

# Оптимизация экономического портфеля с использованием методов машинного обучения

Бакалаврский диплом

Михаил Давыдов

ФПМИ МФТИ, кафедра дискретной математики

9 июня 2024 г.

# Оглавление

# Проблема

# Пreamble

Инвесторы вкладывают свои деньги в акции, и хотят использовать для распределения денег эффективные алгоритмы. Для этого они могут использовать модели для приближенного вычисления изменения стоимости акций.

# Пreamble

Инвесторы вкладывают свои деньги в акции, и хотят использовать для распределения денег эффективные алгоритмы. Для этого они могут использовать модели для приближенного вычисления изменения стоимости акций. Будем считать, что  $i$ -ая акция за определенный фиксированный промежуток времени увеличивается в своей стоимости на значение случайной величины  $\xi_i$ , причем  $\forall i, j, i \neq j \rightarrow \xi_i \perp \xi_j$ .

# Пreamble

Инвесторы вкладывают свои деньги в акции, и хотят использовать для распределения денег эффективные алгоритмы. Для этого они могут использовать модели для приближенного вычисления изменения стоимости акций. Будем считать, что  $i$ -ая акция за определенный фиксированный промежуток времени увеличивается в своей стоимости на значение случайной величины  $\xi_i$ , причем  $\forall i, j, i \neq j \rightarrow \xi_i \perp \xi_j$ . Тогда необходимо найти оптимальный вектор вероятностей, обозначающий распределение денег по активам, максимизирующий среднюю субъективную прибыль.

# Классическая задача о многоруких бандитах

Если инвестор изначально не обладает информацией о стоимости активов, то в качестве модели можно использовать модель многоруких бандитов:

# Классическая задача о многоруких бандитах

Если инвестор изначально не обладает информацией о стоимости активов, то в качестве иодели можно использовать модель многоруких бандитов:

- Есть  $n$  рычагов,  $i$ -ый рычаг соответствует какому-то распределению со средним  $t_i$ . Изначально распределения, как и средние, неизвестны.
- При нажатии на  $i$ -ый рычаг выдается награда в соответствии с  $i$ -ым распределением.

# Классическая задача о многоруких бандитах

Если инвестор изначально не обладает информацией о стоимости активов, то в качестве иодели можно использовать модель многоруких бандитов:

- Есть  $n$  рычагов,  $i$ -ый рычаг соответствует какому-то распределению со средним  $m_i$ . Изначально распределения, как и средние, неизвестны.
- При нажатии на  $i$ -ый рычаг выдается награда в соответствии с  $i$ -ым распределением.
- Задача – найти  $\arg \max_{\mathbf{p} \in \Delta^n} \mathbf{p}^T \cdot \mathbf{m} = \sum_{i=1}^n p_i m_i$ , где  $\Delta^n = \{\mathbf{p} = (p_1, \dots, p_n) : (\sum_{i=1}^n p_i = 1) \wedge (\forall i p_i \geq 0)\}$ ,  $\mathbf{p}$  отвечает за долю от общих средств, вкладываемых в каждый актив на каждом шаге.

# Классическая задача о многоруких бандитах

Если инвестор изначально не обладает информацией о стоимости активов, то в качестве иодели можно использовать модель многоруких бандитов:

- Есть  $n$  рычагов,  $i$ -ый рычаг соответствует какому-то распределению со средним  $m_i$ . Изначально распределения, как и средние, неизвестны.
- При нажатии на  $i$ -ый рычаг выдается награда в соответствии с  $i$ -ым распределением.
- Задача – найти  $\arg \max_{\mathbf{p} \in \Delta^n} \mathbf{p}^T \cdot \mathbf{m} = \sum_{i=1}^n p_i m_i$ , где  $\Delta^n = \{\mathbf{p} = (p_1, \dots, p_n) : (\sum_{i=1}^n p_i = 1) \wedge (\forall i p_i \geq 0)\}$ ,  $\mathbf{p}$  отвечает за долю от общих средств, вкладываемых в каждый актив на каждом шаге.
- Равносильно нахождению рычага с наибольшим средним

# Проблемы

Инвесторов часто интересует не только максимизация прибыли, но и минимизация рисков. Прошлая модель этого не учитывала.

# Проблемы

Инвесторов часто интересует не только максимизация прибыли, но и минимизация рисков. Прошлая модель этого не учитывала.

Кроме того, часто для простоты рычагам дают нормальные распределения, что плохо отражает реальность, где чаще представлены распределения со степенными хвостами.

# Измененная задача о многоруких бандитах

- Аналогично, есть  $n$  рычагов, каждый соответствует распределению со средним  $m_i$  и дисперсией  $\sigma_i^2$ .  
Аналогично, при нажатии на  $i$ -ый рычаг выдается награда из  $i$ -го распределения.

# Измененная задача о многоруких бандитах

- Аналогично, есть  $n$  рычагов, каждый соответствует распределению со средним  $m_i$  и дисперсией  $\sigma_i^2$ .  
Аналогично, при нажатии на  $i$ -ый рычаг выдается награда из  $i$ -го распределения.
- Задача – найти

$$\arg \max_{\mathbf{p} \in \Delta^n} \left( \mathbf{p}^T \cdot \mathbf{m} - \lambda (\mathbf{p}^T)^2 \cdot \boldsymbol{\sigma}^2 \right) = \sum_{i=1}^n p_i m_i - \lambda \sum_{i=1}^n p_i^2 \sigma_i^2$$

где  $\lambda > 0$  – коэффициент отвращения, или неприятия к риску.

# Измененная задача о многоруких бандитах

- Аналогично, есть  $n$  рычагов, каждый соответствует распределению со средним  $m_i$  и дисперсией  $\sigma_i^2$ . Аналогично, при нажатии на  $i$ -ый рычаг выдается награда из  $i$ -го распределения.
- Задача – найти

$$\arg \max_{\mathbf{p} \in \Delta^n} \left( \mathbf{p}^T \cdot \mathbf{m} - \lambda (\mathbf{p}^T)^2 \cdot \boldsymbol{\sigma}^2 \right) = \sum_{i=1}^n p_i m_i - \lambda \sum_{i=1}^n p_i^2 \sigma_i^2$$

где  $\lambda > 0$  – коэффициент отвращения, или неприятия к риску.

- В этой трактовке задачи вектор вероятностей может не сосредотачиваться в одном рычаге.

# Обзор литературы

# Обзор литературы

- В книге Саттона и Барто “Reinforcement Learning: An Introduction” [3] вторая глава посвящена различным техникам для нахождения оптимального решения в задаче о многоруких бандитах. Однако их подходы не учитывали риск и степень отвращения к нему, а также проверялись только на нормальных распределениях.

# Обзор литературы

- В книге Саттона и Барто “Reinforcement Learning: An Introduction” [3] вторая глава посвящена различным техникам для нахождения оптимального решения в задаче о многоруких бандитах. Однако их подходы не учитывали риск и степень отвращения к нему, а также проверялись только на нормальных распределениях.
- Классическое решение из портфельной теории Марковица (его можно найти, например, в книге [1]) находит

$$\arg \max_{\mathbf{p} \in R^n : \mathbf{p}^T \cdot \mathbf{1} = 1} \left( \mathbf{p}^T \cdot \mathbf{m} - \lambda (\mathbf{p}^T)^2 \cdot \sigma^2 \right)$$
 при условии, что средние

и дисперсии известны, а вероятности могут быть любыми вещественными числами с единичной суммой. В случае, когда одно из условий не выполнено, это решение не работает.

# Задачи

# Задачи

- 1 Проанализировать известные подходы в классической задаче о многоруких бандитах на предмет применимости для распределений, отличных от нормального.

# Задачи

- 1 Проанализировать известные подходы в классической задаче о многоруких бандитах на предмет применимости для распределений, отличных от нормального.
- 2 Придумать алгоритмы и подходы для решения задачи о многоруких бандитах с учетом степени отвращения к риску.

# Задачи

- 1 Проанализировать известные подходы в классической задаче о многоруких бандитах на предмет применимости для распределений, отличных от нормального.
- 2 Придумать алгоритмы и подходы для решения задачи о многоруких бандитах с учетом степени отвращения к риску.
- 3 Протестировать созданные подходы на различных распределениях.

# Начальные эксперименты

# Подсчет параметров

- $R_t$  – награда, полученная на  $t$ -ом шагу (то есть нажали на рычаг в  $t$ -ый раз).
- $N_t(a) := \sum_{i=1}^{t-1} I(A_i = a)$  – количество нажатий на рычаг  $a$  на  $t$ -ом шагу.

# Подсчет параметров

- $R_t$  – награда, полученная на  $t$ -ом шагу (то есть нажали на рычаг в  $t$ -ый раз).
- $A_t$  – номер рычага, выбранный на  $t$ -ом шагу.
- $N_t(a) := \sum_{i=1}^{t-1} I(A_i = a)$  – количество нажатий на рычаг  $a$  на  $t$ -ом шагу.

# Подсчет параметров

- $R_t$  – награда, полученная на  $t$ -ом шагу (то есть нажали на рычаг в  $t$ -ый раз).
- $A_t$  – номер рычага, выбранный на  $t$ -ом шагу.
- $N_t(a) := \sum_{i=1}^{t-1} I(A_i = a)$  – количество нажатий на рычаг  $a$  на  $t$ -ом шагу.
- $Q_t(a) := \frac{\sum_{i=1}^{t-1} R_i \cdot I(A_i = a)}{N_t(a)}$  – средняя награда при нажатии рычага с номером  $a$ . Более короткая формула для выбранного действия  $a$ :  $Q_t(a) = Q_t(a) + \frac{1}{N_t(a)+1}(R_t - Q_t(a))$ .

