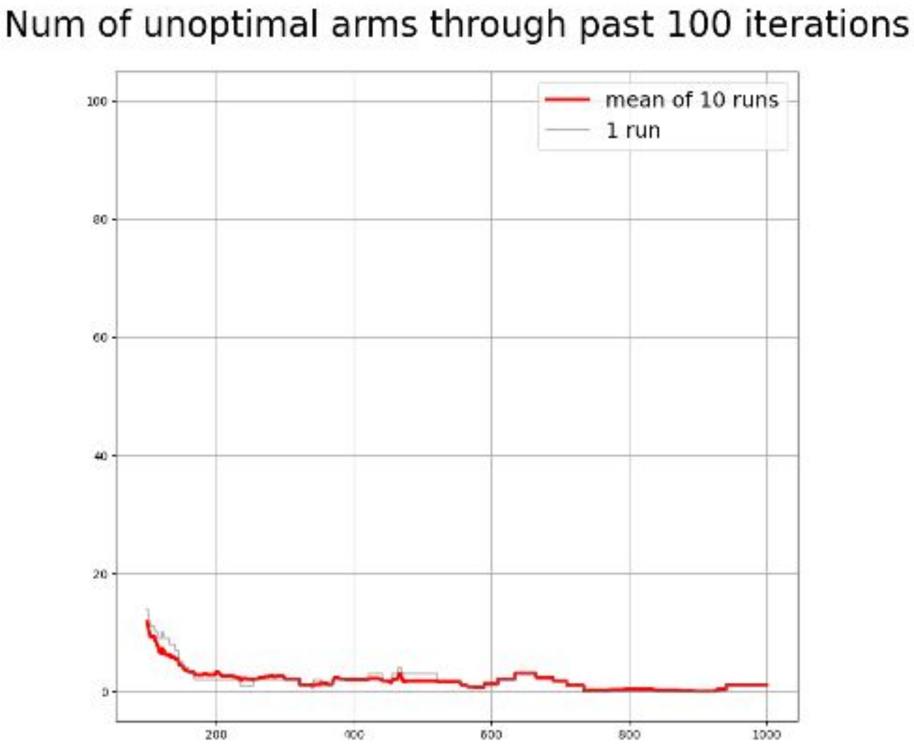
Play arm $a(t) := \arg\max_i b_i(t)^T \tilde{\mu}(t)$, and observe reward r_t.

Update $B = B + b_{a(t)}(t) b_{a(t)}(t)^T,\ f = f + b_{a(t)}(t) r_t,\ \hat{\mu} = B^{-1} f$.

