Московский Государственный Университет
имени М.В. Ломоносова
Российская Академия Наук
Международная Академия Технологических Наук
Российская Академия Естественных Наук

Интеллектуальные Системы.

Теория и приложения

ТОМ **28** ВЫПУСК 3 * **2024**

МОСКВА

УДК 519.95; 007:159.955 ISSN 2411–4448
ББК 32.81 Издаётся с 1996 г.

В журнале «Интеллектуальные системы. Теория и приложения» публикуются научные достижения в области теории и приложений интеллектуальных систем, новых информационных технологий и компьютерных наук.
Издание журнала осуществляется под эгидой МГУ имени М.В. Ломоносова, Научного Совета по комплексной проблеме «Кибернетика» РАН, Отделения «Математическое моделирование технологических процессов» МАТН.
Учредитель журнала: ООО «Интеллектуальные системы».
Журнал входит в список изданий, включенных ВАК РФ в реестр публикаций материалов по кандидатским и докторским диссертациям по математике и механике.

Индекс подписки на журнал: 64559 в каталоге НТИ «Роспечать».

Адрес редакции: 119991, Москва, ГСП-1, Ленинские Горы, д. 1, механико-математический факультет, комн. 12-01.
Адрес издателя: 115230, Россия, Москва, Хлебозаводский проезд, д. 7, стр. 9, офис 9. Тел. +7 (495) 939-46-37, e-mail: mail@intsysjournal.org

*) Прежнее название журнала: «Интеллектуальные системы».

ОГЛАВЛЕНИЕ

Часть 1
Общие проблемы теории интеллектуальных систем

Точность алгоритмов сингулярного разложения матриц с различным спектром

И. Ю. Дроздов[1] Д. В. Парфенов[2]

Развивается наш новый подход по рассмотрению сингулярного спектра как функции плотности распределения с целью углубленного анализа зависимости погрешности нахождения сингулярных значений от спектра. Проведены масштабные численные эксперименты, выявляющие данные зависимости в новых предложенных метриках: среднеквадратичная относительная погрешность и медиана. Построены иллюстрирующие зависимости графики и оценена информативность предложенных метрик.

Ключевые слова: сингулярное разложение, число обусловленности, спектр матрицы, численная устойчивость

1. Введение

Сингулярное разложение (SVD) – факторизация матриц общего вида, применяющаяся для решения широкого круга практических задач, обычно в численном виде. Алгоритмы сингулярного разложения реализованы во множестве программных пакетов линейной алгебры. В [1] нами продемонстрирована зависимость точности нахождения сингулярных значений не только от общепринятого критерия – числа обусловленности матрицы, но и от особенностей распределения сингулярных значений в спектре матрицы. Там же отмечена недостаточность широко используемых в линейной алгебре метрик для оценки точности нахождения сингулярных значений в целом. Данное исследование посвящено анализу конкретных зависимостей между различными распределениями сингулярных значений и точностью их нахождения в разных метриках оценки этой точности.

[1] *Дроздов Игорь Юрьевич* — старший преподаватель каф. высшей математики Института искусственного интеллекта РТУ МИРЭА, e-mail: drozdov_i@mirea.ru.

Drozdov Igor Yurievich — senior lecturer, Russian Technological University (MIREA), Institute of Artificial Intelligence, Department of Higher Mathematics.

[2] *Парфенов Денис Васильевич* — к.т.н., доцент каф. высшей математики Института искусственного интеллекта РТУ МИРЭА, e-mail: parfenov@mirea.ru.

Parfenov Denis Vasilevich, Ph.D. — associate professor, Russian Technological University (MIREA), Institute of Artificial Intelligence, Department of Higher Mathematics.

2. Алгоритм, используемые в распространенных реализациях SVD

Сингулярное разложение определяется как

$$A = U\Sigma V^T, \tag{1}$$

где A – исходная матрица размером $m \times n$, U и V – унитарные матрицы размерами $m \times m$ и $n \times n$ соответственно, Σ – диагональная матрица размером $\min(m,n) \times \min(m,n)$ с сингулярными значениями на главной диагонали. У матрицы A размером $m \times n$ существует $\min(m,n)$ сингулярных значений.

В нашей предыдущей статье [1] для проведения численных экспериментов выделены три реализации полного сингулярного разложения из двух наиболее распространенных библиотек линейной алгебры: методы `dgesvd` и `dgesdd` из библиотеки LAPACK [2] и `BDCSVD` из библиотеки Eigen [3]. Все эти алгоритмы характеризуются общей стратегией, но различаются ее непосредственной реализацией, заключающейся в следующем:

1) Матрица A общего вида представляется как произведение $U_1 B V_1^T$, где U_1 и V_1 – ортогональные матрицы, B – бидиагональная матрица, в ней ненулевые элементы расположены на главной диагонали и либо на нижней поддиагонали (если $m < n$), либо на верхней наддиагонали (если $m \geq n$);

2) Находится сингулярное разложение бидиагональной матрицы: $B = U_2 \Sigma V_2^T$. Тогда значения на главной диагонали матрицы Σ являются сингулярными значениями исходной матрицы A, а матрицы из сингулярных векторов получаются как $U = U_1 U_2,\ V = V_1 V_2$.

Далее опишем два основных подхода для нахождения сингулярного разложения бидиагональной матрицы B, используемые в LAPACK. Их принципиальное отличие заключается в следующем: `dgesvd` применяет итеративную схему, основанную на реализации неявного QR-алгоритма с нулевым сдвигом [4] к матрице B целиком, в то время как `dgesdd` использует рекурсивное разбиение [5] матрицы B на блоки достаточного малого размера, вычисляет сингулярное разложение этих малых блоков методом `dgesvd`, а затем в обратном порядке реконструирует сингулярное разложение исходной матрицы B.