# Подсчет параметров

- $R_t$  – награда, полученная на  $t$ -ом шагу (то есть нажали на рычаг в  $t$ -ый раз).
- $A_t$  – номер рычага, выбранный на  $t$ -ом шагу.
- $N_t(a) := \sum_{i=1}^{t-1} I(A_i = a)$  – количество нажатий на рычаг  $a$  на  $t$ -ом шагу.
- $Q_t(a) := \frac{\sum_{i=1}^{t-1} R_i \cdot I(A_i = a)}{N_t(a)}$  – средняя награда при нажатии рычага с номером  $a$ . Более короткая формула для выбранного действия  $a$ :  $Q_t(a) = Q_t(a) + \frac{1}{N_t(a)+1}(R_t - Q_t(a))$ .
- $\bar{R}_t := \frac{\sum_{i=1}^{t-1} R_i}{\max(t-1, 1)}$  – средняя награда за все предыдущие шаги, или, как ее называют по-другому, *baseline*.

# Параметры

- Кол-во рычагов: 10

# Параметры

- Кол-во рычагов: 10
- Распределения рычагов (все рычаги брались из одного семейства распределений):
  - Стандартное нормальное ( $N(0, 1)$ или  $t_\infty$ )
  - Распределение Стьюдента с дисперсией 1 и 3-мя степенями свободы  $t_3$  (для единичной дисперсии распределение было домножено на  $\sqrt{\frac{1}{3}}$ )
  - Распределение Стьюдента с 2-мя степенями свободы  $t_2$
  - Распределение Коши  $t_1$

# Параметры

- Кол-во рычагов: 10
- Распределения рычагов (все рычаги брались из одного семейства распределений):
  - Стандартное нормальное ( $N(0, 1)$ или  $t_\infty$ )
  - Распределение Стьюдента с дисперсией 1 и 3-мя степенями свободы  $t_3$  (для единичной дисперсии распределение было домножено на  $\sqrt{\frac{1}{3}}$ )
  - Распределение Стьюдента с 2-мя степенями свободы  $t_2$
  - Распределение Коши  $t_1$

Медианы рычагов брались из нормального распределения  $N(1, 1)$

# Параметры

- Количество тестов – 2000

# Параметры

- Количество тестов – 2000
- Длина каждого теста – 1000 шагов

# Параметры

- Количество тестов – 2000
- Длина каждого теста – 1000 шагов
- Метрики:
  - 1 Средняя награда за шаг
  - 2 Процент оптимальных действий (нажатий на рычаг с максимальным матожиданием или медианой для распря Коши)

# Алгоритмы

## ① Greedy и $\epsilon$ -greedy

$$A_t = \begin{cases} \arg \max_a Q_t(a), & \text{with probability } 1 - \epsilon, \\ \text{a random action,} & \text{with probability } \epsilon. \end{cases}$$

# Алгоритмы

## 1 Greedy и $\epsilon$ -greedy

$$A_t = \begin{cases} \arg \max_a Q_t(a), & \text{with probability } 1 - \epsilon, \\ \text{a random action,} & \text{with probability } \epsilon. \end{cases}$$

## 2 Стратегия с позитивной инициализацией

$$(\forall a Q_t(a) = d, d > 0)$$

# Алгоритмы

## ① Greedy и $\epsilon$ -greedy

$$A_t = \begin{cases} \arg \max_a Q_t(a), & \text{with probability } 1 - \epsilon, \\ \text{a random action,} & \text{with probability } \epsilon. \end{cases}$$

## ② Стратегия с позитивной инициализацией

$$(\forall a Q_t(a) = d, d > 0)$$

## ③ Upper-Confidence Bound selection

$$A_t = \arg \max_a \left[ Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right], \quad c > 0$$

# Алгоритмы

## 1 Greedy и $\epsilon$ -greedy

$$A_t = \begin{cases} \arg \max_a Q_t(a), & \text{with probability } 1 - \epsilon, \\ \text{a random action,} & \text{with probability } \epsilon. \end{cases}$$

## 2 Стратегия с позитивной инициализацией

$$(\forall a Q_t(a) = d, d > 0)$$

## 3 Upper-Confidence Bound selection

$$A_t = \arg \max_a \left[ Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right], \quad c > 0$$

## 4 Gradient bandit

# Алгоритмы

## 1 Greedy и $\epsilon$ -greedy

$$A_t = \begin{cases} \arg \max_a Q_t(a), & \text{with probability } 1 - \epsilon, \\ \text{a random action,} & \text{with probability } \epsilon. \end{cases}$$

## 2 Стратегия с позитивной инициализацией

$(\forall a Q_t(a) = d, d > 0)$

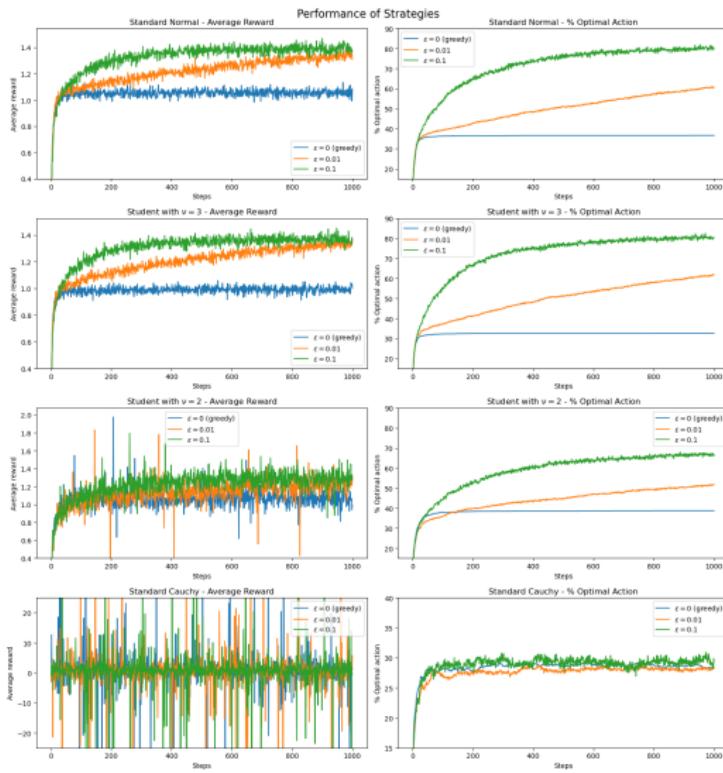
## 3 Upper-Confidence Bound selection

$$A_t = \arg \max_a \left[ Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right], \quad c > 0$$

