

УДК 519.81, 004.021, 004.942

ББК 22.18

МОДЕЛЬ ПРИНЯТИЯ РЕШЕНИЙ ПРИ НАЛИЧИИ ЭКСПЕРТОВ КАК МОДИФИЦИРОВАННАЯ ЗАДАЧА О МНОГОРУКОМ БАНДИТЕ

ДМИТРИЙ С. СМИРНОВ

ЕКАТЕРИНА В. ГРОМОВА*

Санкт-Петербургский государственный университет
199034, Санкт-Петербург, Университетская наб., 7-9
e-mail: st016315@student.spbu.ru, e.v.gromova@spbu.ru

В работе сформулирована модификация задачи о многоруком бандите, позволяющая игроку в процессе принятия решения использовать так называемые экспертные подсказки. Под игроком в данной задаче понимается некоторая автоматизированная система, использующая определенную стратегию (алгоритм) для принятия решения в условиях неопределенности. Подход развит для случая m экспертов. Предложена модификация известного алгоритма UCS1 для решения задачи о многоруком бандите. Приведены результаты численного эксперимента, показывающие, каким образом экспертные подсказки влияют на величину выигрыша игрока.

Ключевые слова: задача о многоруком бандите, принятие решений, методы оптимизации, алгоритмы машинного обучения.

1. Введение

Задача о многоруком бандите впервые была сформулирована в работе [21]. В настоящее время данная математическая модель находит свое применение в различных сферах человеческой деятельности, где в условиях неопределенности необходимо принимать решения с целью максимизации выигрыша.

Такие задачи часто возникают в реальных ситуациях, например, в области клинических исследований (назначение методов лечения, минимизируя потери среди пациентов), при адаптивной маршрутизации (минимизация задержек в сети), в финансах (формирование инвестиционного портфеля) и т.д. (см. [5, 9, 12, 13, 23]). Отдельно стоит отметить многочисленные приложения задачи о многоруком бандите в области интернет-маркетинга, например, тестирование интернет-страниц с целью увеличения конверсии [5, 4], персонализация web-контента [18], динамическое изменение цен в интернет-магазине [22], публикация контента в социальных сетях [14] и размещение рекламных объявлений (баннеров) для максимизации дохода [20].

Задача о многоруком бандите (*multi-armed bandit problem*) также известна как задача оптимального управления в случайной среде [3], которая заключается в поиске компромисса между обучением и применением ранее полученных знаний с целью максимизации выигрыша. Формальное определение задачи будет дано ниже, а здесь ограничимся лишь её словесным описанием. Представим ситуацию, в которой игрок многократно выбирает одну из n альтернатив (действий), каждый раз получая за это некоторую награду, величина которой зависит от неизвестного вероятностного распределения. Каждый такой выбор будем называть *игрой*. Задача игрока состоит в том, чтобы максимизировать суммарный выигрыш за конечное число последовательных игр. Заметим, что в качестве игрока, как правило, выступает некоторая автоматизированная система. Каждую альтернативу можно интерпретировать как рычаг игрового автомата, называемого «одноруким бандитом». Тогда задача представляется в виде игрового автомата с несколькими рычагами, который по аналогии можно назвать «многоруким бандитом».

Задача о многоруком бандите была сформулирована в работе [21]. В другой работе [15] того же автора вводится так называемая *функ-*

ция сожаления и доказывается, что она асимптотически не меньше $\ln(T)$, или, более формально, $R(T) = \Omega(\ln(T))$.

В работе [25] представлены некоторые алгоритмы для решения задачи о многоруком бандите. Среди них ε -жадный (ε -greedy), softmax, алгоритм преследования (pursuit) и алгоритм сравнения с подкреплением (reinforcement comparison).

В работе [10] впервые предложена стратегия UCB для задачи о двуруком бандите. Оптимальность стратегии UCB впервые доказана в статье [16].

Большую значимость имеет работа [7], в которой предложены модифицированный ε -жадный алгоритм (ε_n -greedy) и модификации алгоритма UCB1: UCB1-Normal (для нормального распределения) и UCB1-Tuned (с учетом дисперсии). Для всех алгоритмов, за исключением последнего, найдены теоретические оценки функции сожаления.

На практике в условиях задачи о многоруком бандите кроме значений полученных выигрышей при выборе альтернатив часто доступна некоторая дополнительная информация, которую можно использовать для улучшения выбора. В интернет-маркетинге это может быть индивидуальная информация о посетителе веб-ресурса, например, источник перехода, предполагаемый пол, возраст и т.д. Модификация задачи о многоруком бандите, которая учитывает дополнительную информацию, называется *задачей о контекстном бандите (contextual bandit problem)* [17,11].