2.1. dgesvd из LAPACK

Если матрица $m \times n$ достаточно "высокая" или "широкая" ($m \gg n$ или $n \gg m$), в целях повышения производительности вначале выполняется QR

или LQ разложение соответственно. В этом случае алгоритм продолжает работу с матрицами R размерности $n \times n$ или L размерности $m \times m$ вместо исходной A. Если количество строк или столбцов исходной матрицы A сопоставимы, эти предварительные разложения пропускаются. Далее матрица представляется в бидиагональном виде алгоритмом, известным как бидиагонализация Голуба-Кахана [7], использующего преобразования Хаусхолдера. Сингулярное разложение бидиагональной матрицы B находится с помощью алгоритма, предложенного в [4]. Проиллюстрируем его основную идею на примере матрицы 4×4.

Обозначим $G(\theta, i, j)$ матрицу поворота Гивенса по координатам i и j на задающий параметризовацию угол θ, ненулевые элементы которой задаются как

$$G(\theta, i, j) = \begin{cases} g_{kk} = 1 & \text{для } k \neq i, j, \\ g_{kk} = \cos\theta & \text{для } k = i, j, \\ g_{ji} = -gij = -\sin\theta & . \end{cases}$$

Бидиагональная матрица $B_i^{(0)}$ выглядит следующим образом:

$$B_i^{(0)} = \begin{bmatrix} b_{11}^{(0)} & b_{12}^{(0)} & & \\ & b_{22}^{(0)} & b_{23}^{(0)} & \\ & & b_{33}^{(0)} & b_{34}^{(0)} \\ & & & b_{44}^{(0)} \end{bmatrix}.$$

Выбирается угол θ_1, такой что $\operatorname{tg}\theta_1 = -b_{12}/b_{11}$. Матрица B умножается на матрицу $G_1(2, 1, \theta_1)$:

$$B_i^{(1)} = B_i^{(0)} G_1 = \begin{bmatrix} b_{11}^{(1)} & 0 & & \\ b_{21}^{(1)} & b_{22}^{(1)} & b_{23}^{(1)} & \\ & & b_{33}^{(1)} & b_{34}^{(1)} \\ & & & b_{44}^{(1)} \end{bmatrix}.$$

Далее выбирается такой угол θ_2, чтобы умножение на $G_2(1, 2, \theta_2)$ слева позволило получить 0 в элементе b_{21}:

$$B_i^{(2)} = G_2 B_i^{(1)} = \begin{bmatrix} b_{11}^{(2)} & b_{12}^{(2)} & b_{13}^{(2)} & \\ 0 & b_{22}^{(2)} & b_{23}^{(2)} & \\ & & b_{33}^{(2)} & b_{34}^{(2)} \\ & & & b_{44}^{(2)} \end{bmatrix}.$$

Алгоритм продолжается подобным образом, выбирая θ_i так, чтобы обнулить единственный ненулевой элемент, лежащий вне главной и

наддиагонали. Обнуление этого элемента с помощью поворота Гивенса приводит к появлению другого ненулевого элемента по другой сторону от ненулевой полосы матрицы и ближе к ее правому краю. Говоря неформально, алгоритм "гоняется" за ненулевым элементов вне полосы, смещая его за край матрицы. В конечном счете, для матрицы 4×4, имеем следующее:

$$B_i^{(7)} = G_6 G_4 G_2 B_i^{(0)} G_1 G_3 G_5 = \begin{bmatrix} b_{11}^{(7)} & b_{12}^{(7)} & & \\ & b_{22}^{(7)} & b_{23}^{(7)} & \\ & & b_{33}^{(7)} & b_{34}^{(7)} \\ & & & b_{44}^{(7)} \end{bmatrix},$$

что является снова бидиагональной матрицей. Эта матрица становится начальной для следующей итерации алгоритма: $B_{i+1}^{(0)} = B_i^{(7)}$. Повторение таких итераций сводит матрицу к диагональному виду. Авторы алгоритма предлагают адаптивное использование нескольких критериев сходимости в [4].

После достижения критерия сходимости бидиагональная матрица B сведена к диагональной Σ, содержащей сингулярные значения. Сингулярные векторы матрицы B в матрицах U_2 и V_2^T получаются в виде произведения всех матриц G, накопленных в ходе алгоритма слева и справа соответственно.

2.2. dgesdd из LAPACK

Подобно методу `dgesvd`, метод `dgesdd` также может провести QR или LQ разложение и выполняет бидиагонализацию Голуба-Кахана. Далее производится сингулярное разложение бидиагональной матрицы с помощью рекурсивного алгоритма "разделяй и властвуй" [5].

Бидиагональная матрица B размера $N \times N + 1$ представляется в блочном виде:

$$B = \begin{bmatrix} B_1 & \alpha_k e_k & 0 \\ 0 & \beta_k e_1 & B_2 \end{bmatrix},$$

где B_1 и B_2 – бидиагональные блоки размера $k \times k - 1$ и $N - k + 1 \times N - k$ соответственно, e_k – вектор из канонического базиса с единицей в k-м элементе, k обычно выбирается как $N/2$. Идея алгоритма заключается в нахождении сингулярного разложения B_1 и B_2, из которых в дальнейшем получается разложение B. Алгоритм применяется рекурсивно к блокам матрицы, пока отдельные блоки не станут достаточно малого размера, сингулярное разложение которых находится другим методом. В `dgesdd` LAPACK для этого применяется алгоритм, аналогичный используемому в `dgesvd`, описанному выше.