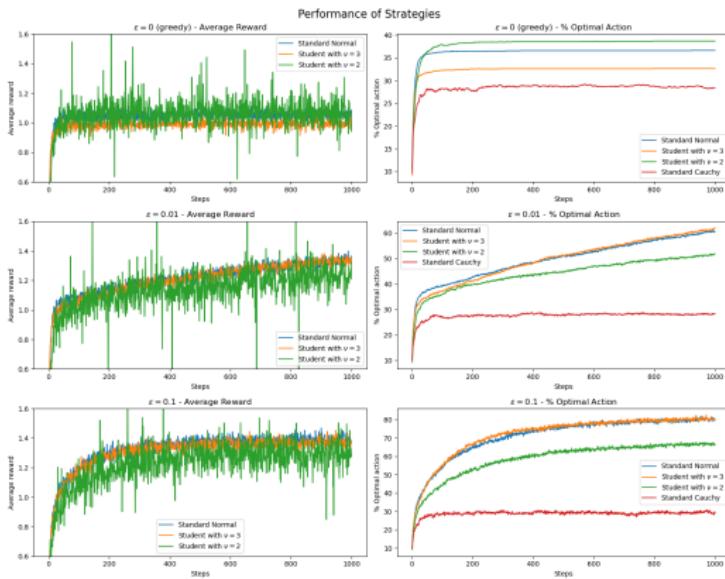
## 4 Gradient bandit

В конце сравнил все алгоритмы в зависимости от их гиперпараметров

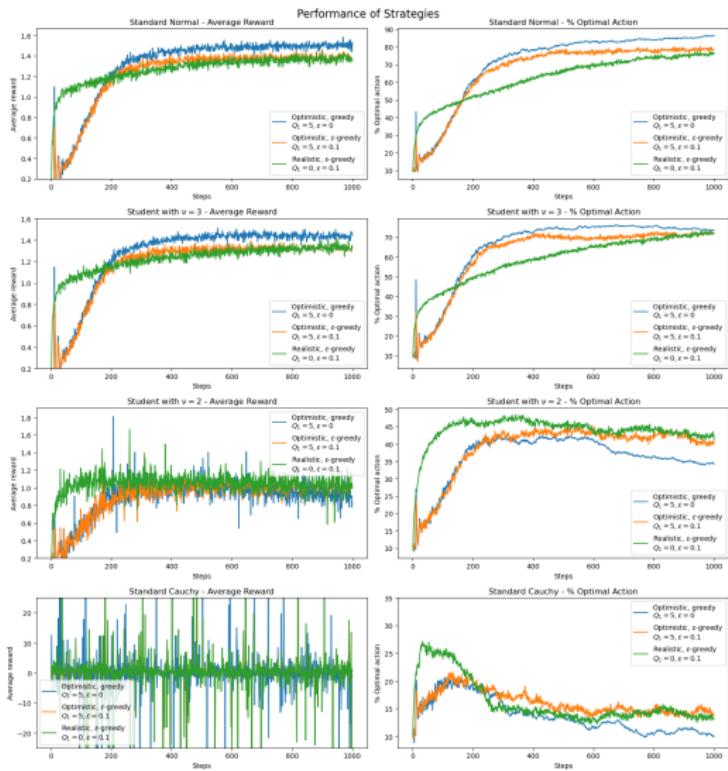
# Результаты – $\epsilon$ -greedy



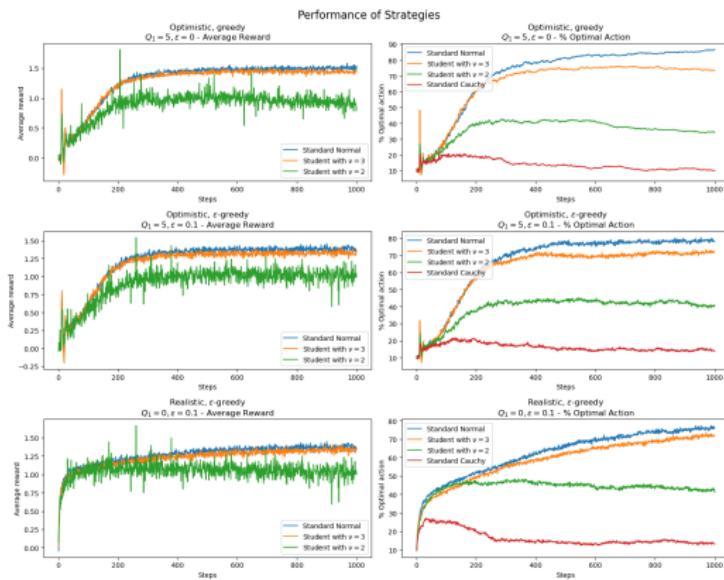
# Результаты – $\epsilon$ -greedy



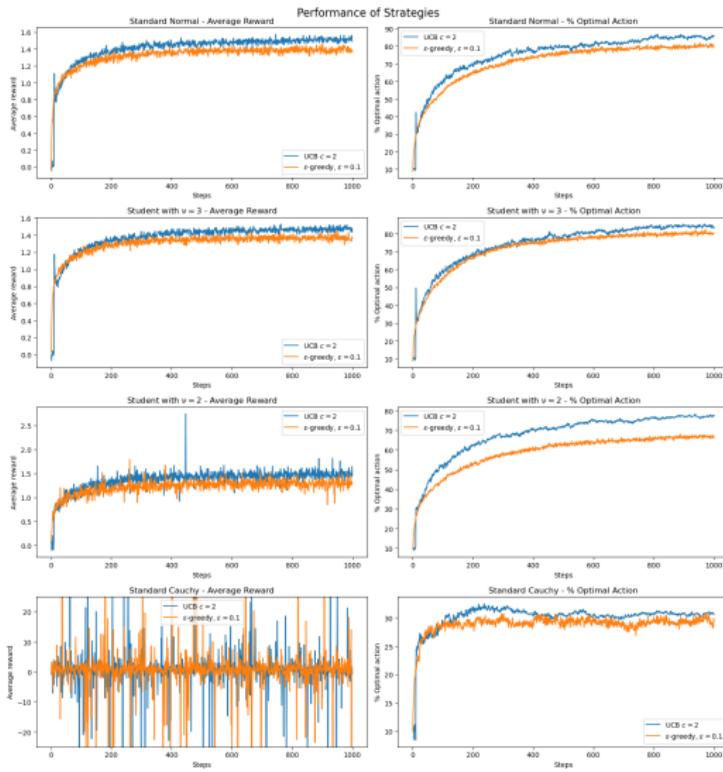
# Результаты – optimistic initialization



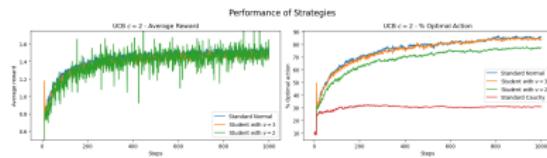
# Результаты – optimistic initialization



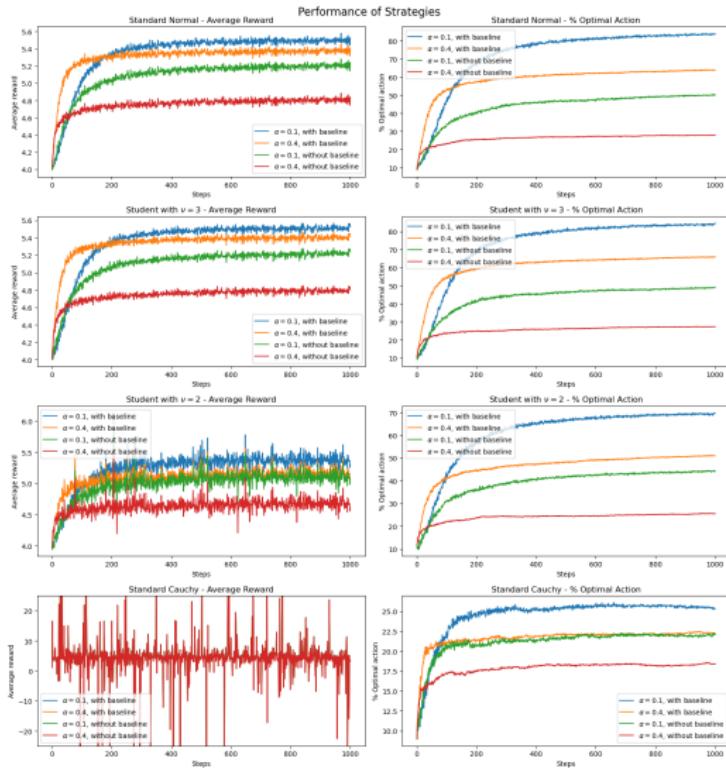
# Результаты – UCB



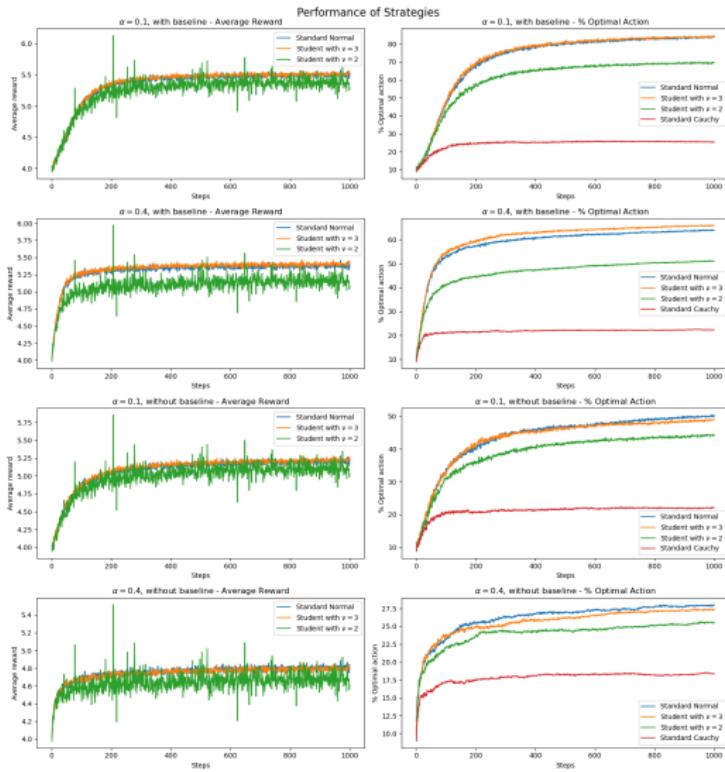
# Результаты – UCB



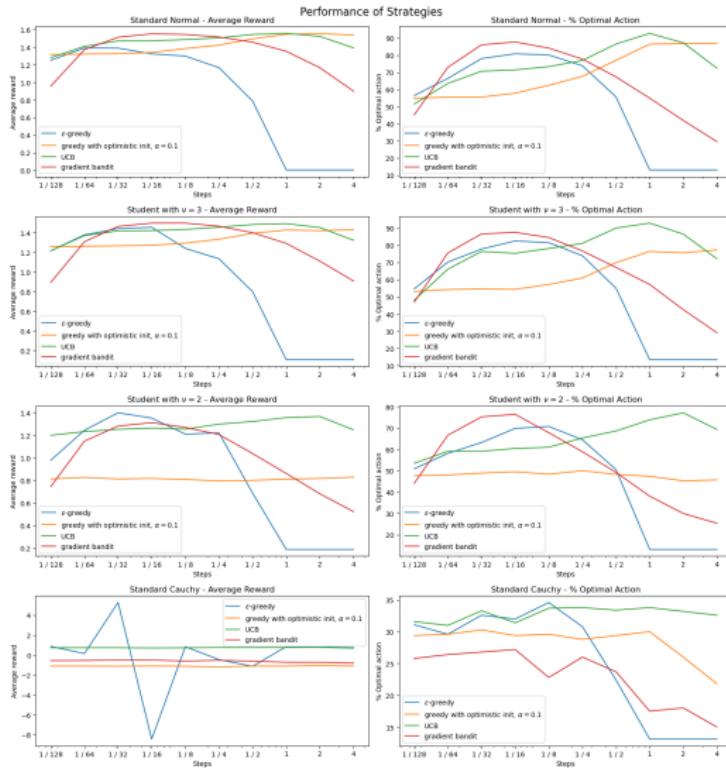
# Результаты – gradient bandits



# Результаты – gradient bandits



# Результаты – gradient bandits



# Выводы

Проделанные эксперименты позволяют судить о том, что Gradient bandits,  $\epsilon$ -greedy и UCB – стратегии показывают высокую эффективность на степенных распределениях. Так как UCB – единственная из стратегий, показывающая высокую эффективность на всех метриках и всех распределениях, то эта стратегия – лучший из кандидатов для применения в оптимизации портфолио в модели прироста стоимости акций как многоруких бандитов.

