Реализация полученных теоретических алгоритмов для задачи о многоруких бандитах с учетом степени отвращения к риску

Михаил Давыдов

12 июня 2024 г.

Аннотация

В этой статье описываются эксперименты по применению теоретических результатов, полученных в прошлой работе, для задачи о многоруких бандитах с учетом степени отвращения к риску, приводятся обоснования полученных результатов.

1 Введение

В предыдущей работе были приведены возможные алгоритмы, которые можно применять для аппроксимации оптимального решения для задачи о многоруких бандитах с учетом степени отвращения к риску. В этой статье будут показаны проводимые эксперименты, построены графики для интерпретации результатов. Напомним, если дано n рычагов, и каждый рычаг i соответствует распределению со средним m_i и дисперсией σ_i^2, то цель – найти такой вероятностный вектор $\mathbf{p}^* \in \Delta^n$, что $\mathbf{p}^* = \arg\max\limits_{\mathbf{p}\in\Delta^n} \sum_{i=1}^n p_i m_i - \lambda \sum_{i=1}^n p_i^2 \sigma_i^2$. При этом начальная информация о рычагах неизвестна, и каждый ход, выбирая один из рычагов, мы получаем награду из распределения, соответствующего этому рычагу.

2 Проверка greedy-алгоритмов

Algorithm 1 Our proposed algorithm

Require: $\varepsilon \leftarrow 10^{-10}$ (Tolerance for the zero set)
$w \leftarrow (1/n, \cdots, 1/n)$
while termination conditions not met **do**
$S \leftarrow \{i = 1, \cdots, n \mid w_i > \varepsilon\}$
$Q \leftarrow \{i = 1, \cdots, n \mid w_i \leq \varepsilon\}$

Choose $\eta_t \geq 0$
$\eta_{\max} \leftarrow \dfrac{1}{\max_{i\in S}(\nabla f_i) - w \cdot \nabla f}$
$\eta_t \leftarrow \min(\eta_t, \eta_{\max})$

$\hat{w}^{t+1} \leftarrow w^t - \eta_t w^t (\nabla f - w^t \cdot \nabla f)$

$\hat{w}_j^{t+1} \leftarrow 0, \quad \forall j \in Q$
$w_i^{t+1} \leftarrow \hat{w}_i^{t+1} / \sum_j \hat{w}_j^{t+1}$ (Normalizing for numerical stability)
end while

Рис. 1: Псевдокод для алгоритма CauchySimplex

Для начала была проведена проверка 2 алгоритмов для нахождения решения задачи в случае, когда все матожидания и дисперсии известны. Первый алгоритм – алгоритм нахождения максимума на симплексе, представленный в конце главы “Стратегии” (назовем его StandardGreedy).

Второй алгоритм – алгоритм градиентного подъема CauchySimplex, представленный в [CV23]. Среди неградиентных методов был рассмотрен только метод StandardGreedy, поскольку он обладает самой меньшим алгоритмической сложностью шага $O(n \log n)$. В CauchySimplex выбраны такие гиперпараметры: $\epsilon = 10^{-3}$, $stop = 10^{-9}$ – когда изменение $V = \sum_{i=1}^{n} p_i m_i - \lambda \sum_{i=1}^{n} p_i^2 \sigma_i^2$ за один шаг меньше, чем $stop$, то алгоритм останавливается (см. 1).

Далее происходит запуск 500 тестов, в каждом тесте было 10 распределений со средними, выбранными случайно из $\mathcal{N}(1, 1)$, и дисперсиями σ^2, т.ч. $\sigma \sim Exp(2)$. Чтобы включить среди рычагов "безрисковые" рычаги, каждая из дисперсий с вероятностью $\frac{1}{n}$ $(n = 10)$ домножалось на 0. Для каждого такого набора матожиданий и дисперсий запускались алгоритмы StandardGreedy и CauchySimplex, после чего полученные вероятности сравнивались. Если хотя бы 2 вероятности отличаются больше, чем на гиперпараметр error_rate, тест считался проваленным. Алгоритм, сравнивающий подходы, был запущен 2 раза для error_rate= 0.02 и error_rate= 0.05.

В результате при error_rate= 0.02 количество проваленных тестов равнялось 2.8%, а при error_rate= 0.05 все тесты прошли успешно (см. 2). При этом среднее выполнение CauchySimplex составляет от 90 до 160 миллисекунд, в то время как StandardGreedy – меньше 0.1 миллисекунды. Это позволяет сделать 3 вывода:

1. Ввиду того, что CauchySimplex относится к градиентным методам и потому обладает некоторой погрешностью, и этот алгоритм выдает оптимальное решение, то и алгоритм StandardGreedy выдает оптимальное решение.
2. Погрешность метода CauchySimplex иногда достаточно большая.
3. Алгоритм StandardGreedy намного быстрее CauchySimplex.