Основная сложность заключается в нахождении сингулярного разложения B, имея сингулярные разложения B_1 и B_2. Представим сингулярное разложение B_i как

$$B_i = \begin{bmatrix} Q_i & q_i \end{bmatrix} \begin{bmatrix} D_i \\ 0 \end{bmatrix} V_i^T. \tag{2}$$

Пусть l_1^T и λ_1 - последняя строка и элемент Q_1 и q_1 соответственно. Также пусть f_1^T и ϕ_1 - первая строка и элемент Q_2 и q_2 соответственно. Тогда (2) можно представить в следующем виде:

$$B = \begin{bmatrix} Q_1 & q_1 & 0 & 0 \\ 0 & 0 & Q_2 & Q_1 \end{bmatrix} \begin{bmatrix} \alpha_k\lambda_1 & 0 & 0 \\ \alpha_k l_1 & D_1 & 0 \\ \beta_k f_2 & 0 & D_2 \\ \beta_k\phi_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & V_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & V_2 \end{bmatrix}.$$

Применяя поворот Гивенса с целью получения 0 на месте элемента $\beta_k\phi_2$, имеем

$$\begin{aligned} B &= \begin{bmatrix} c_0 q_1 & Q_1 & 0 & -s_0 q_1 \\ s_0 q_2 & 0 & Q_2 & c_0 Q_2 \end{bmatrix} \begin{bmatrix} r_0 & 0 & 0 \\ \alpha_k l_1 & D_1 & 0 \\ \beta_k f_2 & 0 & D_2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & V_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & V_2 \end{bmatrix} \\ &= \begin{bmatrix} Q & q \end{bmatrix} \begin{bmatrix} M \\ 0 \end{bmatrix} V_i^T, \end{aligned}$$

где $r_0 = \sqrt{(\alpha_k\lambda_1)^2 + (\beta_k\phi_2)^2}$, $c_0 = \alpha_k\lambda_1/r_0$, $s_0 = \beta_k\phi_2/r_0$. Далее проблема сводится к нахождению сингулярного разложения матрицы M вида

$$M = \begin{bmatrix} z_1 & & & \\ z_2 & d_2 & & \\ \vdots & & \ddots & \\ z_n & & & d_n \end{bmatrix}.$$

Относительно этой матрицы применяются следующие предположения и обозначения: $0 \equiv d_1 \leq d_2 \leq \ldots \leq d_n$, $z = (z_1 z_2 \ldots z_n)^T$. Также предполагается, что

$$d_{j+1} - d_j \geq \tau\|M\|_2 \text{ и } |z_j| \geq \tau\|M\|_2, \tag{3}$$

где τ – малое число, кратное машинной точности. Авторы алгоритма предлагают процедуру, обеспечивающую удовлетворение этих условий для любой матрицы такого вида. В [6] показано, что сингулярные значения σ_i матрицы M удовлетворяют условию

$$0 \equiv d_1 < \sigma_1 < d_2 < \ldots < d_n < \sigma_n < d_n + \|z\|_2$$

и характеристическому уравнению

$$f(\sigma) = 1 + \sum_{k=1}^{n} \frac{z_k^2}{d_k^2 - \sigma^2} = 0.$$

Собственные векторы находятся как

$$u_i = (\frac{z_1}{d_1^2 - \sigma_i^2}, \cdots, \frac{z_n}{d_n^2 - \sigma_i^2})^T / \sqrt{\sum_{k=1}^{n} \frac{z_k^2}{(d_k^2 - \sigma_i^2)^2}},$$

$$v_i = (-1, \frac{d_2 z_2}{d_1^2 - \sigma_i^2}, \cdots, \frac{d_n z_n}{d_n^2 - \sigma_i^2})^T / \sqrt{1 + \sum_{k=2}^{n} \frac{(d_k z_k)^2}{(d_k^2 - \sigma_i^2)^2}}.$$

Таким образом, решение характеристического уравнения численными методами позволяет найти сингулярные значения, далее находятся сингулярные векторы. Для повышения численной устойчивости применяются соображения, подробно описываемые авторами в [5].

2.3. BDCSVD из Eigen

Данный метод применяет тот же подход "разделяй и властвуй", приведенный в [5] и используемый методом `dgesdd` из LAPACK (раздел 2.2. Основное различие между `BDSVD` из Eigen и `dgesdd` из LAPACK заключается в способе вычисления сингулярного разложения матриц достаточного малого размера в конце рекурсии: `BDSVD` из Eigen использует метод Якоби [7], в то время как `dgesdd` из LAPACK применяет метод `dgesvd`. Также `BDSVD` принимает малые сингулярные значения (которые, как правило, находятся неточно) равными нулю, что повышает его общую точность в выбранных нами метриках.