# Теория

# Неудачные алгоритмы

слайд 1

В каждой из стратегий найдена формула для вычисления оптимальных действий и получен алгоритм, реализующий стратегию.

- 1 Каждый ход увеличиваем одну из вероятностей  $p_i$  на  $\Delta p \geq 0$ , а другую вероятность  $p_j$  – уменьшать на  $\Delta p$  так, чтобы  $V = \mathbf{p}^T \cdot \mathbf{m} - \lambda(\mathbf{p}^2)^T \sigma^2$  увеличилось на максимально возможное значение.

# Неудачные алгоритмы

## слайд 1

В каждой из стратегий найдена формула для вычисления оптимальных действий и получен алгоритм, реализующий стратегию.

- ① Каждый ход увеличиваем одну из вероятностей  $p_i$  на  $\Delta p \geq 0$ , а другую вероятность  $p_j$  – уменьшать на  $\Delta p$  так, чтобы  $V = \mathbf{p}^T \cdot \mathbf{m} - \lambda(\mathbf{p}^2)^T \sigma^2$  увеличилось на максимально возможное значение. Проблемы:
  - ① Каждый ход изменяется только 2 вероятности, поэтому может долго сходиться к оптимальному значению
  - ② Долгое время работы одного шага –  $O(n^2)$
  - ③ Нет гарантий, что сойдется к оптимальному вектору вероятностей

# Неудачные алгоритмы

слайд 2

- ② Каждый ход увеличиваем одну из вероятностей  $p_i$  на  $\Delta p_{\uparrow}$ , а все остальные ненулевые вероятности – уменьшать на  $\frac{\Delta p_{\uparrow}}{\varphi_{t,i}}$ , где  $\varphi_{t,i} := \varphi_t - I_{p_i \neq 0}$ .

# Неудачные алгоритмы

слайд 2

- ② Каждый ход увеличиваем одну из вероятностей  $p_i$  на  $\Delta p_{\uparrow}$ , а все остальные ненулевые вероятности – уменьшать на  $\frac{\Delta p_{\uparrow}}{\varphi_{t,i}}$ , где  $\varphi_{t,i} := \varphi_t - I_{p_i \neq 0}$ . Проблемы:
- ① Долгое время работы одного шага –  $O(n^2)$
  - ② Нет гарантий, что сойдется к оптимальному вектору вероятностей

# Первый рабочий алгоритм

Замечание 1

$V$  вогнута на  $R^n$

Замечание 2

$\Delta^n$  замкнуто и выпукло в  $R^n$

# Первый рабочий алгоритм

## Замечание 1

$V$  вогнута на  $R^n$

## Замечание 2

$\Delta^n$  замкнуто и выпукло в  $R^n$

Тогда можно найти оптимальный вектор вероятностей с помощью метода градиентного подъема. В качестве алгоритма был взят метод за  $O(1)$ , описанный в статье [2]

# Первый рабочий алгоритм

## Замечание 1

$V$  вогнута на  $R^n$

## Замечание 2

$\Delta^n$  замкнуто и выпукло в  $R^n$

Тогда можно найти оптимальный вектор вероятностей с помощью метода градиентного подъема. В качестве алгоритма был взят метод за  $O(1)$ , описанный в статье [2] Проблема: получаемый вектор может иметь небольшую погрешность

# Второй рабочий алгоритм

## Утверждение 1

Пусть  $w_i = \frac{\partial V}{\partial p_i} = m_i - 2\lambda p_i \sigma_i^2$ . Тогда  $P \in \Delta^n$  является решением уравнения  $\mathbf{p} = \arg \max_{\mathbf{p} \in \Delta^n} \left( \mathbf{p}^T \cdot \mathbf{m} - \lambda (\mathbf{p}^T)^2 \cdot \boldsymbol{\sigma}^2 \right)$  тогда и только тогда, когда  $\forall i, j : i \neq j \wedge p_i \neq 0 \wedge p_j \neq 0 \hookrightarrow w_i(p_i) = w_j(p_j) = w$  и  $\forall i, j : p_i = 0, p_j \neq 0 \hookrightarrow w_i(p_i) \leq w_j(p_j)$

# Второй рабочий алгоритм

## Утверждение 1

Пусть  $w_i = \frac{\partial V}{\partial p_i} = m_i - 2\lambda p_i \sigma_i^2$ . Тогда  $P \in \Delta^n$  является решением уравнения  $\mathbf{p} = \arg \max_{\mathbf{p} \in \Delta^n} \left( \mathbf{p}^T \cdot \mathbf{m} - \lambda (\mathbf{p}^T)^2 \cdot \boldsymbol{\sigma}^2 \right)$  тогда и только тогда, когда  $\forall i, j : i \neq j \wedge p_i \neq 0 \wedge p_j \neq 0 \hookrightarrow w_i(p_i) = w_j(p_j) = w$  и  $\forall i, j : p_i = 0, p_j \neq 0 \hookrightarrow w_i(p_i) \leq w_j(p_j)$

## Утверждение 2

Если упорядочить все  $m_i$  по возрастанию и сопоставить каждому  $m_i$  свой  $p_i$  из оптимального вектора вероятностей, то все нулевые вероятности будут находиться “не правее” ненулевых вероятностей, причем в какой-то точке могут находиться одновременно ненулевые и нулевые вероятности только в том случае, когда неулевым вероятностям соответствуют безрисковые рычаги.

# Второй рабочий алгоритм

Утв. 3 (из портфельной теории Марковица)

Если  $\forall i \sigma_i^2 > 0$  и  $\exists i, j : m_i \neq m_j$ , то существует метод нахождения  $\mathbf{p} = \arg \max_{\mathbf{p} \in R^n} V(\mathbf{p})$  на гиперплоскости  $p_1 + \dots + p_n = 1$ , и для  $\mathbf{p}$  в таком случае верно, что

$$p_i = \frac{m_i}{2\lambda\sigma_i^2} + \frac{1 - \Sigma_1}{\Sigma_0} \cdot \frac{1}{2\lambda\sigma_i^2}$$

где

$$\Sigma_0 = \sum_{i=1}^n \frac{1}{2\lambda\sigma_i^2}, \quad \Sigma_1 = \sum_{i=1}^n \frac{m_i}{2\lambda\sigma_i^2}$$

# Второй рабочий алгоритм

## Утв. 4 (из портфельной теории Марковица)

Если  $\exists! i : \sigma_i^2 = 0$ , то есть существует безрисковый рычаг (пусть его среднее  $m_0$ ), то существует метод нахождения

$\mathbf{p} = \arg \max_{\mathbf{p} \in R^n} V(\mathbf{p})$  на гиперплоскости  $p_1 + \dots + p_n = 1$ , и

$$p_i = \frac{m_i - m_0}{2\lambda\sigma_i^2} \cdot \left(1 + m_0 \frac{\Sigma'_1}{\Sigma'_2}\right), \text{ где } \Sigma'_k = \sum_{i=1}^n \frac{(m_i - m_0)^k}{2\sigma_i^2} \text{ и } i \neq 0.$$

## Второй рабочий алгоритм

Пусть мы отсортировали все рычаги и отбросили все, что хуже наилучшего безрискового. Достаточно найти такое  $t$ , что  $\forall i$  либо  $m_i \leq t \wedge p_i = 0$ , либо  $m_i > t \wedge w_i = t \Leftrightarrow p_i = \frac{m_i - t}{2\lambda\sigma_i^2}$  и  $\sum_{i=1}^n p_i = \sum_{i=1}^n p_i(t) = 1$ .

## Второй рабочий алгоритм

Пусть мы отсортировали все рычаги и отбросили все, что хуже наилучшего безрискового. Достаточно найти такое  $t$ , что  $\forall i$  либо  $m_i \leq t \wedge p_i = 0$ , либо  $m_i > t \wedge w_i = t \Leftrightarrow p_i = \frac{m_i - t}{2\lambda\sigma_i^2}$  и  $\sum_{i=1}^n p_i = \sum_{i=1}^n p_i(t) = 1$ .