end for

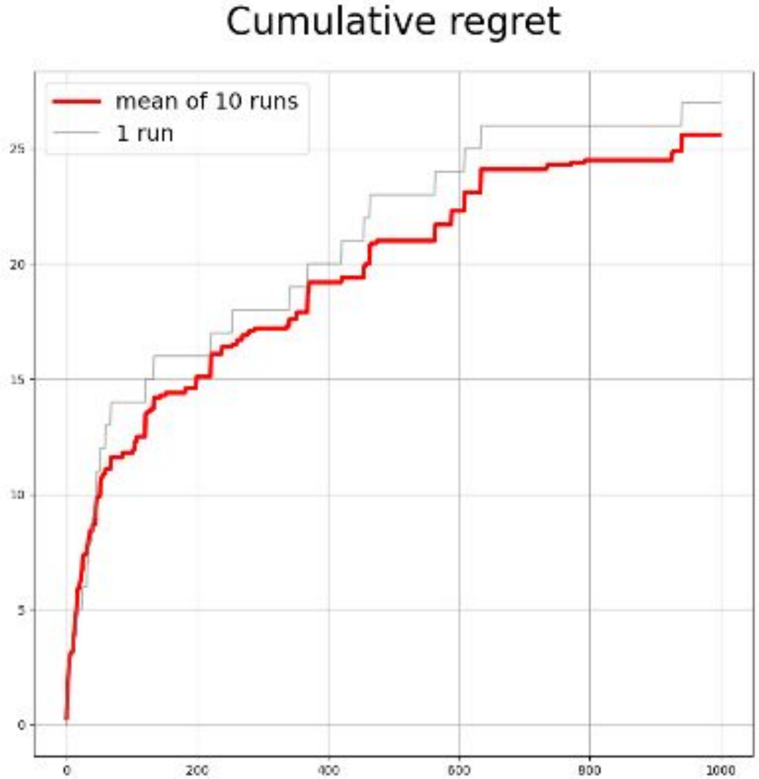
LinTS

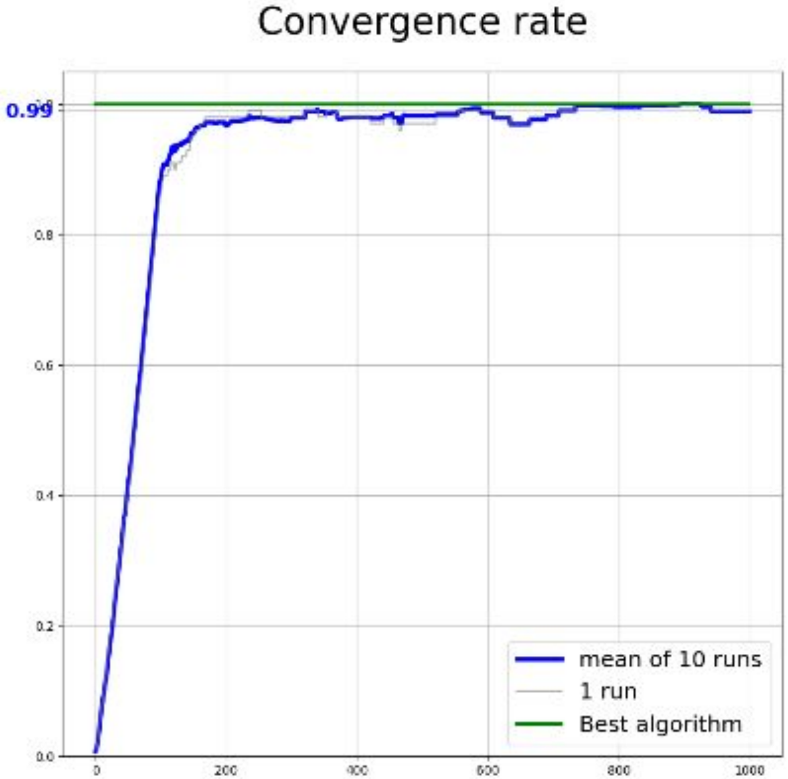




Mushroom dataset

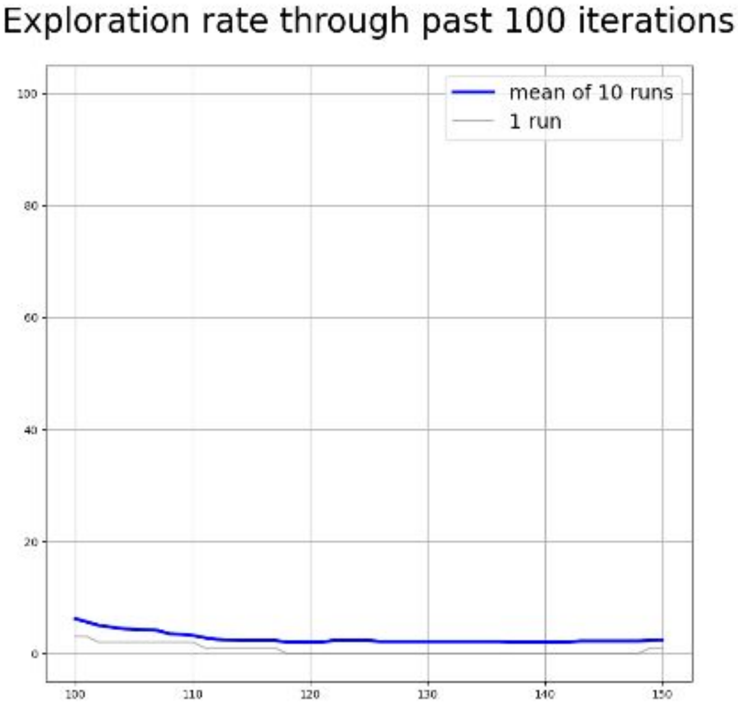
LinTS