3. Цели и методы исследования

Целью исследования является изучение зависимости точности нахождения сингулярных значений от особенностей распределения этих значений в спектре матрицы. Мы используем матрицы размером 3000×2000 с фиксированным числом обусловленности 10^{20} и крайними сингулярными значениями $\sigma_{\min} = 10^{-10}$ и $\sigma_{\max} = 10^{10}$. Остальные сингулярные значения будем называть внутренними. Их различные распределения моделируются с помощью параметризуемой прямоугольной функции

вида:

$$\mathrm{rect}_\sigma(x;\texttt{center},\texttt{width})=\begin{cases}0, & x<\texttt{center}-\frac{\texttt{width}}{2},\\ \frac{1}{\texttt{width}}, & \texttt{center}-\frac{\texttt{width}}{2}\leq x\leq\texttt{center}+\frac{\texttt{width}}{2},\\ 0, & x>\texttt{center}-\frac{\texttt{width}}{2}.\end{cases} \tag{4}$$

Параметр $0 < \texttt{center} < 1$ задает центр прямоугольника, `width` устанавливает его общую ширину, $\texttt{center} - \texttt{width}/2 > 0$, $\texttt{center} + \texttt{width}/2 < 1$. Таким образом, все внутренние сингулярные значения отвечают интервалу $(0, 1)$. Для любых `center` и `width` наименьшее сингулярное значение $\sigma_{\min}$ соотвествует $x = 0$, а наибольшее $\sigma_{\max}$ отвечает $x = 1$. Генерация набора тестов заключалась, неформально, в следующем: выбираются 10 значений ширины прямоугольника `width`: 0.05 и от 0.1 до 0.9 с равным шагом. Независимо от этого для каждой фиксированной ширины прямоугольника равномерно фиксируются 10 положений `center`: от левого края $(0, 1)$ до правого. Таким образом, получается сетка из 100 пар параметров $(\texttt{center}, \texttt{width})$, соответствующих различным прямоугольникам с различными положениями на интервале $(0, 1)$ и различной шириной.

Функция $\mathrm{rect}_\sigma(x; \texttt{center}, \texttt{width})$ для каждой пары (`center`, `width`) понимается как функция равномерного распределения сингулярных значений внутри интервала, то есть гистограмма наборов внутренних сингулярных значений выглядит как функция $\mathrm{rect}_\sigma(x; \texttt{center}, \texttt{width})$. Крайние и внутренние распределенные на отрезке $[0, 1]$ сингулярные значения далее проецируются на логарифмическую шкалу в отрезке от $[10^{-10}, 10^{10}]$ для получения спектра с заданными постоянными $\sigma_{\min}$ и $\sigma_{\max}$. Рисунок 1 иллюстрирует идею создания набора прямоугольных распределений внутренних сингулярных значений. На нем намеренно не показаны одинаковые во всех случаях крайние $\sigma_{\min}$ и $\sigma_{\max}$. Матрица с заданным спектром создается по методу, предложенному в [7] и детально описанному нами в [1]. Исследуется точность сингулярного разложения тремя рассмотренными методами из программных библиотек линейной алгебры на наборе из 100 матриц, каждая из которых имеет прямоугольный спектр внутренних сингулярных значений с разными свойствами. Все, кроме крайних, сингулярные значения могут находиться на одном узком отрезке спектра, а могут быть равномерно распределены по достаточно широкой его части; большинство сингулярных значений могут быть сгруппированы как у левого конца спектра, так и у правого.

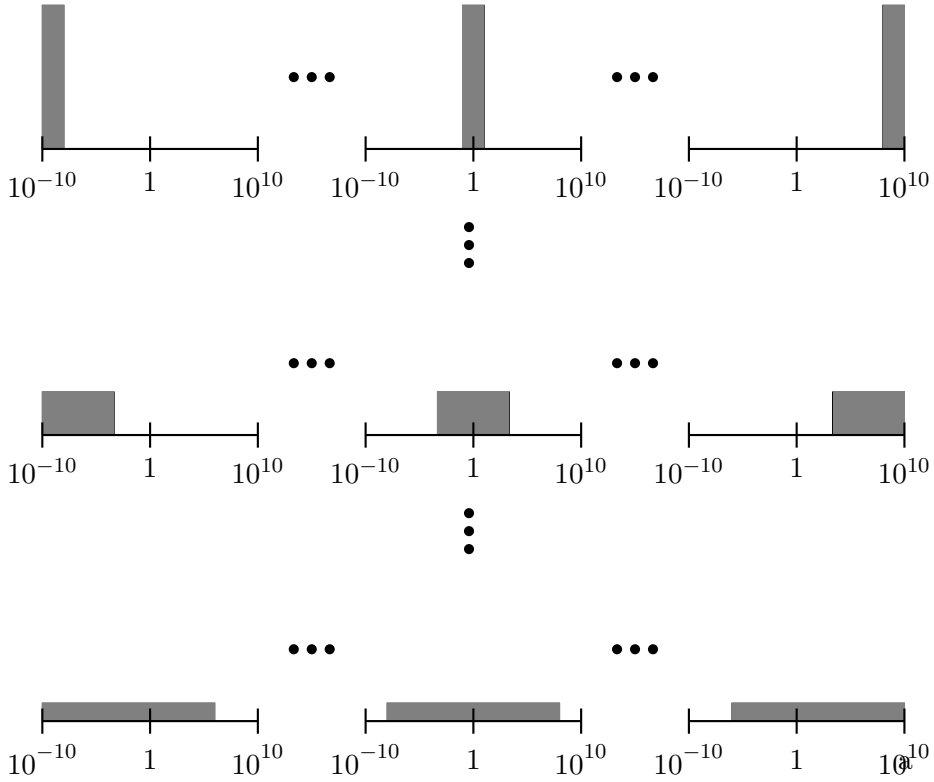


Рисунок 1. Иллюстрации набора тестовых распределений внутренних сингулярных значений.

4. Эксперименты

Результаты моделирования представлены в виде трехмерных графиков на рисунках 2-7. На них по оси `center` отмечены нормированные к ширине спектра положения центра прямоугольной функции: 0 соответствует левому краю (малые сингулярные значения), 1 соответствует правому краю (большие сингулярные значения). По оси `width` отложена ширина прямоугольной функции: от 0.05 до 0.9 от ширины всего спектра. По вертикальной оси строятся значения выбранной метрики погрешности нахождения сингулярных значений в логарифмической шкале.