Для каждого  $t$  однозначно определен набор тех  $i$ , для которых вероятности ненулевые, а именно такие  $i$ , что  $m_i > t$ . Кроме того, заметим, что при уменьшении  $t$  сумма вероятностей увеличивается, а при  $t = m_{(j)}$  (то есть  $j$ -ая порядковая статистика) верно, что

$$\sum_{i=1}^n p_i(m_{(j)}) = \sum_{i=j+1}^n \frac{m_{(i)} - m_{(j)}}{2\lambda\sigma_{(i)}^2} = \sum_{i=j+1}^n \frac{m_{(i)}}{2\lambda\sigma_{(i)}^2} - m_{(j)} \sum_{i=j+1}^n \frac{1}{2\lambda\sigma_{(i)}^2}$$

# Второй рабочий алгоритм

Если обозначить  $\sum_{i=j+1}^n \frac{m_{(i)}}{2\lambda\sigma_{(i)}^2} := \Sigma_1(j+1)$ , а  
 $\sum_{i=j+1}^n \frac{1}{2\lambda\sigma_{(i)}^2} := \Sigma_0(j+1)$ , то

$$\sum_{i=1}^n p_i(m_{(j)}) = \Sigma_1(j+1) - m_{(j)}\Sigma_0(j+1)$$

причём  $\Sigma_1(n+1) = \Sigma_0(n+1) = 0$  и

$$\Sigma_1(i) = \Sigma_1(i+1) + \frac{m_{(i)}}{2\lambda\sigma_{(i)}^2}, \quad \Sigma_0(i) = \Sigma_0(i+1) + \frac{1}{2\lambda\sigma_{(i)}^2}$$

## Второй рабочий алгоритм

Ввиду увеличения суммы вероятностей оптимальное  $t$  либо лежит на полуинтервале

$(m_{(i)}, m_{(i+1)})$  и  $\sum_{j=1}^n p_j(m_{(i+1)}) \leq 1 < \sum_{j=1}^n p_j(m_{(i)})$ , либо лежит на луче  $(-\infty, m_{(1)})$  и  $\sum_{j=1}^n p_j(m_{(i+1)}) \leq 1$ . Во всех случаях мы можем вычислить  $p_i$  по формулам из пункта, учитывая только те  $i$ , для которых  $m_{(i)} \geq t$ , то есть все  $m_{(i)}$  от конца интервала, где находится  $t$ , и до  $m_{(n)}$ .

# Описание алгоритма

- 1 Сортируем все  $m_i$  по убыванию, в случае равенства по возрастанию  $\sigma_i^2$ . Работает за  $O(n \log n)$ .

# Описание алгоритма

- ① Сортируем все  $m_i$  по убыванию, в случае равенства по возрастанию  $\sigma_i^2$ . Работает за  $O(n \log n)$ .
- ② С начала массива ищем безрисковый рычаг с наибольшим матожиданием. Если нашли (это первый рычаг с  $\sigma_i^2 = 0$ ), то отбрасываем все рычаги правее найденного ( $O(n)$ ).

# Описание алгоритма

- 1 Сортируем все  $m_i$  по убыванию, в случае равенства по возрастанию  $\sigma_i^2$ . Работает за  $O(n \log n)$ .
- 2 С начала массива ищем безрисковый рычаг с наибольшим матожиданием. Если нашли (это первый рычаг с  $\sigma_i^2 = 0$ ), то отбрасываем все рычаги правее найденного ( $O(n)$ ).
- 3 Если отбросили все рычаги, кроме безрискового, то вероятность выбора безрискового рычага равна 1, заканчиваем работу ( $O(1)$ ).

# Описание алгоритма

- ① Сортируем все  $m_i$  по убыванию, в случае равенства по возрастанию  $\sigma_i^2$ . Работает за  $O(n \log n)$ .
- ② С начала массива ищем безрисковый рычаг с наибольшим матожиданием. Если нашли (это первый рычаг с  $\sigma_i^2 = 0$ ), то отбрасываем все рычаги правее найденного ( $O(n)$ ).
- ③ Если отбросили все рычаги, кроме безрискового, то вероятность выбора безрискового рычага равна 1, заканчиваем работу ( $O(1)$ ).
- ④ Иначе считаем  $\Sigma_1(n+1) = \Sigma_0(n+1) = 0$ , проходимся слева направо по  $m_{(i)}$ , пересчитывая за  $O(1)$   $\Sigma_1(i+1)$  и  $\Sigma_0(i+1)$ , вычисляя за  $O(1)$  сумму вероятностей. Находим первое такое  $i$ , что  $\sum_{i=1}^n p_i(m_{(i+1)}) \leq 1$  и  $\sum_{i=1}^n p_i(m_{(i+1)}) > 1$  ( $O(n)$ ).

# Описание алгоритма

- 1 Сортируем все  $m_i$  по убыванию, в случае равенства по возрастанию  $\sigma_i^2$ . Работает за  $O(n \log n)$ .
- 2 С начала массива ищем безрисковый рычаг с наибольшим матожиданием. Если нашли (это первый рычаг с  $\sigma_i^2 = 0$ ), то отбрасываем все рычаги правее найденного ( $O(n)$ ).
- 3 Если отбросили все рычаги, кроме безрискового, то вероятность выбора безрискового рычага равна 1, заканчиваем работу ( $O(1)$ ).
- 4 Иначе считаем  $\Sigma_1(n+1) = \Sigma_0(n+1) = 0$ , проходимся слева направо по  $m_{(i)}$ , пересчитывая за  $O(1)$   $\Sigma_1(i+1)$  и  $\Sigma_0(i+1)$ , вычисляя за  $O(1)$  сумму вероятностей. Находим первое такое  $i$ , что  $\sum_{i=1}^n p_i(m_{(i+1)}) \leq 1$  и  $\sum_{i=1}^n p_i(m_{(i+1)}) > 1$  ( $O(n)$ ).
- 5 Если такое  $i$  нашлось, то  $p_j$  для  $j \geq i + 1$  вычисляются по формулам из пункта а для  $j \leq i \rightarrow p_j = 0$ . Если же не нашлось, то у всех рычагов ненулевые вероятности, и, аналогично, по формулам из пункта вычисляются все  $p_i$



# Особенности алгоритма

Плюсы:

- ➊ Работает за  $O(n \log n)$  вместо  $O(n^2)$ , причем этот  $n \log n$  “лёгкий”, поскольку в него включена лишь сортировка.

# Особенности алгоритма

Плюсы:

- ① Работает за  $O(n \log n)$  вместо  $O(n^2)$ , причем этот  $n \log n$  “лёгкий”, поскольку в него включена лишь сортировка.
- ② Дает точное решение

# Особенности алгоритма

Плюсы:

- ① Работает за  $O(n \log n)$  вместо  $O(n^2)$ , причем этот  $n \log n$  “лёгкий”, поскольку в него включена лишь сортировка.
- ② Дает точное решение

Минусы:

- ① Проблема холодного старта

# Модификации исходных алгоритмов

## 1 $\epsilon$ -greedy стратегии:

- Классический
- Adaptive: на  $t$ -ом шаге  $\epsilon_t = \frac{1}{t}$
- VDBE:  $\epsilon_{t+1} = \delta \frac{|e^{w_t(a)/\tau} - e^{w_{t+1}(a)/\tau}|}{e^{w_t(a)/\tau} + e^{w_{t+1}(a)/\tau}} + (1 - \delta)\epsilon_t$ , где  $w_t(a) = \frac{\partial V}{\partial p_a}$  на  $t$ -ом шаге,  $\tau$  – температура

# Модификации исходных алгоритмов

## 1 $\epsilon$ -greedy стратегии:

- Классический
- Adaptive: на  $t$ -ом шаге  $\epsilon_t = \frac{1}{t}$
- VDBE:  $\epsilon_{t+1} = \delta \frac{|e^{w_t(a)/\tau} - e^{w_{t+1}(a)/\tau}|}{e^{w_t(a)/\tau} + e^{w_{t+1}(a)/\tau}} + (1 - \delta)\epsilon_t$ , где  $w_t(a) = \frac{\partial V}{\partial p_a}$  на  $t$ -ом шаге,  $\tau$  – температура

## 2 Positive initialization: инициализируем

$$Q_0(a) = q > 0, \quad Q_0^2(a) = q^2.$$

# Модификации исходных алгоритмов

## 1 $\epsilon$ -greedy стратегии:

- Классический
- Adaptive: на  $t$ -ом шаге  $\epsilon_t = \frac{1}{t}$
- VDBE:  $\epsilon_{t+1} = \delta \frac{|e^{w_t(a)/\tau} - e^{w_{t+1}(a)/\tau}|}{e^{w_t(a)/\tau} + e^{w_{t+1}(a)/\tau}} + (1 - \delta)\epsilon_t$ , где  $w_t(a) = \frac{\partial V}{\partial p_a}$  на  $t$ -ом шаге,  $\tau$  – температура

## 2 Positive initialization: инициализируем

$$Q_0(a) = q > 0, \quad Q_0^2(a) = q^2.$$

## 3 Аналог UCB: В алгоритме за $O(n \log n)$ считаем, что

$$m_i = q_i + c \sqrt{\frac{\ln T}{N_t(a)}}$$

# Модификации исходных алгоритмов

## 1 $\epsilon$ -greedy стратегии:

- Классический
- Adaptive: на  $t$ -ом шаге  $\epsilon_t = \frac{1}{t}$
- VDBE:  $\epsilon_{t+1} = \delta \frac{|e^{w_t(a)/\tau} - e^{w_{t+1}(a)/\tau}|}{e^{w_t(a)/\tau} + e^{w_{t+1}(a)/\tau}} + (1 - \delta)\epsilon_t$ , где  $w_t(a) = \frac{\partial V}{\partial p_a}$  на  $t$ -ом шаге,  $\tau$  – температура

## 2 Positive initialization: инициализируем

$$Q_0(a) = q > 0, \quad Q_0^2(a) = q^2.$$

## 3 Аналог UCB: В алгоритме за $O(n \log n)$ считаем, что

$$m_i = q_i + c \sqrt{\frac{\ln T}{N_t(a)}}$$

## 4 Аналог gradient bandits

# Итоговые эксперименты

# Параметры

- Кол-во тестов – 2000, длина теста – 1000 шагов

# Параметры

- Кол-во тестов – 2000, длина теста – 1000 шагов
- Распределения рычагов (все рычаги брались из одного семейства распределений):
  - Нормальное ( $t_\infty$ )
  - Распределение Стьюдента с  $\nu = 3$  ( $t_3$ )
  - Распределение Стьюдента с  $\nu = 2.1$  ( $t_{2.1}$ )
  - Распределение Стьюдента с 2 степенями свободы  $t_2$

# Параметры

- Кол-во тестов – 2000, длина теста – 1000 шагов
- Распределения рычагов (все рычаги брались из одного семейства распределений):
  - Нормальное ( $t_\infty$ )
  - Распределение Стьюдента с  $\nu = 3$  ( $t_3$ )
  - Распределение Стьюдента с  $\nu = 2.1$  ( $t_{2.1}$ )
  - Распределение Стьюдента с 2 степенями свободы  $t_2$
- Матожидание бралось из  $N(1, 1)$ , дисперсия бралась из  $Exp(2)$

# Метрики

- Среднее сожаление: если  $V_{max} = (\mathbf{p}_{max}^T \cdot \mathbf{m} - \lambda(\mathbf{p}_{max}^2)^T \cdot \boldsymbol{\sigma}^2)$ , где  $\mathbf{p}_{max}$  – оптимальный вектор вероятностей, то  $Reg_t = V_{max} - (\mathbf{p}_t^T \cdot \mathbf{q} - \lambda(\mathbf{p}_t^2)^T \cdot \mathbf{s}^2)$ . Если  $Reg_t < 0$ , то алгоритм “переоценивает” себя.