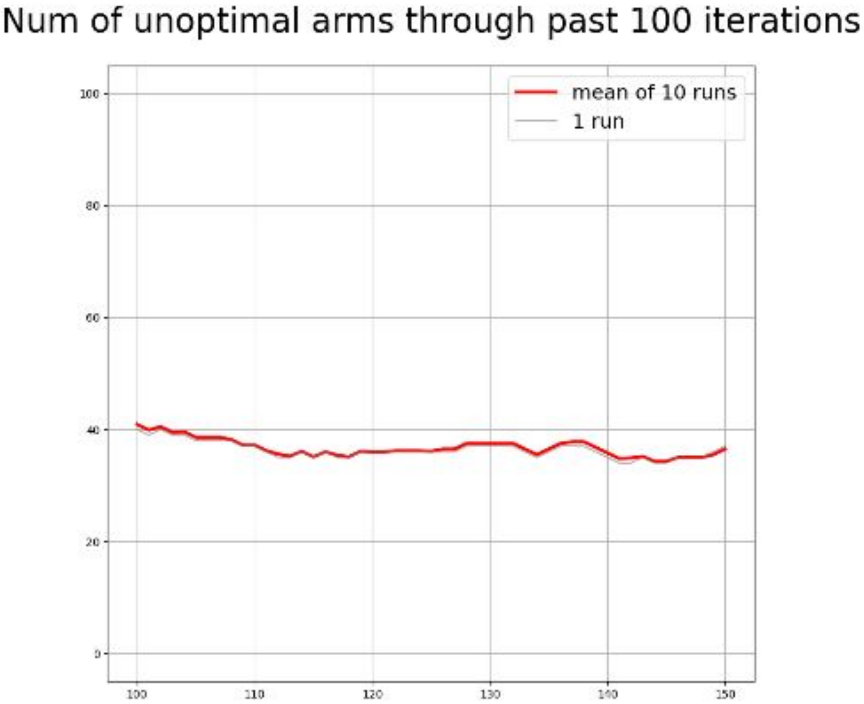




Mushroom dataset

LinTS

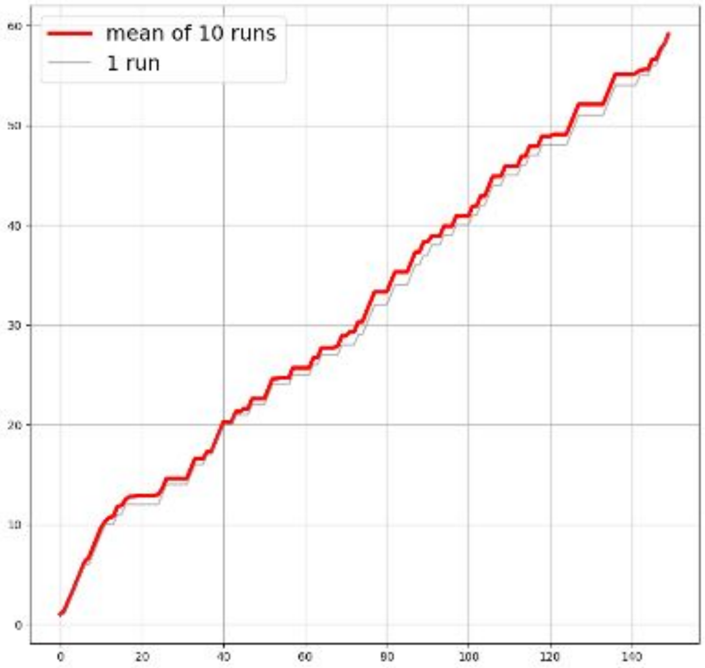




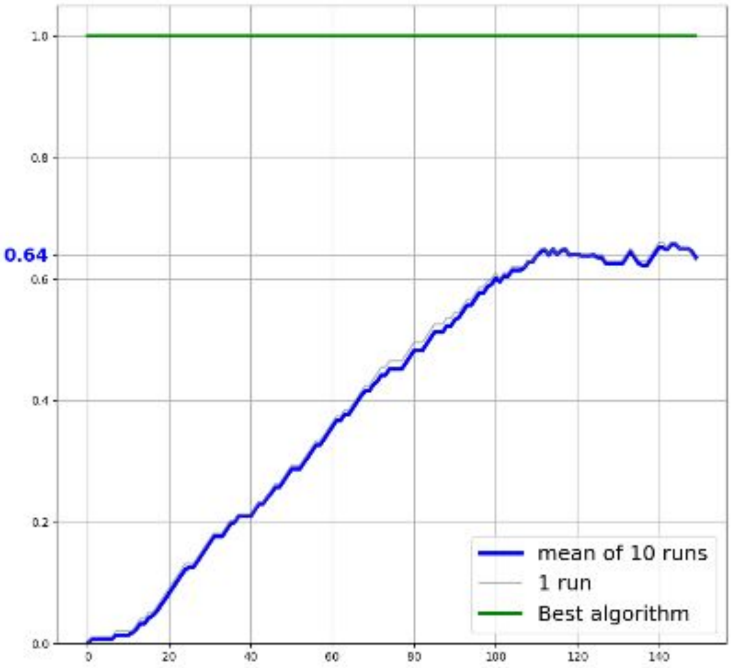
Iris dataset

LinTS