Для демонстрации результатов используется предложенная нами в [1] среднеквадратичная относительная мера погрешности:

$$\mathrm{RMSRE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\sigma_i - \hat{\sigma}_i}{\sigma_i}\right)^2}. \qquad (5)$$

Отметим, что погрешность RMSRE чувствительна к выбросам – большая относительная погрешность отыскания отдельных сингулярных значений значительно влияет на общий результат. Рассмотрим подробнее два крайних случая в наших экспериментах.

В первом из них `center` = 0.8 и `width` = 0.05, прямоугольная функция распределения сингулярных значений "узкая" и сдвинута к правому краю, т.е. все сингулярные значения, кроме самого малого $\sigma_{\min}$, расположены кучно и велики (находятся в интервале от 10^9 до 10^{10}). Вектор относительных погрешностей сингулярных значений, отсортированных от больших к меньшим, выглядит следующим образом:

$$(7.6\text{e}{-16}, 4.8\text{e}{-15}, 4.4\text{e}{-15}, \ldots, 7\text{e}{-15}, 8.6\text{e}{-15}, 4.6\text{e}{+04}).$$

Все сингулярные значения находятся правильно, за исключением одного самого малого, имеющего большую относительную погрешность, что и приводит к увеличению (4), например, при использовании метода dgesvd до величины RMSRE = 1023.4.

Обратная ситуация имеет место при `center` = 0.1 и `width` = 0.05, то есть все сингулярные значения, кроме самого большого $\sigma_{\max}$, кучно сгруппированы в интервале от 10^{-10} до 10^{-9} около $\sigma_{\min}$. В этом случае вектор относительных погрешностей таков:

$$(1.9\text{e}{-16}, 1.5\text{e}{+04}, 1.2\text{e}{+04}, \ldots, 1.6, 1.5, 1.4). \tag{6}$$

Все сингулярные значения, за исключением одного самого большого, найдены с очень большой относительной погрешностью. Тем не менее, для этого случая RMSRE = 839.8 и сопоставима для обоих случаев, несмотря на принципиальную разницу точности нахождения сингулярных значений в целом (большинство значений найдены точно против большинства значений найдены очень неточно).

При увеличении `width` ширина внутренней части сингулярного спектра между $\sigma_{\min}$ и $\sigma_{\max}$ расширяется, расстояния между смежными сингулярными значениями увеличиваются и выраженность важного описанного выше эффекта уменьшается. Как мы продемонстрировали в [1], традиционно используемые абсолютная погрешность $E_{\text{абс}}(\hat{x}) = \|\hat{x} - x\|$, относительная погрешность $E_{\text{отн}}(\hat{x}) = \|\hat{x} - x\|/\|x\|$ и поэлементная относительная погрешность $\max\limits_i \|\hat{x}_i - x_i\|/\|x_i\|$ слабо отражают этот эффект численно. Чувствительность метрики RMSRE к нему может быть улучшена, что позволит избежать влияния значительной ошибки нахождения самого малого сингулярного значения. Для этого возьмём медиану набора относительных погрешностей

$$\text{med\_rel} = \text{median}\ \frac{|\hat{\sigma}_i - \sigma_i|}{\sigma_i}. \tag{7}$$

Дополнительное вычисление RMSRE с изъятием самого малого значения $\sigma_{\min}$ хорошо отражает особенности наших численных экспериментов, но не подходит для использования в общем случае, в отличие от (7). На рисунках 2, 4, 6 приведены медианы относительных погрешностей. На рисунках 3, 5, 7 для сравнения приведены значения метрики RMSRE для всех сингулярных значений, кроме самого малого.

5. Выводы

Продемонстрировано, что разные метрики могут как по сути игнорировать, так и преувеличивать особенности найденных численно сингулярных значений; например, широко используемая в численной линейной алгебре погрешность по L^2-норме вектора результатов $\|\hat{\sigma} - \sigma\|$ обычно скрывает неточное вычисление основной массы сингулярных значений при найденных верно нескольких больших значениях, что хорошо соответствует идеям метода главных компонент. Наоборот, предложенная нами в [1] метрика RMSRE чувствительна к неточному нахождению любых, даже немногих малых сингулярных значений, при точном отыскании большей их части, что может быть важно для методов опознания по сжатию (англ. compressive sensing), при обработке мультиспектральных данных и реконструкции объектов по проекциям, где желательно знать весь сингулярный спектр с высокой точностью.

При использовании медианы набора относительных погрешностей в метрике RMSRE удается продемонстрировать явные зависимости общей точности нахождения сингулярных значений от их распределения, обеспечивая паритет между двумя упомянутыми выше метриками. При больших числах обусловленности матрицы (10^{20}), можно сделать следующий вывод: чем меньше по модулю сингулярное значение, тем хуже оно находится. Таким образом, точнее всего сингулярные значения вычисляются, если их большинство плотно сгруппировано около самого большего. Наоборот, наихудшим случаем является группировка сингулярных значений около меньшего при наличии одного значительно большего.

Список литературы

[1] Дроздов И.Ю., Парфенов Д.В., “Влияние распределения спектра матрицы на точность сингулярного разложения”, *Интеллектуальные системы. Теория и приложения*, 2023.

[2] Anderson E. et al., *LAPACK Users' Guide*, SIAM, 1999.