В литературе данную задачу можно встретить под разными названиями, включая следующие: «задача об ассоциативном бандите» (associative bandit problem) [24], «задача о многоруком бандите с советами эксперта» (multi-armed bandit problem with expert advice) [8], «задача о многоруком бандите с внешней информацией» (multi-armed bandit problem with side information) [19] и «задача о многоруком бандите с ковариатами» (bandit problems with covariates) [27]. Название «contextual bandit problem» впервые использовано в статье [17], причем настолько удачно, что в дальнейшем стало основным.

В данной работе выделяется особый тип дополнительной информации, содержащий прямые рекомендации по выбору альтернатив. Такого рода информация формализуется в виде *экспертных подска-*

зок. Для учета подсказок эксперта сформулирована новая постановка задачи о многоруком бандите, а также модифицирован алгоритм для её решения.

Статья имеет следующую структуру. В разделе 2 приведено формальное описание стохастической задачи о многоруком бандите, сформулирован известный алгоритм UCS1. Раздел 3 посвящен модифицированной задаче о многоруком бандите при наличии эксперта: сформулирована новая постановка задачи, модифицирован алгоритм UCS1, приведены результаты численного эксперимента. В разделе 4 задача о многоруком бандите рассматривается для случая m экспертов. Предложены три метода формирования экспертной подсказки, а также приведены результаты программной симуляции. В последнем разделе обсуждается применение предложенного подхода для решения задачи тестирования интернет-страниц.

2. Задача о многоруком бандите

2.1. Постановка задачи

Перейдем к формальному описанию стохастической задачи о многоруком бандите [13]. Пусть имеется набор из n действий a_1, \dots, a_n . Каждому действию a_i , $i = 1, \dots, n$ соответствует случайная величина $\xi_i \sim D_i$ с математическим ожиданием μ_i и дисперсией σ_i^2 . Вид распределения D_i , математическое ожидание μ_i и дисперсия σ_i^2 игроку неизвестны. В каждый момент времени $t = 1, \dots, T$ игрок выбирает действие $a_{j(t)}$ и получает выигрыш $r_{j(t)}$, который является реализацией соответствующей случайной величины $\xi_{j(t)} \sim D_{j(t)}$. Игрок преследует две цели: во-первых, определить действие, дающее наибольший средний выигрыш, и во-вторых, получить наибольший суммарный выигрыш за время T .

Для решения описанной задачи существуют алгоритмы (стратегии) [25], которые в каждый момент времени t определяют, какое действие следует выбрать. Распространенной мерой производительности данных алгоритмов является функция суммарного сожаления [15].

Определение 2.1. *Функцией суммарного сожаления называется функция, определяемая следующим образом:*

$$R(T) = \mu^*T - \sum_{t=1}^T \mu_{j(t)}, \quad (2.1)$$

где $\mu^* = \max_{1 \leq i \leq n} \mu_i$.

Определение 2.2. *Если для функции сожаления справедлива оценка $R(T) = O(\ln(T))$, то говорят, что алгоритм решает задачу о многоруком бандите. Иногда также такой алгоритм называют оптимальным.*

В дальнейшем будем рассматривать задачу о многоруком бандите, в которой случайные величины ξ_1, \dots, ξ_n имеют распределение Бернулли [2] с неизвестными параметрами p_1, \dots, p_n . Таким образом, выигрыш игрока в результате выбора действия равен 0 или 1.

Задача о многоруком бандите может быть рассмотрена как статистическая игра [1] или игра с природой. В таких играх одним из игроков является статистик (исследователь), другим – природа того явления, которое исследуется.

В нашем случае природа выбирает значения параметров p_1, \dots, p_n распределения случайных величин ξ_1, \dots, ξ_n , соответствующих выбираемым действиям a_1, \dots, a_n . Таким образом, каждый набор значений параметров p_1, \dots, p_n , удовлетворяющих условиям $p_i \in [0, 1]$, $i = 1, \dots, n$, является стратегией природы. Отметим, что природа как игрок не стремится к максимальному выигрышу (и, таким образом, не стремится минимизировать выигрыш игрока-исследователя).

Игрок же выбирает стратегию (правило, алгоритм), согласно которой происходит выбор действий a_1, \dots, a_n . В случае задачи о многоруком бандите стратегия игрока представляет собой функцию или алгоритм, результатом которых является одно из действий a_1, \dots, a_n . Выбранная стратегия применяется на всем горизонте игры. Стратегия игрока ориентирована на то, чтобы максимально точно «угадать» неизвестные для игрока значения параметров p_1, \dots, p_n и, вместе с тем, определить наибольший из них. Все известные авторам стратегии используют историю ранее сделанных выборов для выбора очередного действия. Определение некоторых стратегий можно найти,

например, в работе [25]. Далее мы подробно рассмотрим стратегию UCSB1.

Иногда мы будем использовать словосочетание *лучшее (оптимальное) действие*, понимая под этим действие, которому соответствует случайная величина с наибольшим математическим ожиданием. Слова *алгоритм* и *стратегия* будем использовать как синонимы.