# Метрики

- Среднее сожаление: если  $V_{max} = (\mathbf{p}_{max}^T \cdot \mathbf{m} - \lambda(\mathbf{p}_{max}^2)^T \cdot \boldsymbol{\sigma}^2)$ , где  $\mathbf{p}_{max}$  – оптимальный вектор вероятностей, то  $Reg_t = V_{max} - (\mathbf{p}_t^T \cdot \mathbf{q} - \lambda(\mathbf{p}_t^2)^T \cdot \mathbf{s}^2)$ . Если  $Reg_t < 0$ , то алгоритм “переоценивает” себя.
- Среднее реальное сожаление:  
 $RReg_t = V_{max} - (\mathbf{p}_t^T \cdot \mathbf{m} - \lambda(\mathbf{p}_t^2)^T \cdot \boldsymbol{\sigma}^2)$ . Всегда  $\geq 0$ .

# Метрики

- Среднее сожаление: если  $V_{max} = (\mathbf{p}_{max}^T \cdot \mathbf{m} - \lambda(\mathbf{p}_{max}^2)^T \cdot \boldsymbol{\sigma}^2)$ , где  $\mathbf{p}_{max}$  – оптимальный вектор вероятностей, то  $Reg_t = V_{max} - (\mathbf{p}_t^T \cdot \mathbf{q} - \lambda(\mathbf{p}_t^2)^T \cdot \mathbf{s}^2)$ . Если  $Reg_t < 0$ , то алгоритм “переоценивает” себя.
- Среднее реальное сожаление:  $RReg_t = V_{max} - (\mathbf{p}_t^T \cdot \mathbf{m} - \lambda(\mathbf{p}_t^2)^T \cdot \boldsymbol{\sigma}^2)$ . Всегда  $\geq 0$ .
- Процент оптимальных действий:

$$\delta = 1 - \frac{1}{2} \sum_{i=1}^n |p_i - p_{i,max}|$$

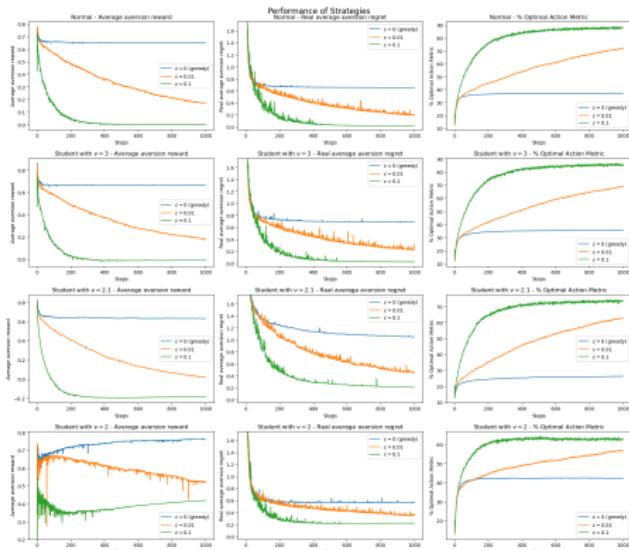
# Графики

Были построены графики, сгруппированные по каждому семейству распределений и по каждому алгоритму. Кроме того, были построены графики для различных значений  $\lambda \in [0.1, 0.3, 0.6, 1, 2, 4]$ .

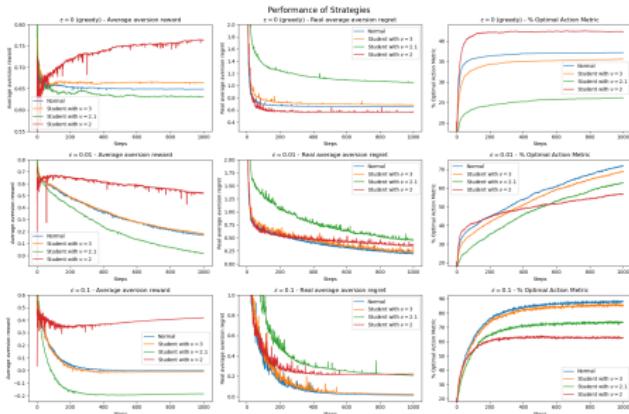
# Первый рабочий алгоритм

Предварительно был реализован первый рабочий алгоритм через градиентный подъем, после чего на примерах были сравнены оба алгоритма. Результаты совпадают в пределах погрешности.