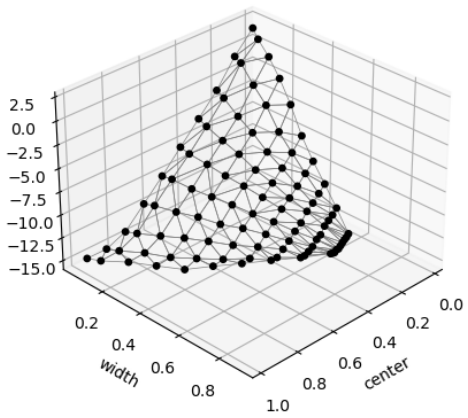


Рис. 2. Метод xgesvd, медиана

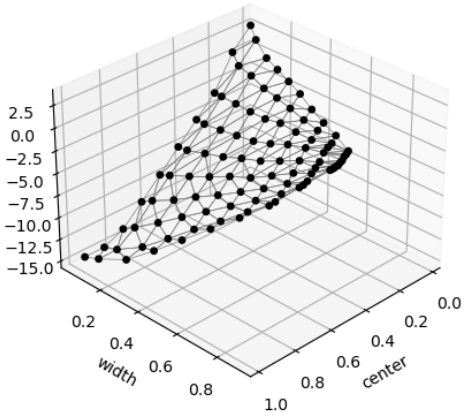


Рис. 3. Метод xgesvd, RMSRE без малого сингулярного значения

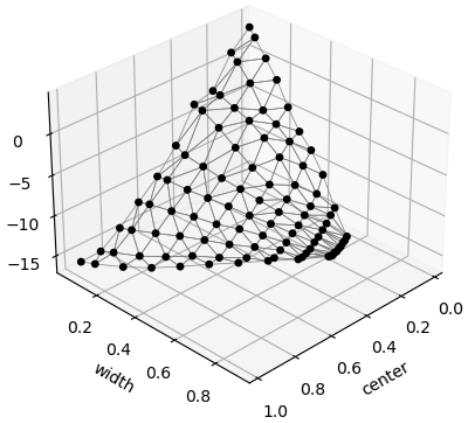


Рис. 4. Метод xgesdd, медиана

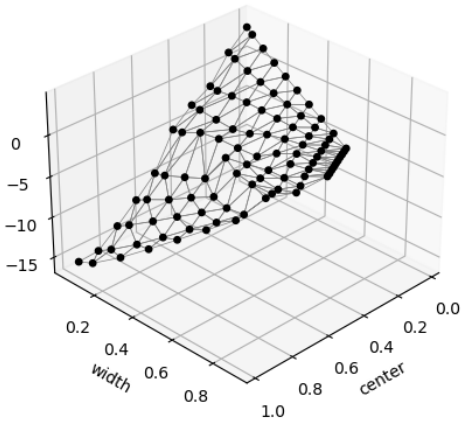


Рис. 5. Метод xgesdd, RMSRE без малого сингулярного значения

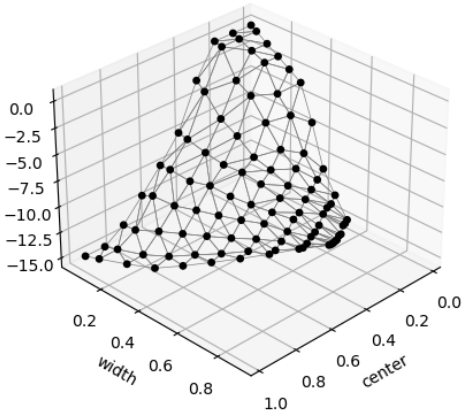


Рис. 6. Метод BDC, медиана

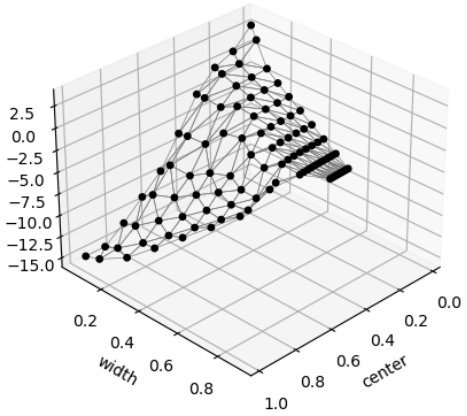


Рис.7. Метод BDC, RMSRE без малого сингулярного значения

[3] Guennebaud G., Jacob B. et al., *Eigen v3*, http://eigen.tuxfamily.org, 2010.

[4] Demmel J., Kahan W., "Computing Small Singular Values of Bidiagonal Matrices With Guaranteed High Relative Accuracy", *SIAM J. Sci. Stat. Comput*, **11**:5 (1990), 873-912.

[5] Gu M., Eisenstat S., "A Divide-and-Conquer Algorithm for the Bidiagonal SVD", *SIAM J. Matrix Anal. Appl.*, **16**:1 (1995), 79-92.

[6] Jessup R., Sorensen D., "A Parallel Algorithm for Computing the Singular Value Decomposition of a Matrix", *SIAM J. Matrix Anal. Appl.*, **15**:2 (1995), 530-548.

[7] Golub G.H., Van Loan C.F., *Matrix Computations*, Johns Hopkins University Press, 2013.

Accuracy of algorithms of singular value decomposition for matrices with various spectra

Drozdov I.Yu., Parfenov D.V.

We continue to develop our new approach of treating singular spectrum of a matrix as a probability density function to investigate dependencies between accuracy of numerical computation of singular values and spectrum. We conduct massive numerical experiments to

demonstrate such dependencies in our new suggested metrics: root-mean-square relative error and median. We present illustrative plots of such dependencies and analyze conclusiveness of these metrics.