2.2. Алгоритм UCSB1

Алгоритм UCSB1 предложен в работе [7]. На каждом шаге данный алгоритм выбирает действие с индексом:

$$j(t) = \arg \max_{i=1, \dots, n} \left(\bar{p}_i + \sqrt{\frac{2 \ln t}{n_i}} \right), \quad (2.2)$$

где n_i – количество раз, когда выбиралось действие a_i , а $\bar{p}_1(t), \dots, \bar{p}_n(t)$ – средние выигрыши действий a_1, \dots, a_n , соответственно, после t игр.

Очевидными преимуществами алгоритма являются отсутствие параметров и детерминированность. В [7] показано, что для функции сожаления $R(T)$ данного алгоритма справедлива оценка:

$$R(T) \leq 8 \sum_{i:p_i < p^*} \left(\frac{\ln T}{\Delta_i} \right) + \left(1 + \frac{\pi^2}{3} \right) \left(\sum_{j=1}^n \Delta_j \right),$$

где как и ранее $\Delta_i = p^* - p_i$. Таким образом, UCSB1 – оптимальный алгоритм.

Существуют различные вариации алгоритма UCSB1. Например, в работе [7] предложен алгоритм UCSB1-Normal для случая, когда случайные величины, соответствующие действиям a_1, \dots, a_n , распределены нормально.

В той же статье предлагается модификация алгоритма UCSB1, названная UCSB1-Tuned, которая, по мнению авторов, на практике может работать лучше, чем UCSB1, однако не имеет никаких теоретических гарантий. Главной особенностью алгоритма UCSB1-Tuned является учет выборочной дисперсии для выбора следующего действия.

3. Задача о многоруком бандите при наличии эксперта

3.1. Постановка задачи для случая одного эксперта

Рассмотрим следующую модификацию задачи о многоруком бандите [6]. Пусть имеется эксперт, который в каждый момент времени t делает предположение о значениях выигрышей действий a_1, \dots, a_n . Каждый раз, когда игрок выбирает действие, эксперт предлагает вектор $(b_1(t), \dots, b_n(t))$, в котором компонента $b_i(t)$ есть предположение о значении выигрыша в результате выбора действия a_i в момент времени t . Допустим, что эксперт владеет некоторой информацией о значениях выигрышей выбираемых действий, что позволяет ему с некоторой степенью точности «предсказывать» выигрыши в каждый момент времени.

Формализуем такие подсказки эксперта следующим образом. Пусть компоненты вектора $(b_1(t), \dots, b_n(t))$ являются реализациями случайных величин $\hat{\xi}_1, \dots, \hat{\xi}_n$, имеющих распределение Бернулли с неизвестными параметрами $\hat{p}_1, \dots, \hat{p}_n$. Здесь рассматривается распределение Бернулли, поскольку случайные величины, соответствующие действиям, имеют распределение Бернулли. Задача игрока состоит в том же, однако теперь он, помимо данных о значениях полученных выигрышей, может воспользоваться подсказками эксперта.

3.2. Модификация алгоритма UCS1

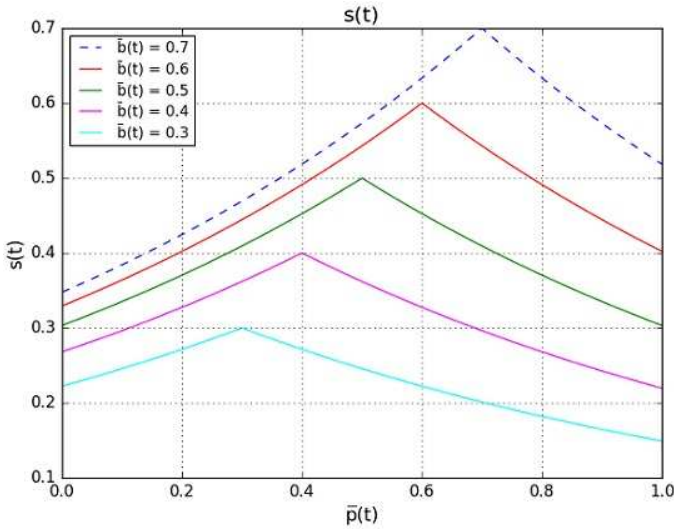
Модифицируем алгоритм UCS1 так, чтобы выбор игрока зависел от значения подсказки эксперта и от её точности. Для этого добавим в формулу (2.2) слагаемое $\bar{b}_i(t)k_i(t)$, где $\bar{b}_i(t)$ – среднее значение b_i за время t , т. е. средняя подсказка для действия a_i , а величину, характеризующую точность подсказки для действия a_i введем по следующей формуле:

$$k_i(t) = e^{-|\bar{b}_i(t) - \bar{p}_i(t)|}. \quad (3.1)$$

Множитель $k_i(t)$ (3.1) позволяет учитывать неточные подсказки с меньшим весом. Таким образом, модифицированный алгоритм UCS1 на каждом шаге выбирает действие с индексом:

$$j(t) = \arg \max_{i=1, \dots, n} \left(\bar{p}_i + \sqrt{\frac{2 \ln t}{n_i}} + \bar{b}_i(t)k_i(t) \right), \quad (3.2)$$

где $k_i(t)$ вычисляется по формуле (3.1).