На основании этого можно сделать вывод, что StandardGreedy более применим для проведения экспериментов.

```
100%|██████████| 500/500 [01:18<00:00,  6.33it/s]
Quantity of failed tests: 2.8%
Mean time for cauchy_simples: 156.37 milliseconds
Mean time for greedy: 0.07 milliseconds
100%|██████████| 500/500 [00:48<00:00, 10.28it/s]
Quantity of failed tests: 0.0%
Mean time for cauchy_simples: 95.91 milliseconds
Mean time for greedy: 0.06 milliseconds
```

Рис. 2: Результаты сравнения алгоритмов CauchySimplex и StandardGreedy

3 Методология

Для рассмотрения алгоритмов было использовано 10 рычагов. Все рычаги относятся к одному семейству распределений. Было рассмотрено 4 разлтчных семейства распределений: нормальное распределение ($\mathcal{N}(a, \sigma^2)$ или t_∞), которое является предельным случаем распределения Стьюдента при числе степеней свободы, стремящемся к бесконечности; распределение Стьюдента с 3 степенями свободы (t_3), домноженное на $\frac{1}{\sqrt{3}} \cdot \sigma$, чтобы дисперсия была равна σ^2 (значение σ^2 будет введено позже); распределение Стьюдента с $\mu = 2.1$ ($t_{2.1}$) домноженное на $\frac{1}{\sqrt{21}} \cdot \sigma$, чтобы дисперсия была равна σ^2; распределение Стьюдента с 2 степенями свободы t_2 домноженное на σ_{scale}. Хотя для последнего распределения цель максимизации $\sum_{i=1}^{n} p_i m_i - \lambda \sum_{i=1}^{n} p_i^2 \sigma_i^2$ некорректна, поскольку у t_2 нет дисперсии, можно вместо дисперсии σ^2 подставить параметр растяжения σ^2_{scale}, возведенный в квадрат, то есть пытаться найти $\sum_{i=1}^{n} p_i m_i - \lambda \sum_{i=1}^{n} p_i^2 \sigma^2_{i,\text{scale}}$. Естественно, вся разработанная до этого теория не работает для максимизации измененной величины.
Каждое из распределений было смещено на значение, выбранное случайно из $\mathcal{N}(1, 1)$, а также

домножено на $\sigma \sim Exp(2)$. Чтобы включить среди рычагов "безрисковые" рычаги, после домножения на σ каждое из распределений с вероятностью $\frac{1}{n}$ ($n = 10$) домножалось на 0. Таким образом, с вероятностью примерно $1 - e^{-1} \approx 63\%$ среди рычагов есть хотя бы один с нулевой дисперсией.

Изнаально выбирались следующие стратегии:

1. ϵ-greedy стратегии, то есть стратегии, действующие по формуле

$$\mathbf{p}_t = \begin{cases} \arg\max\limits_{\mathbf{p}\in\Delta^n} \sum_{i=1}^n p_i Q_t(i) - \lambda \sum_{i=1}^n p_i^2 S_t^2(i), & \text{with probability } 1-\epsilon, \\ (\frac{1}{n}, ..., \frac{1}{n}), & \text{with probability } \epsilon. \end{cases}$$

   Максимум здесь и далее находится с помощью алгоритма StandardGreedy. Далее выбираются ϵ-greedy стратегии с $\epsilon = 0$ (то есть просто жадная стратегия), $0.01, 0.1$. Изначально $\forall a\ Q_t(a) = 0,\ Q_t^2(a) = 0$.

2. ϵ-greedy стратегии с $\epsilon = 0.1$ и скорректированной дисперсией. Об этом подробнее в этом разделе ВСТАВЬ ПОТОМ ССЫЛКУ!!!

3. Далее берутся ϵ-greedy стратегии с адаптивным ϵ, то есть меняющимся со временем. Были взяты adaptive ϵ-greedy (когда $\epsilon = \frac{\epsilon_0}{t}$) с $\epsilon_0 = 1$ и $\epsilon_0 = 10$ (если $\epsilon > 1$, то берется $\epsilon = 1$), и VDBE с $\tau = 1$, $\delta = 0.1$, примененные к $\epsilon_0 = 1$ и $\epsilon = 10$. Полученные результаты были сравнены с ϵ-greedy стратегией с $\epsilon = 0.1$.