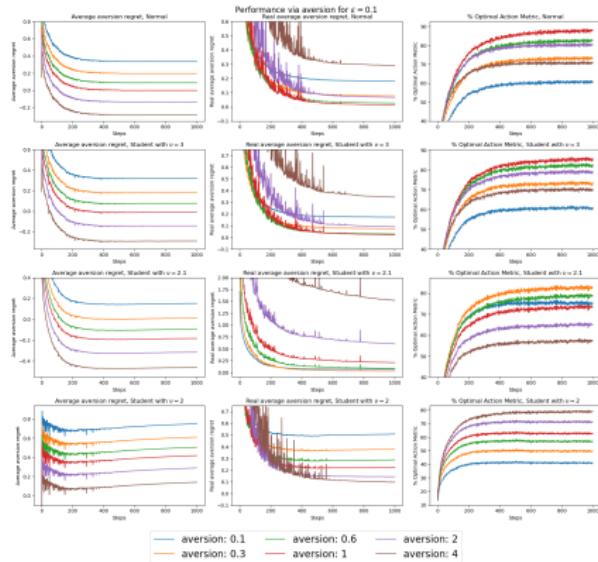
# Результаты – $\epsilon$ -greedy



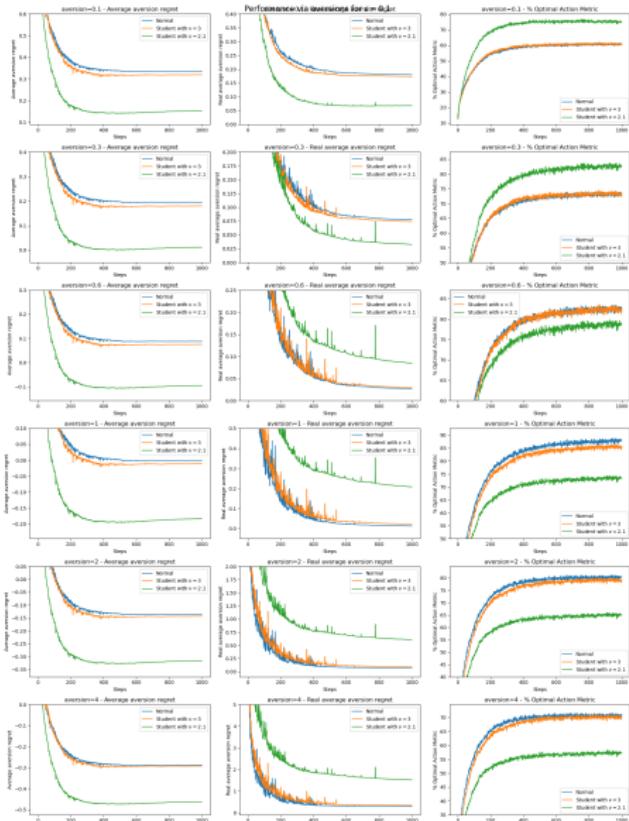
# Результаты – $\epsilon$ -greedy



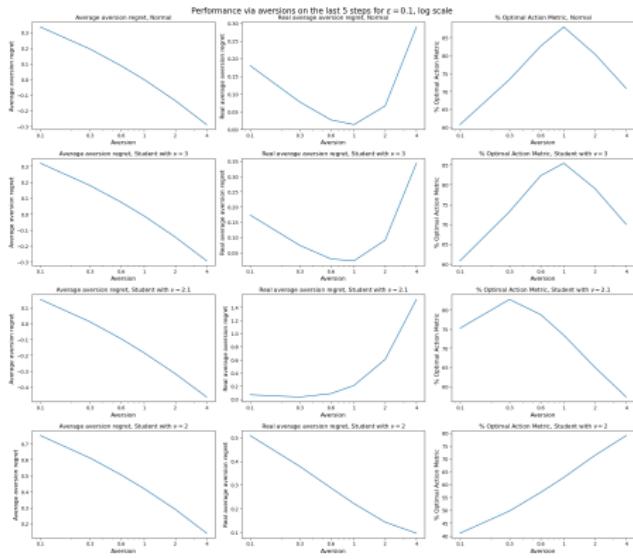
# Результаты – $\epsilon$ -greedy



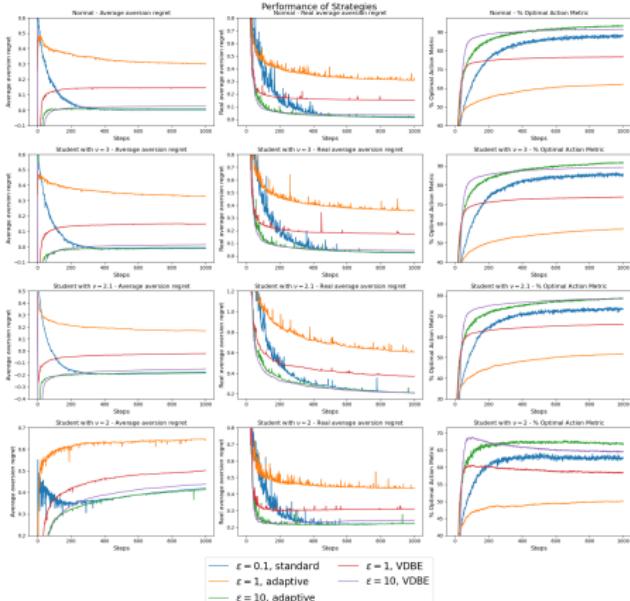
# Результаты – $\epsilon$ -greedy



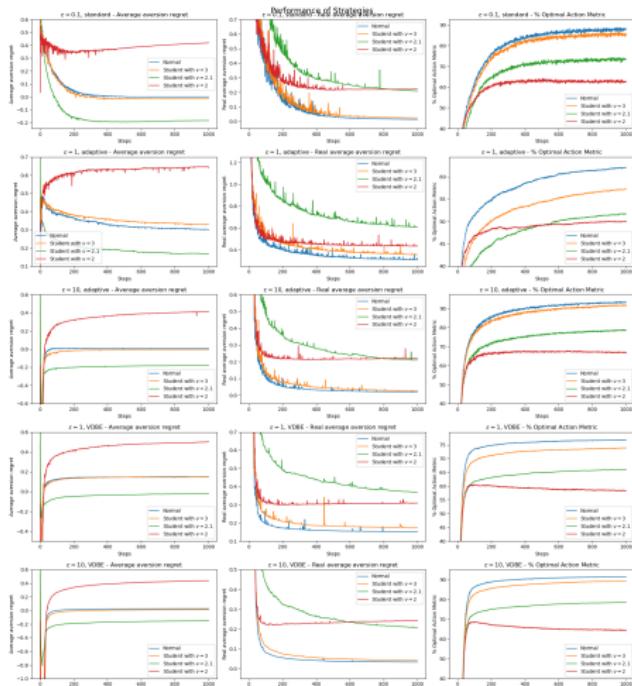
# Результаты – $\epsilon$ -greedy



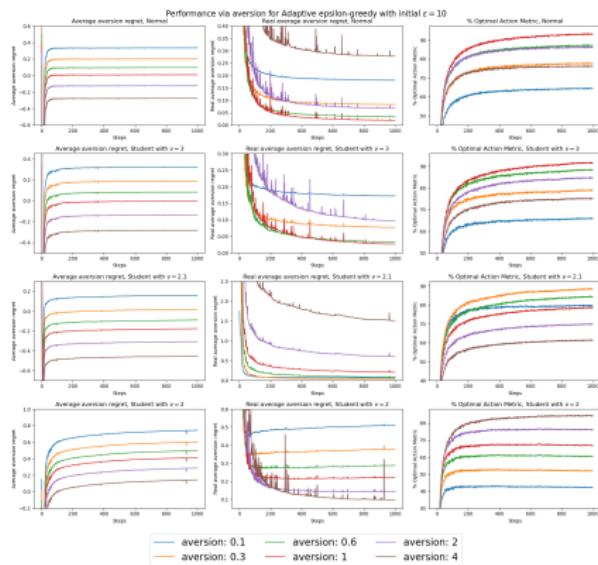
# Результаты – adaptive $\epsilon$



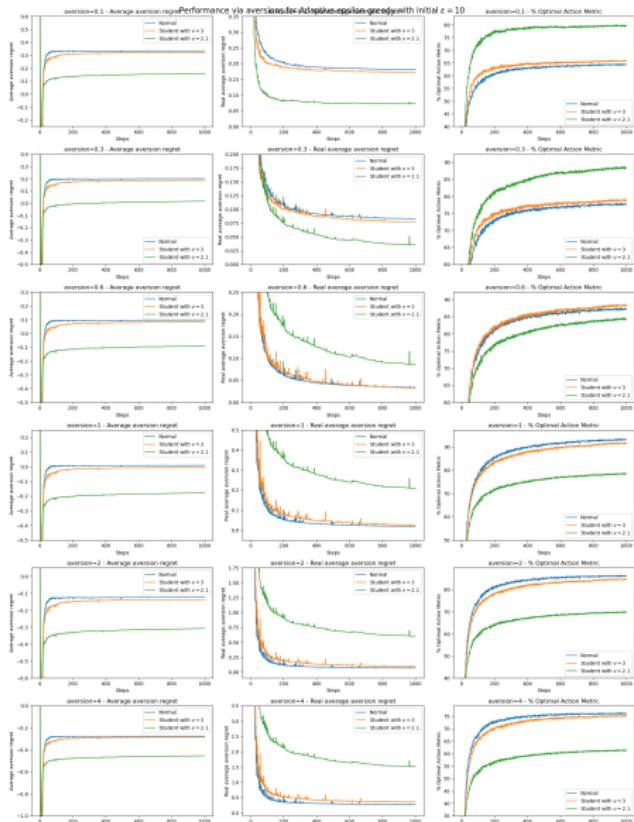
# Результаты – adaptive $\epsilon$



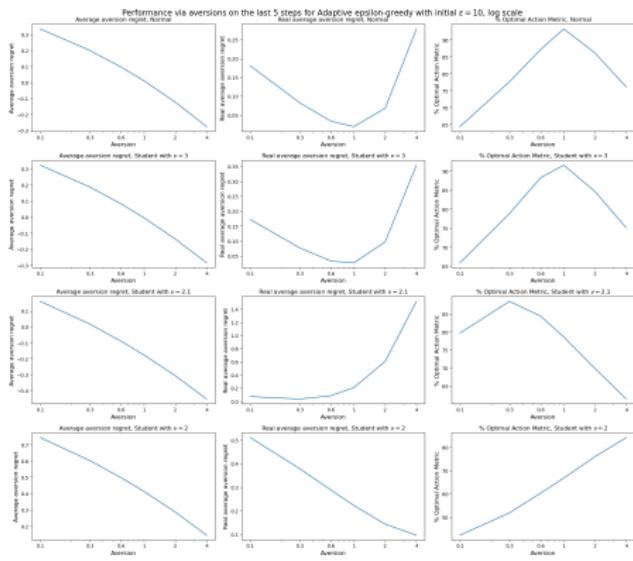
# Результаты – adaptive $\epsilon$



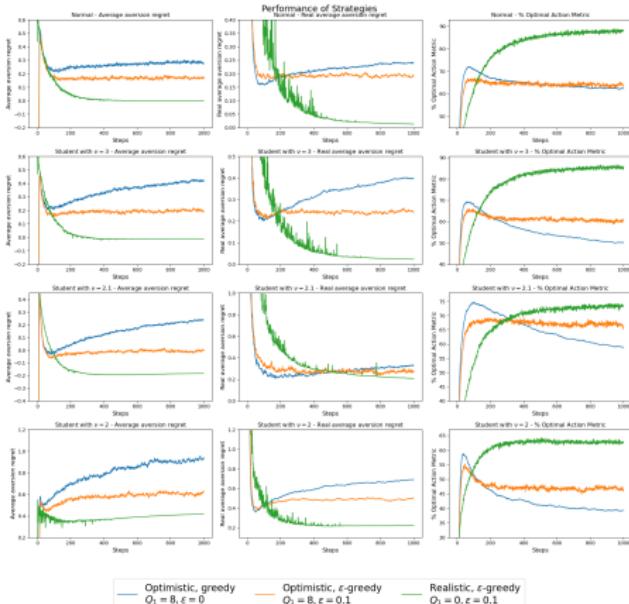
# Результаты – adaptive $\epsilon$



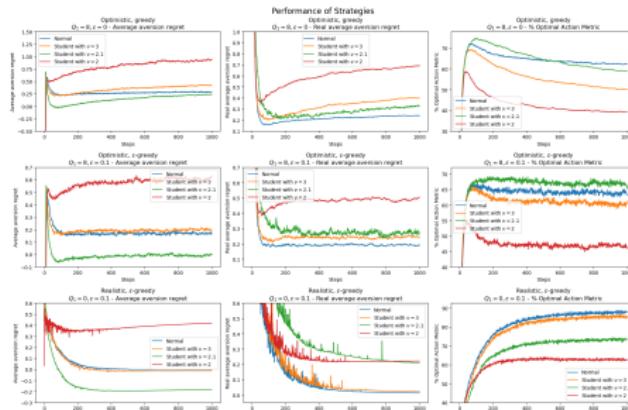
# Результаты – adaptive $\epsilon$



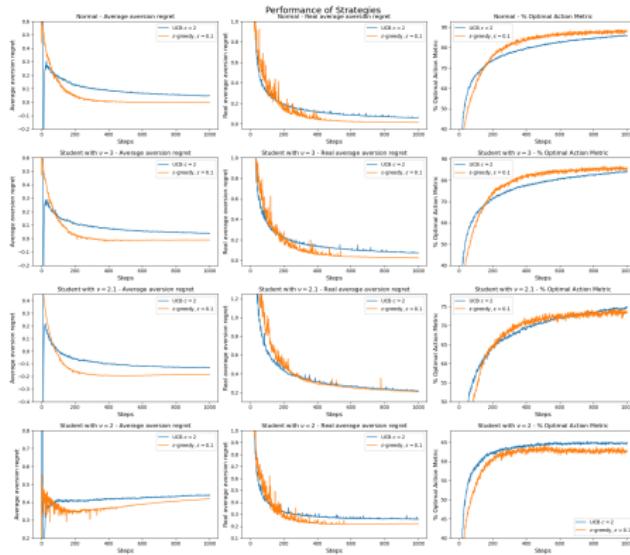
# Результаты – positive initialization



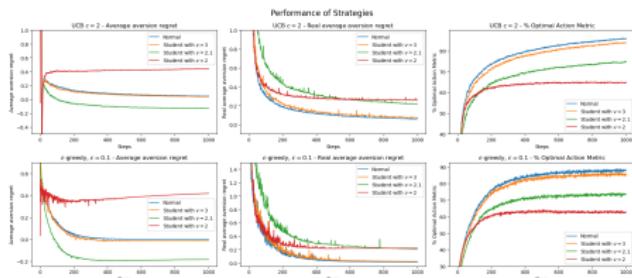
# Результаты – positive initialization



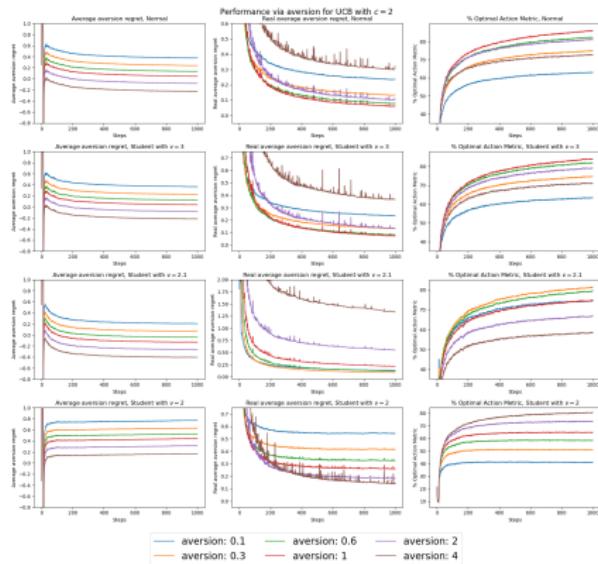
# Результаты – UCB



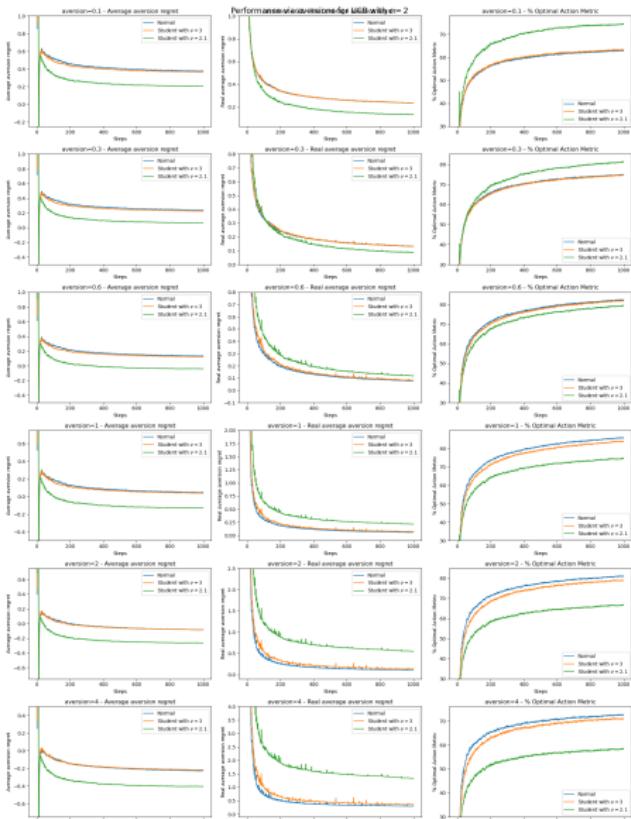
# Результаты – UCB



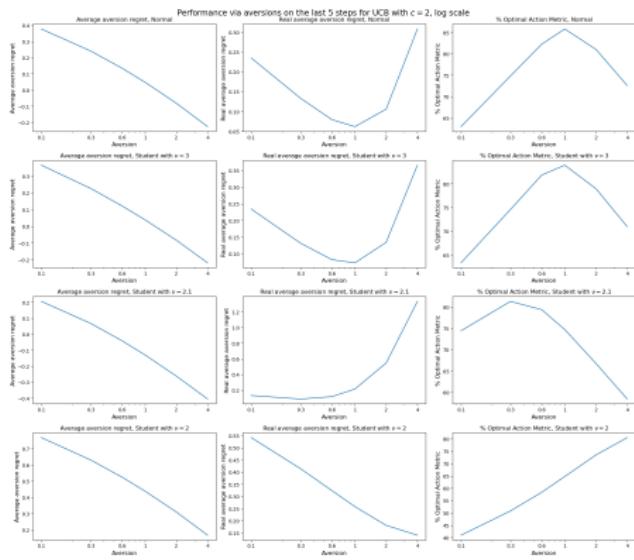
# Результаты – UCB



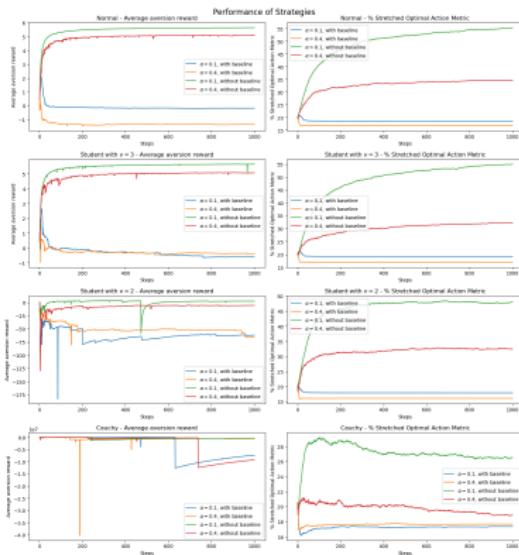
# Результаты – UCB



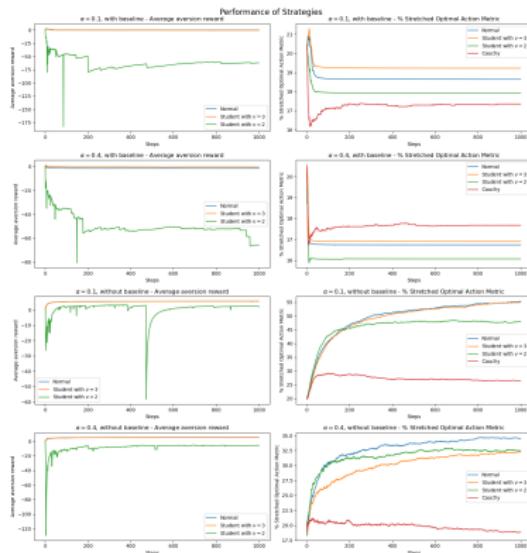
# Результаты – UCB



# Результаты – gradient bandits



# Результаты – gradient bandits



# Выводы

- Алгоритм в сочетании с  $\epsilon$ -greedy дает отличное приближение вероятностей.

# Выводы

- Алгоритм в сочетании с  $\epsilon$ -greedy дает отличное приближение вероятностей.
- При уменьшении  $\nu$  VDBE и UCB начинают работать лучше, чем  $\epsilon$ -greedy.