Рисунок 1: $s(t)$ при фиксированных $\bar{b}(t)$

Проведем краткий анализ добавленного слагаемого $s(t) = \bar{b}(t)k(t)$. На рис. 1 показаны графики $s(t)$ при различных фиксированных значениях $\bar{b}(t)$. Нетрудно заметить, что максимум величины $s(t)$ при фиксированном значении средней подсказки $\bar{b}^*(t)$ достигается при $\bar{p}(t) = \bar{b}^*(t)$ и равен значению средней величины подсказки. Иначе говоря, максимум значения величины $s(t)$ при фиксированном значении $\bar{b}(t)$ достигается когда значение средней подсказки равно значению среднего выигрыша. Также видно, что $s(t)$ симметрично убывает относительно вершины $\bar{b}^*(t)$. Отсюда следует, что, например, при $\bar{b}(t) = 0.5$ значение $s(t)$ будет одинаковым для $\bar{p}(t) = 0.8$ и $\bar{p}(t) = 0.3$. Такой, казалось бы, противоречивый результат, не должен настораживать читателя, поскольку значение величины среднего выигрыша является первым слагаемым в формуле (3.2), и, таким образом, при прочих равных условиях, будет выбрано действие с $\bar{p}(t) = 0.8$.

На рис. 2 продемонстрированы графики функции $s(t)$ при различных фиксированных значениях $\bar{p}(t)$. Видно, каким образом возрастает $s(t)$ при фиксированном $\bar{p}(t)$: быстрый рост до значения $\bar{b}(t) = \bar{p}(t)$, затем медленное возрастание до $\bar{b}(t) = 1$.

Заметим, что предложенная в настоящей работе модификация алгоритма UCS1 не является оптимальной для любых значений $\hat{p}_1, \dots, \hat{p}_n$. Об этом, в частности, свидетельствуют результаты численных экспе-

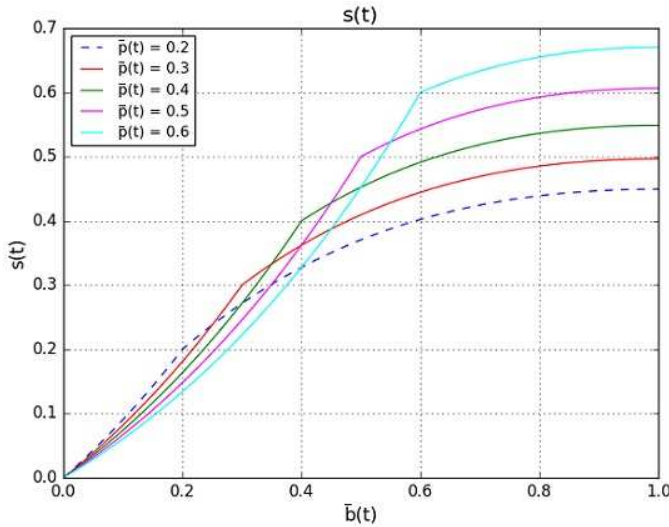


Рисунок 2: $s(t)$ при фиксированных $\bar{p}(t)$

риментов, представленных ниже. Одним из направлений дальнейшего исследования является вывод условий оптимальности предложенного алгоритма.

3.3. Численный эксперимент (один эксперт)

Для анализа результатов работы модифицированного алгоритма УСВ1 была выполнена его программная реализация на языке Python. На вход программе подается число действий n , вектор математических ожиданий случайных величин, соответствующих выбираемым действиям (p_1, \dots, p_n) , вектор математических ожиданий случайных величин, соответствующих подсказкам эксперта $(\hat{p}_1, \dots, \hat{p}_n)$ и число итераций T .

Процесс симуляции моделируется методом Монте-Карло. Одна итерация алгоритма выполняется следующим образом.

Согласно величинам математических ожиданий с помощью встроенных функций языка Python генерируется вектор подсказок эксперта $(b_1(t), \dots, b_n(t))$. Затем алгоритм выбирает одно из n действий по формуле (3.2). Далее генерируется выигрыш согласно математическому ожиданию случайной величины, соответствующей выбранному действию. Наконец, согласно формуле (2.1) вычисляется значение функции сожаления.

Таким образом в результате T итераций программы получаем данные для построения графика функции сожаления. Для получения гладкого графика функции $R(t)$ делается несколько прогонов программы на одних входных данных, и в качестве результирующих данных для построения графика функции сожаления берутся средние значения $R(t)$ на каждой итерации.

В результате описанной симуляции на выход программы выдается, в том числе, график функции сожаления.