4. Затем были взяты стратегии с оптимистичной инициализацией, то есть стратегии, для которых изначально $\forall a\ Q_t(a) = d,\ d > 0$. В качестве d был выбран $d = 6$. Обновление $Q_t(a)$ происходит с константынм шагом (step-size) в соответствии с материалом на этой странице. Был выбран step-size= 0.1. Сравниваются greedy-стратегию с оптимистичной инициализацией с 0.1-greedy стратегиями с обычной и оптимистичной инициализациями.

5. Upper-Confidence-Bound Action Selection – вместо $Q_t(a)$ берется верхняя граница доверительного интервала для $Q_t(a)$. Поскольку неравенства Хаффдинга дли дисперсии не существует, а в классическом UCB для обоснования причины, почему в качестве верхней граница доверительного интервала берется $Q_t(a) + c\sqrt{\frac{ln\ t}{N_t(a)}}$, используется именно неравенство Хаффдинга [Sli19], то подаваемая в StandardGreedy дисперсия не меняется. Итоговая формула выглядит так:

$$\mathbf{p}_t = \arg\max_{\mathbf{p}\in\Delta^n} \sum_{i=1}^n p_i \cdot \left[Q_t(i) + c\sqrt{\frac{ln\ t}{N_t(i)}} \right], -\lambda \sum_{i=1}^n p_i^2 S_t^2(i)\ c > 0$$

   То есть вместо $Q_t(a)$ в StandardGreedy подается $Q_t(i) + c\sqrt{\frac{ln\ t}{N_t(i)}}$. В случае, когда $N_t(a) = 0$, подставляется $N_t(a) = 0.001$. UCB с $c = 2$ сравнивается с 0.1-greedy стратегией.

6. Gradient bandits: ИСПРАВЬ КОД И НАПИШИ В ИЗМЕНЕННОМ ВИДЕ. Я сравнил результаты gradient bandits для $\alpha = 0.1,\ 0.4$, и в случаях, когда baseline есть и когда его нет. При этом моды распределений специально смещены на 4, чтобы показать, что, в отличие от других стратегий gradient bandit невосприимчив к смещению распределений.

Каждая из стратегий была запущена для каждого из распределений на 2000 независимых тестах (для каждого теста свое смещение распределений) длиной 1000 шагов каждый, после чего посчитаны среднее сожаление $\text{Regret} = \left(\sum_{i=1}^n p_i^* Q_t(i) - \lambda \sum_{i=1}^n (p_i^*)^2 S_t^2(i)\right) - \left(\sum_{i=1}^n p_i Q_t(i) - \lambda \sum_{i=1}^n (p_i)^2 S_t^2(i)\right)$, среднее реальное сожаление $\text{Regret}_{\text{real}} = \left(\sum_{i=1}^n p_i^* m_i - \lambda \sum_{i=1}^n (p_i^*)^2 \sigma_i^2\right) - \left(\sum_{i=1}^n p_i m_i - \lambda \sum_{i=1}^n (p_i)^2 \sigma_i^2\right)$ и процент выбранных оптимальных действий на каждом шаге $Opt = 1 - 2\sum_{i=1}^n |p_i - p_i^*|$, где $\mathbf{p}^*$ – решение исходной задачи. Для каждого распределения и каждой метрики результаты для одной группы стратегий визуализированы на графике. Кроме того, для каждой стратегии и для каждой метрики на одном графике были изображены результаты по всем распределениям.
Так как коэффициент отвращения к риску λ также может влиять на эффективность алгоритмов, то будем дополнительно строить графики зависимости метрик от числа шагов для разных

распределений и для одной фиксированной стратегии. Дополнительно построим график средних значений метрик на последних 5 шагах от λ в зависимости от распределения, чтобы понять изменение конечного результата работы алгоритмов в зависимости от важности риска.

Сам код можно найти в этом репозитории в папке "theory_tester".

4 Результаты

На всех графиках графики средней реального сожаления и процента оптимальных действий для распределения t_2 не имеет смысла ввиду отсутствия у распределения дисперсии.

TODO: ПЕРЕНЕСТИ РЕЗУЛЬТАТЫ ИЗ ЮПИТЕР-НОУТБУКА.

Список литературы

[CV23] James Chok and Geoffrey M. Vasil. Convex Optimization Over a Probability Simplex. pages 1–6, 2023.

[Sli19] Aleksandrs Slivkins. *Introduction to Multi-Armed Bandits*, chapter 1.3. Foundations and Trends in Machine Learning, 2019.