Keywords: singular value decomposition, SVD, condition number, matrix spectrum, numerical stability

References

[1] Drozdov I.Yu, Parfenov D.V., "Relationship between accuracy of singular value decomposition and distribution of singular values", *Intelligent Systems. Theory and Applications*, 2023.

[2] Anderson E. et al., *LAPACK Users' Guide*, SIAM, 1999.

[3] Guennebaud G., Jacob B. et al., *Eigen v3*, http://eigen.tuxfamily.org, 2010.

[4] Demmel J., Kahan W., "Computing Small Singular Values of Bidiagonal Matrices With Guaranteed High Relative Accuracy", *SIAM J. Sci. Stat. Comput*, **11**:5 (1990), 873-912.

[5] Gu M., Eisenstat S., "A Divide-and-Conquer Algorithm for the Bidiagonal SVD", *SIAM J. Matrix Anal. Appl.*, **16**:1 (1995), 79-92.

[6] Jessup R., Sorensen D., "A Parallel Algorithm for Computing the Singular Value Decomposition of a Matrix", *SIAM J. Matrix Anal. Appl.*, **15**:2 (1995), 530-548.

[7] Golub G.H., Van Loan C.F., *Matrix Computations*, Johns Hopkins University Press, 2013.

Диффузионная модель со скачками с возвращающейся к среднему логнормальной волатильностью

Р. В. Честнов[1]

На данный момент существует множество стохастических моделей, построенных для различных концепций рынка. Однако, практически все подобные модели основаны и протестированы для традиционных рынков, в то время как в настоящее время рынок криптоактивов набирает колоссальные обороты по объемам и капитализации рынка. По данной причине напращивается идея о создании такой модели, которая была бы построена специально под рынок криптоактивов с учетом всех ее особенностей и моделей поведения. В данной работе мы попробуем посмотреть на некоторые признаки, которые заметны для криптоактивов, и построить стохастическую модель, учитывающую их, после чего сравнить её с другой похожей по структуре моделью. Сразу стоит отметить, что по аналогии с тем, как используются традиционные модели для крипторынков, наша модель для крипторынков так же будет хорошо интерпретировать традиционный рынок. Более того, даже лучше, чем в обратном случае, так как основная идея моей модели заключается в учитывании различных критических событий, происходящих с тем или иным активом, но калибруя модель нужным образом, мы можем их не учитывать.

Ключевые слова: Финансовая математика, стохастическое исчисление, криптоактивы, нейронные сети, уравнение Фейнмана-Каца, модель Бэйтса, функция правдоподобия.

1. Вступление

Текущие реалии финансовых рынков находится в периоде активных преобразований, ведь новые и старые финансовые инструменты сталкиваются на фоне динамичной рыночной обстановки. Традиционные активы, такие как акции и облигации, на протяжении многих лет привлекали внимание финансовых аналитиков и моделей прогнозирования. Однако, в последние десятилетия мы стали свидетелями стремительного развития нового класса активов — криптовалют и криптоактивов. Это привносит важные вызовы в разработке новых финансовых моделей,

[1] *Честнов Роберт Валентинович* — количественный исследователь по DeFi продуктам в Qset, выпускник каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: rvchestnov@gmail.com.

Chestnov Robert Valentinovich — quantitative researcher on DeFi products in Qset, graduate student of Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

способных адаптироваться к уникальным особенностям и рискам, связанным с крипторынками. Одной из ключевых особенностей является высокая волатильность и нестабильность цен, что может привести к резким колебаниям. Этот фактор создает повышенные риски как для инвесторов, так и для аналитиков, и подчеркивает важность разработки специализированных моделей, способных учитывать эту динамику цен.

Кроме того, криптовалютные рынки отличаются не только волатильностью, но и своим характером торговли, структурой участников и воздействием различных факторов на динамику цены, таких как новости о регулировании, технические обновления и события в сообществе, что отчасти объясняет такую непредсказуемость в поведении криптоактивов. На данный момент невозможно смоделировать описанные внешние социальные факторы, но в данной работе будет применен классический подход к данной проблеме - отделение подобных деталей от рыночной составляющей цены путем добавления соответствующие параметры в модель.

В предложенной стохастической модели планируется учесть несколько особенностей динамики цен криптоактивов, которые существенно отличаются от традиционных финансовых инструментов. Одной из таких особенностей является ненормальное распределение волатильности, которая характеризуется более высокой вероятностью крупных колебаний цен и резких движений на рынке. В отличие от нормального распределения, которое используется в большинстве стохастических моделей, логнормальное распределение лучше отражает действительность на крипторынках, где наблюдается большое количество экстремальных событий и высокая волатильность. Кроме того, еще одной важной особенностью криптовалютных рынков является наличие скачков цен, то есть внезапных и резких изменений цены, которые могут происходить в результате крупных событий. Учет таких скачков в стохастической модели играет ключевую роль в повышении ее точности и надежности, а также в обеспечении более реалистичного прогнозирования ценовых траекторий на крипторынках. Таким образом, разработка стохастической модели, учитывающая следующие два фактора, представляет собой значимый шаг в совершенствовании аналитических инструментов для криптовалютных рынков. Это позволит не только более точно оценивать риски и возможности инвестиций в криптоактивы, но и развивать более эффективные стратегии управления портфелем в условиях высокой волатильности и нестабильности рыночной среды.

Работа организована по разделам следующим образом: в разделе 2 мы подробнее посмотрим на некоторые характеристики криптоактивов и подробнее распишем процесс создания стохастической модели и идею, которая за ней стоит; в разделе 3 расписана методология вычисления

кумулятивной функции распределения для модели; в разделе 4 мы займемся численным решением уравнения, полученного в предыдущей главе; в разделе 5 нашей задачей будет построение логарифмической функции правдоподобия для калибровки коэффициентов модели и сравнение полученных результатов с моделью Бэйтса. В разделе 6 будут приведены выводы и предложения по доработке работы.