Напомним, что функция сожаления и величина выигрыша связаны обратным соотношением: чем меньше значение функции сожаления, тем больше значение выигрыша. Поэтому для сравнения полученных выигрышей достаточно привести графики функций сожаления.

Рассмотрим пример входных данных. Пусть $n = 5, T = 5000$ и

$$(p_1, p_2, p_3, p_4, p_5) = (0.3, 0.45, 0.5, 0.47, 0.1).$$

Для сравнения приведем графики функции сожаления алгоритма UCSV1 без учета эксперта и функции сожаления модифицированного алгоритма со следующими значениями математических ожиданий подсказок эксперта $(\hat{p}_1, \dots, \hat{p}_n)$:

$$e_1 = (0.5, 0.45, 0.1, 0.5, 0.7), \quad e_2 = (0.1, 0.1, 0.6, 0.1, 0.1),$$

$$e_3 = (0.3, 0.45, 0.5, 0.47, 0.1), \quad e_4 = (0.1, 0.1, 0.1, 0.1, 0.1),$$

$$e_5 = (0.2, 0.3, 0.45, 0.32, 0.05).$$

Результаты работы программы в виде графиков функций сожаления представлены на рис. 3. В подписи к графикам указаны векторы математических ожиданий подсказок экспертов. Подпись «UCSV1» соответствует функции сожаления алгоритма UCSV1 без эксперта. Как видно из рисунка, медленнее всех возрастает функция сожаления, соответствующая вектору математических ожиданий подсказок эксперта e_2 . Данное поведение алгоритма объясняется тем, что третья компонента, которая и соответствует действию с наибольшим средним выигрышем, намного больше других. Таким образом эксперт уверен, что третье действие приносит наибольший выигрыш.

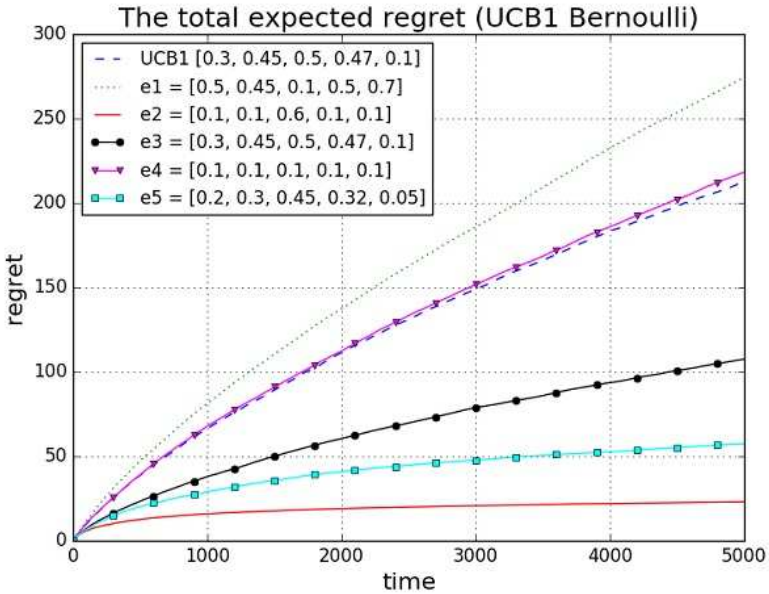


Рисунок 3: Функции сожаления ($n = 5, m = 1$)

Максимум функции сожаления достигается для вектора e_1 , в котором, наоборот, компоненты, соответствующие неоптимальным действиям, завышены. Второй по величине является функция сожаления, соответствующая вектору e_4 , в котором все компоненты равны 0.1. Важно отметить, что подсказки эксперта с высокой степенью точности (вектор e_3) способствуют увеличению выигрыша.

4. Задача о многоруком бандите с m экспертами

4.1. Постановка задачи для случая m экспертов

Рассмотрим задачу о многоруком бандите с m экспертами. Аналогично постановке задачи в п. 3.1 каждый эксперт $i, i = 1, \dots, m$ в момент времени t предлагает вектор подсказок $(b_{i1}(t), \dots, b_{in}(t))$. Таким образом, можно сформировать матрицу подсказок в момент времени t

$$B(t) = \begin{pmatrix} b_{11}(t) & b_{12}(t) & \dots & b_{1n}(t) \\ b_{21}(t) & b_{22}(t) & \dots & b_{2n}(t) \\ \dots & \dots & \dots & \dots \\ b_{m1}(t) & b_{m2}(t) & \dots & b_{mn}(t) \end{pmatrix},$$

где строка $b_i(t) = (b_{i1}(t), \dots, b_{in}(t))$ есть вектор подсказок i -го эксперта в момент времени t . Как и ранее, подсказка $b_{ij}(t)$ есть реализация распределенной по Бернулли случайной величины $\widehat{\xi}_{ij}$. Задача для игрока остается прежней, однако теперь он может принимать решения, основываясь на подсказках каждого из m экспертов. Матрицу средних подсказок в момент времени t будем обозначать через $\overline{B}(t)$.