Cumulative regret

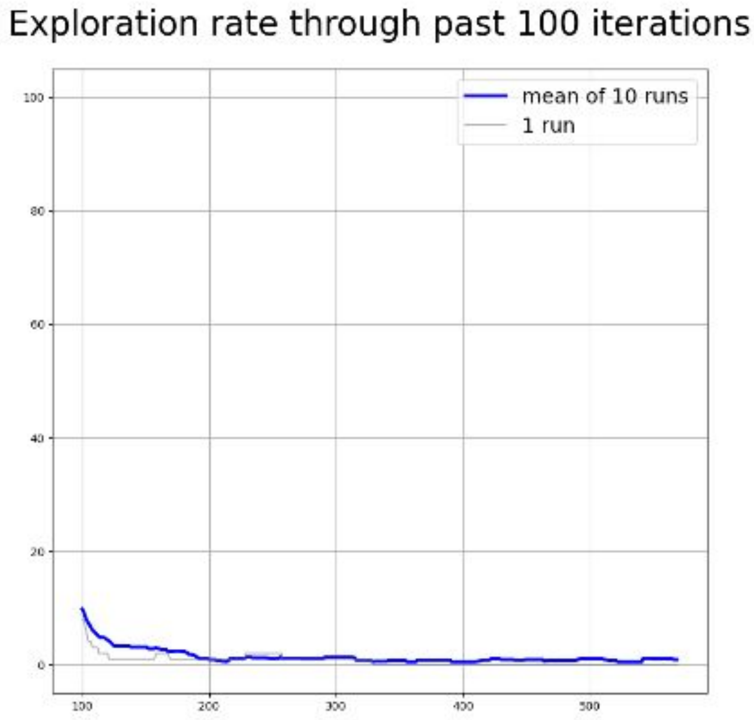


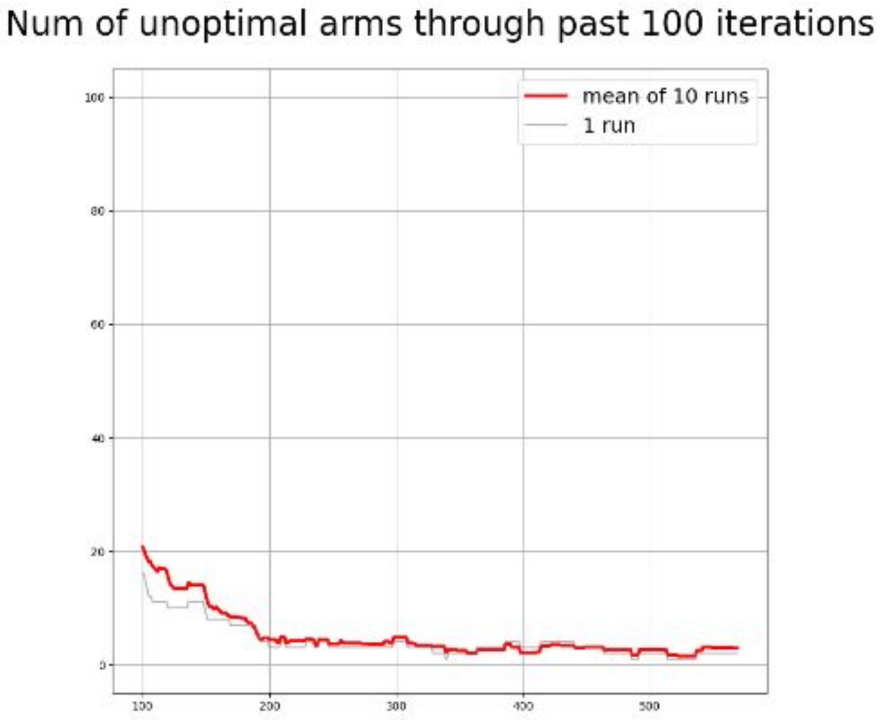
Convergence rate



Iris dataset

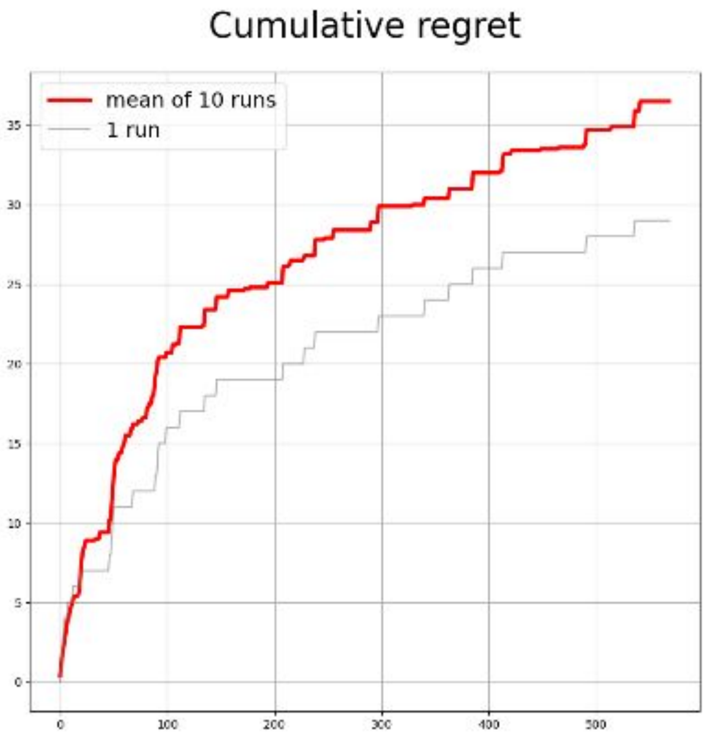