# Выводы

- Алгоритм в сочетании с  $\epsilon$ -greedy дает отличное приближение вероятностей.
- При уменьшении  $\nu$  VDBE и UCB начинают работать лучше, чем  $\epsilon$ -greedy.
- График зависимости точности оптимальных действий от коэффициента неприятния к риску имеет выраженный максимум.

# Заключение

# Заключение

В результате написания работы:

- Проанализированы известные подходы в классической задаче о многоруких бандитах на предмет применимости для распределений, отличных от нормального.

# Заключение

В результате написания работы:

- Проанализированы известные подходы в классической задаче о многоруких бандитах на предмет применимости для распределений, отличных от нормального.
- Придуманы алгоритмы и подходы для решения задачи о многоруких бандитах с учетом степени отвращения к риску.

# Заключение

В результате написания работы:

- Проанализированы известные подходы в классической задаче о многоруких бандитах на предмет применимости для распределений, отличных от нормального.
- Придуманы алгоритмы и подходы для решения задачи о многоруких бандитах с учетом степени отвращения к риску.
- Протестированы созданные подходы на различных распределениях, получены хорошие результаты.

# Источники

# Источники

- [1] Jean-Philippe Bouchaud и Mark Potters. *Theory of Financial Risk and Derivative Pricing*. 2003. Гл. 10.1.
- [2] James Chok и Geoffrey M. Vasil. «Convex Optimization Over a Probability Simplex». В: (2023), с. 1—6.
- [3] Richard S. Sutton и Andrew G. Barto. *Reinforcement Learning: An Introduction, second edition*. 2018. Гл. 2.