Решим задачу следующим способом: будем каждый раз каким-либо образом извлекать из матрицы подсказок вектор и применять уже модифицированный алгоритм UCS1. Рассмотрим различные способы формирования вектора из матрицы подсказок экспертов.

Как уже было замечено выше, точные подсказки эксперта должны увеличивать выигрыш. На этом замечании построены первые два метода формирования вектора подсказок.

Метод 1. Каждый раз будем выбирать вектор подсказок эксперта, для которого достигается минимум евклидовой метрики:

$$\rho(\overline{b}_i(t), \overline{p}(t)) = \sqrt{\sum_{j=1}^n (\overline{b}_{ij}(t) - \overline{p}_j(t))^2},$$

где $\overline{b}_i(t) = (\overline{b}_{i1}(t), \dots, \overline{b}_{in}(t))$ есть средний вектор подсказок i -го эксперта за время t ; $\overline{p}(t) = (\overline{p}_1(t), \dots, \overline{p}_n(t))$ – вектор средних выигрышей за время t .

Метод 2. В каждом столбце матрицы средних подсказок экспертов $\overline{B}(t)$ будем выбирать компоненту, для которой достигается минимум:

$$\widetilde{b}_j(t) = \min_{i=1, \dots, m} |\overline{b}_{ij}(t) - \overline{p}_j(t)|, j = 1, \dots, n.$$

Метод 3. Здесь в каждый момент времени t будем определять лучшего эксперта и рассматривать его вектор подсказок. Под лучшим экспертом будем понимать эксперта, чьи подсказки приносят наибольший выигрыш. Поскольку выбор эксперта влияет на выигрыш игрока, а его подсказки являются случайными величинами, то поиск наилучшего эксперта можно рассмотреть как еще одну задачу о многоруком бандите. Для решения этой задачи можно воспользоваться любым алгоритмом независимо от того, какой алгоритм используется для выбора действия. Пусть, например, для выбора эксперта используется все тот же алгоритм UCS1. Тогда игрок в каж-

дый момент времени t выбирает эксперта с номером $j(t)$, где

$$j(t) = \arg \max_{i=1, \dots, m} \left(\bar{r}_i + \sqrt{\frac{2 \ln t}{m_i}} \right),$$

где \bar{r}_i – средний выигрыш в результате выбора i -го эксперта, а m_i – число раз, когда выбирался эксперт с номером i .

4.2. Численный эксперимент (m экспертов)

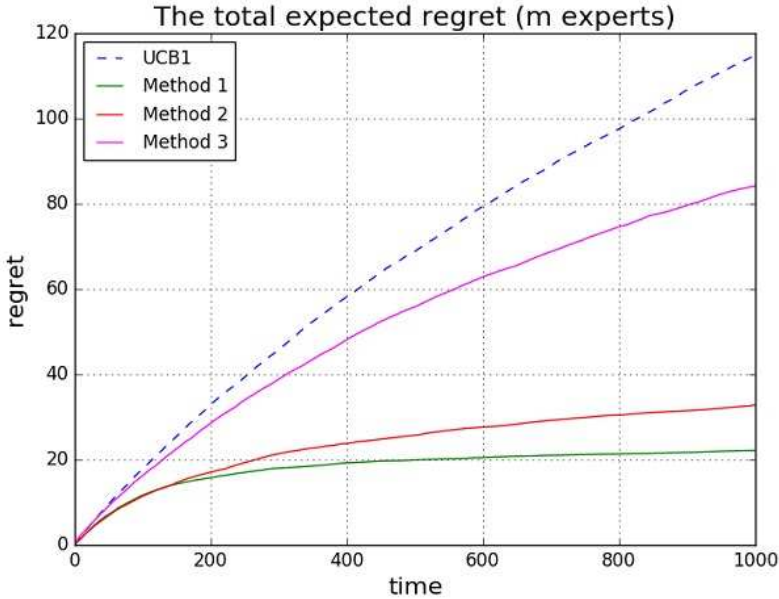
Для проведения эксперимента была модифицирована программа для случая задачи с одним экспертом.

На вход программе подается число действий n , вектор математических ожиданий выбираемых действий (p_1, \dots, p_n) , матрица \hat{P} размерности $m \times n$ математических ожиданий случайных величин, соответствующих подсказкам экспертов, и число игр T . В результате программной симуляции, подробно описанной в п. 3.3, были построены графики функции сожаления.