LinTS

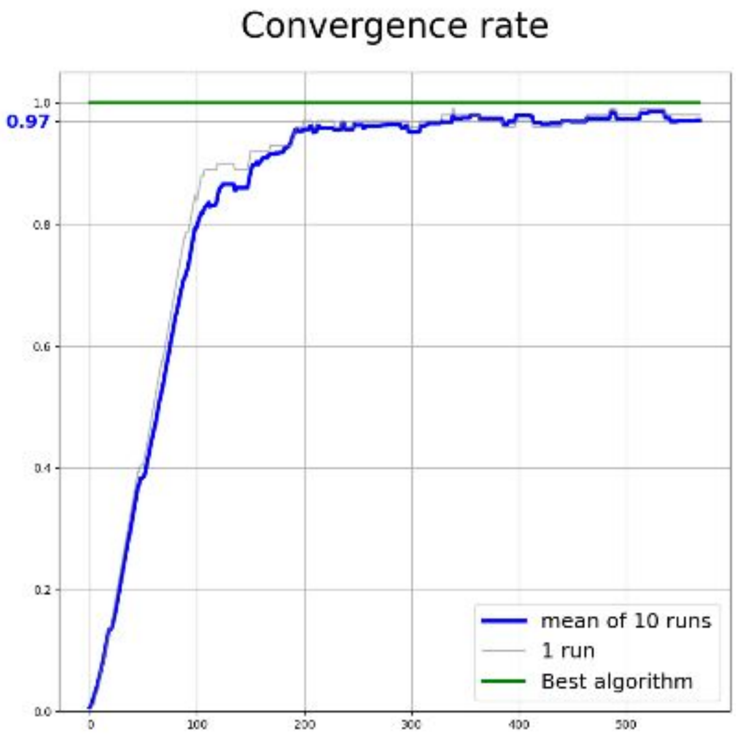




Breast cancer dataset

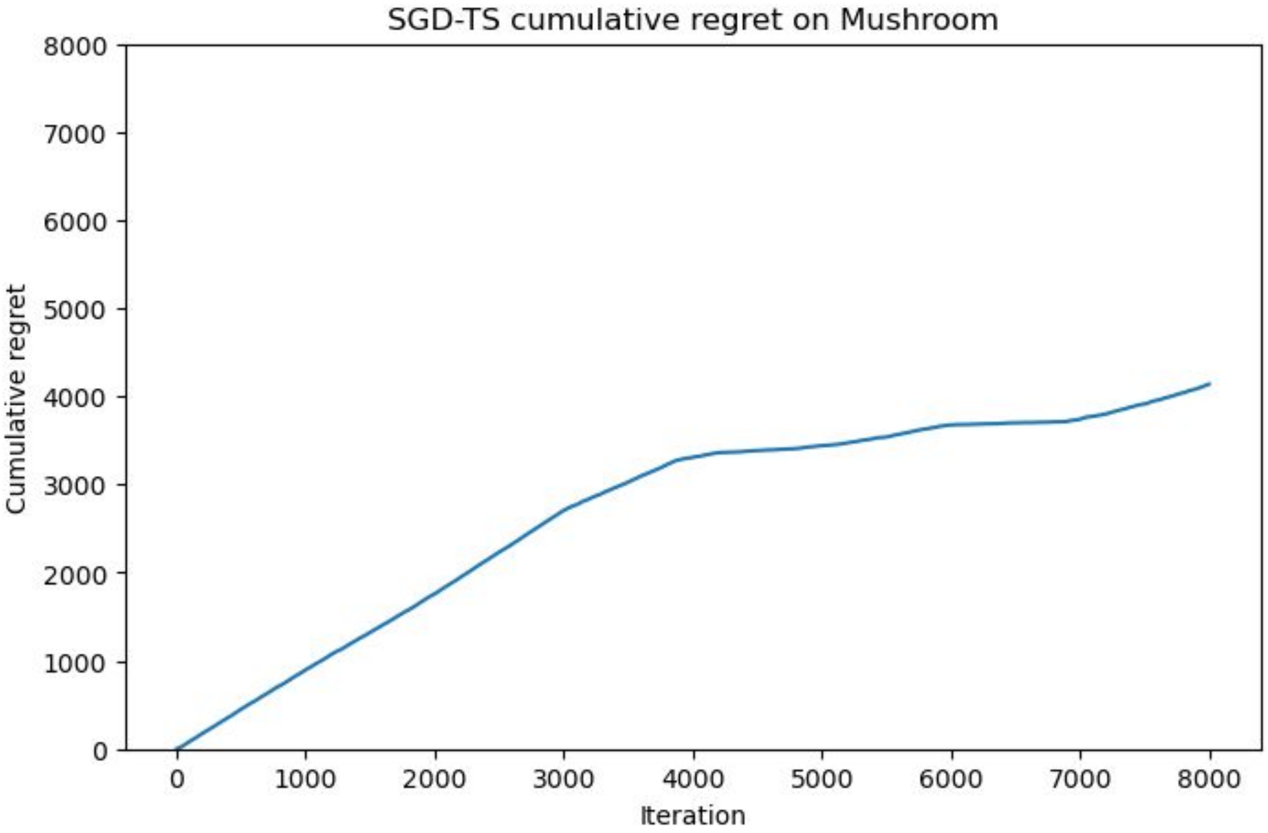
LinTS





Breast cancer dataset

SGD-TS


SGD-TS cumulative regret on Mushroom
Cumulative regret
Iteration
0
1000
2000
3000
4000
5000
6000
7000
8000

GTS

на всём датасете GTS сходится, но медленно (Mushroom dataset)