Пусть $n = 5, p = (0.3, 0.45, 0.6, 0.4, 0.1), T = 1000$ и матрица математических ожиданий подсказок экспертов имеет вид

$$\hat{P} = \begin{pmatrix} \mathbf{0.3} & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & \mathbf{0.45} & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & \mathbf{0.6} & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & \mathbf{0.4} & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & \mathbf{0.1} \end{pmatrix}$$

На рис. 4 продемонстрированы графики функции сожаления каждого из трех методов формирования вектора подсказок. Таким образом, в данном случае наименьшее значение функции сожаления соответствует методу 1, а наибольшее – методу 3. В таблице 1 показано, сколько раз выбирался каждый эксперт с помощью первого и третьего метода (второй метод эксперта не выбирает). При использовании метода 1 подавляющее число раз выбирался третий эксперт, хотя на первый взгляд кажется, что каждый эксперт имеет одинаковую вероятность быть выбранным. Такое поведение алгоритма объясняется тем, что третье действие (с математическим ожиданием 0.6) выбирается чаще всего, поэтому и средний выигрыш его близок к значению 0.6 (см. закон больших чисел [2]). Средние подсказки экспертов обновляются каждый раз, поэтому их значения одинаково сходятся к

Рисунок 4: Функции сожаления ($n = 5, m = 5$)

значениям математических ожиданий. А поскольку математические ожидания на диагонали матрицы \hat{P} совпадают с элементами вектора p и все остальные значения матрицы равны 0.1, то, принимая во внимание вышесказанные рассуждения, можно заключить, что евклидова метрика с наибольшей вероятностью будет минимальна для третьего эксперта.

Каждый элемент матрицы N равен числу выборов подсказки в методе 2. Таким образом, как это и следует из вида входных данных, подсказки, стоящие на диагонали, выбирались чаще всего (за исключением последнего столбца).

$$N = \begin{pmatrix} \mathbf{652} & 48 & 17 & 52 & 192 \\ 89 & \mathbf{816} & 13 & 49 & 179 \\ 91 & 49 & \mathbf{964} & 60 & 194 \\ 118 & 36 & 4 & \mathbf{806} & 224 \\ 50 & 51 & 2 & 33 & 211 \end{pmatrix}$$

Результаты численных экспериментов соответствуют интуитивным представлениям о влиянии подсказок эксперта и подтверждают релевантность предложенного подхода. Точные подсказки способ-

Таблица 1: Число выборов экспертов ($n = 5, m = 5$)

Метод \ Эксперт	1	2	3	4	5
Метод 1	7	30	934	27	2
Метод 3	68	152	619	111	50

ствуют увеличению выигрыша. В большинстве запусков методы 1 и 2 давали результат лучше третьего. В дальнейшем предполагается провести анализ поведения функции сожаления (2.2) модифицированного алгоритма.

5. Практическое применение

Задача о многоруком бандите при наличии экспертов может быть применена для актуальной задачи тестирования интернет-страниц.

Тестирование интернет-страниц – один из способов увеличения конверсии сайта, что, в свою очередь, является ключевой задачей интернет-маркетинга. Под конверсией сайта понимается отношение количества посетителей сайта, выполнивших необходимое действие (покупка товара, заказ обратного звонка, подписка на рассылку и т.д.), к общему числу посетителей.

Тестирование интернет-страниц с целью увеличения конверсии может быть формализовано в виде классической задачи о многоруком бандите следующим образом [4,5]. Пусть набор действий a_1, \dots, a_n соответствует тестируемым страницам. Выбор действия a_i означает показ страницы с номером i . Достижение конверсионной цели при показе страницы соответствует получению выигрыша, равного 1. Если при показе страницы цель не достигнута, выигрыш равен 0. Таким образом, тестирование страниц свелось к задаче о многоруком бандите, в которой случайные величины, соответствующие действиям, имеют распределение Бернулли с математическим ожиданием, равным конверсиям страниц.

Предложенный в данной работе подход о модификации задачи о многоруком бандите также может быть использован для тестирования интернет-страниц. Экспертную подсказку, например, можно построить на основе средней продолжительности визита пользователя. Как правило, при прочих равных условиях, высокое значение средней продолжительности визита свидетельствует о высокой кон-

версии страницы. Следовательно, подсказки можно генерировать на основе средней продолжительности визита: чем выше значение, тем с большей вероятностью прогнозировать достижение конверсионной цели.

Таким образом, предложенная в статье модель принятия решений может быть использована для актуальных практических приложений.

Авторы благодарят анонимного рецензента за высококвалифицированную рецензию.

СПИСОК ЛИТЕРАТУРЫ

1. Боровков А.А. *Математическая статистика: дополнительные главы*. М: Наука, 1984
2. Буре В.М., Парилина Е.М. *Теория вероятностей и математическая статистика*. М.: Лань, 2013. 416 с.
3. Лазутченко А.Н. *О робастном управлении в случайной среде, характеризуемой нормальным распределением доходов с различными дисперсиями* // Труды Карельского научного центра Российской академии наук. 2015. №. 10.
4. Смирнов Д.С. *Тестирование интернет-страниц как решение задачи о многоруком бандите* // Молодой ученый. 2015. № 19. С. 78–86.
5. Смирнов Д.С. *Использование задачи о многоруком бандите в тестировании веб-страниц* // Процессы управления и устойчивость. 2016. Т. 3. № 1. С. 705–710.
6. Смирнов Д.С. *Задача о многоруком бандите при наличии эксперта* // Процессы управления и устойчивость. 2017. Т. 4(20). № 1. С. 681–685.
7. Auer P., Cesa-Bianchi N., Fischer P. *Finite-time Analysis of the Multiarmed Bandit Problem* // Machine Learning. 2002. Vol. 47, No. 2-3. P. 235–256.

8. Auer P. et al. *The nonstochastic multiarmed bandit problem* // SIAM journal on computing. 2002. Vol. 32. No. 1. С. 48–77.
9. Awerbuch B., Kleinberg R. *Online linear optimization and adaptive routing* // Journal of Computer and System Sciences. 2008. Vol. 74. No. 1. С. 97–114.
10. Bather J.A. *The Minimax Risk for the Two-Armed Bandit Problem* // Mathematical Learning Models – Theory and Algorithms. Lecture Notes in Statistics. Vol. 20, P. 1–11. Springer-Verlag, New York Inc. 1983.
11. Chu W. et al. *Contextual Bandits with Linear Payoff Functions* // AISTATS. 2011. Vol. 15. P. 208–214.
12. Hardwick J. et al. *Bandit strategies for ethical sequential allocation* // Computing Science and Statistics. 1991. Vol. 23. No. 6.1. P. 421–424.
13. Kuleshov V., Precup D. *Algorithms for the multi-armed bandit problem* // Journal of Machine Learning Research. 2000. P. 1–48.
14. Lage R. et al. *Choosing which message to publish on social networks: A contextual bandit approach* // Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on. IEEE, 2013. P. 620–627.
15. Lai T.L., Robbins H. *Asymptotically efficient adaptive allocation rules* // Advances in applied mathematics. 1985. No. 6. P. 4–22.
16. Lai T.Z. *Adaptive treatment allocation and the multi-armed bandit problem* // The annals of statistics. 1987. Vol. 15. P. 1091–1114.
17. Langford J., Zhang T. *The epoch-greedy algorithm for multi-armed bandits with side information* // Advances in neural information processing systems. 2008. P. 817–824.
18. Li L. et al. *A contextual-bandit approach to personalized news article recommendation* // Proceedings of the 19th international conference on World wide web. ACM. 2010. P. 661–670.

19. Lu T., Pal D., Pal M. *Contextual Multi-Armed Bandits* // AISTATS. 2010. P. 485–492.
20. Pandey S., Olston C. *Handling advertisements of unknown quality in search advertising* // NIPS. 2006. Vol. 20. P. 1065–1072.
21. Robbins H. *Some aspects of the sequential design of experiments* // Herbert Robbins Selected Papers. Springer New York, 1985. P. 169–177.
22. Schwartz E.M., Misra K., Abernethy J. *Dynamic Online Pricing with Incomplete Information Using Multi-Armed Bandit Experiments*. 2016.
23. Shen W. et al. *Portfolio Choices with Orthogonal Bandit Learning* // IJCAI. 2015. P. 974–980.
24. Strehl A.L. et al. *Experience-efficient learning in associative bandit problems* // Proceedings of the 23rd international conference on Machine learning. ACM, 2006. P. 889–896.
25. Sutton R.S., Barto A.G. *Reinforcement learning: An introduction*. Cambridge : MIT press, 1998. Vol. 1. No. 1.
26. Thompson W.R. *On the likelihood that one unknown probability exceeds another in view of the evidence of two samples* // Biometrika. 1933. Vol. 25. No. 3/4. P. 285–294.
27. Woodroffe M. *A one-armed bandit problem with a concomitant variable* // Journal of the American Statistical Association. 1979. Vol. 74. No. 368. P. 799–806.

DECISION-MAKING MODEL UNDER PRESENCE OF EXPERTS AS A MODIFIED MULTI-ARMED BANDIT PROBLEM

Dmitriy S. Smirnov, Saint-Petersburg State University, Faculty of Applied Mathematics and Control Process, postgraduate student (st016315@student.spbu.ru).

Ekaterina V. Gromova, Saint-Petersburg State University, Faculty of Applied Mathematics and Control Process, Cand.Sc. (e.v.gromova@spbu.ru)

Abstract: The modified multi-armed bandit problem is formulated in the paper which allows the player to use so-called expert hints in the decision making process. As a player in this problem is meant some automated system that uses a certain strategy (algorithm) for making a decision under conditions of uncertainty. The approach is developed for the case of m experts. A modification of the well-known UCB1 algorithm is proposed to solve the multi-armed bandit problem. The results of a numerical experiment are given in order to show influence of expert hints on the player's payoff.

Keywords: multi-armed bandit problem, decision making, optimization methods, machine learning algorithms.