2. Идея создания модели и её описание

Как было упомянуто в введении, первоначальной целью является выделение некоторых особенности поведения криптоактивов, которые нехарактерны для существующих стохастических моделей. Для начала стоит отметить, что на практике и даже в теории в силу сложности стохастического анализа некоторые детали опускаются и не вносятся в модели. К примеру, самая популярная стохастическая модель для описания поведения цены – модель Геометрического Броуновского Движения (GBM) – предполагает, что доходность актива имеет нормальное распределение и что изменения цен независимы и стационарны. GBM широко используется в финансовой математике и анализе рынков для моделирования ценовых процессов. Модель описывается следующим стохастическим уравнением:

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \tag{0}$$

или, переходя к логарифму цены,

$$dX_t = \mu dt + \sigma dW_t; \quad X_t = \ln(S_t) \tag{1}$$

где S_t – цена актива, X_t – лог-цена актива, μ – тренд, σ – волатильность, dW_t – инкремент броуновского движения. Данная модель является наиболее распространённой благодаря удобству использования и простоте реализации, однако это также её главный недостаток. GBM упоминается в данном контексте, чтобы подчеркнуть, что в математической интерпретации жизненных процессов часто приходится жертвовать точностью ради простоты реализации. В последующих исследованиях учёные совершенствовали существующие модели. Среди наиболее распространённых подходов можно выделить преобразование тренда или волатильности из постоянных параметров в параметры, зависящие от времени, выделение различных свойств в динамике активов (например, возврат к среднему, как в модели Орнштайна-Уленбека), адаптацию моделей под соответствующие финансовые инструменты (например, модель Блэка-Шоулза), а также изменение динамики броуновского движения (например, модель дробного геометрического

броуновского движения). Основная цель настоящей работы заключается в создании альтернативной стохастической модели, использующей как известные методы, так и новые подходы.

Для исследования были выбраны 16 криптовалютных токенов, основываясь на рыночной капитализации, торговом объёме и популярности соответствующих блокчейнов и протоколов: BTC, ETH, LINK, MATIC, UNI, MKR, LDO, AAVE, QNT, MANA, CRV, 1INCH, ZRX, FXS, SUSHI и YFI. Очевидно, что при создании модели для крипторынков в первую очередь необходимо учесть их непредсказуемость и значительные колебания в динамике цен. Поэтому волатильность должна быть представлена как величина, зависящая от времени, и выделена в отдельный стохастический процесс, который будет моделироваться соответствующим образом.

Для начала рассмотрим минутные данные за один день по данным токенам и проведём базовый анализ. На практике, без использования сложных инструментов, волатильность часто аппроксимируют через стандартное отклонение по скользящему окну. Применим этот подход для оценки динамики волатильности на концептуальном уровне. Представим графики полученных волатильностей:

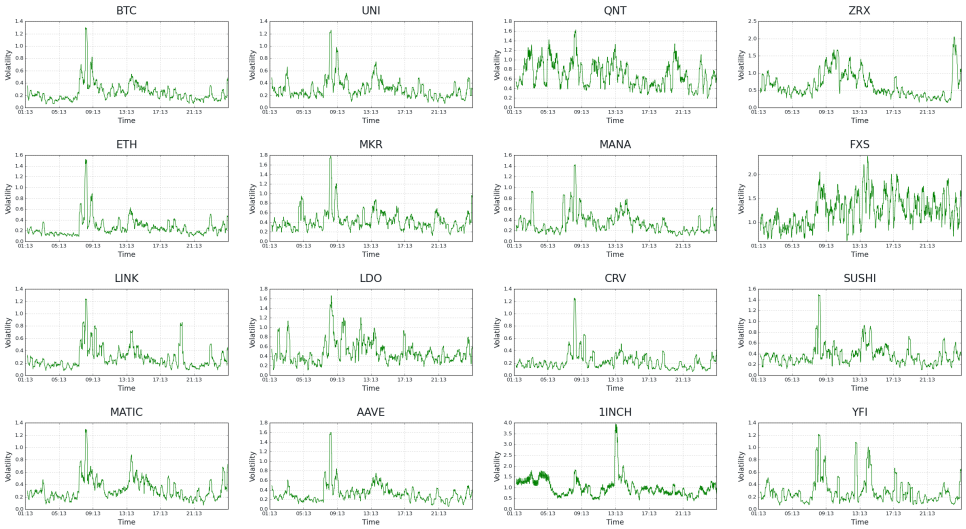


Рисунок 1: Графики аппроксимированных волатильностей для рассматриваемых токенов, построенных по 15-минутному окну

Анализ данных, представленных на Рисунке 1, показывает, что значения волатильности демонстрируют выраженные колебания вверх и вниз на коротких временных интервалах при сохранении относительно постоянного тренда. Это указывает на то, что наиболее корректным решением является описание волатильности в виде процесса, обладающего

свойством возврата к среднему значению. Аналогичный метод используется в модели Орнштайна-Уленбека, где динамика процесса также стремится к среднему состоянию. В классической форме процесс Орнштайна-Уленбека применяется для описания динамики цен активов и описывается следующим стохастическим уравнением:

$$dX_t = \theta(\mu - X_t)dt + \sigma dW_t, \tag{2}$$

где μ – среднее значение цены, θ – скорость возврата к среднему, σ – волатильность актива, dW_t – приращение броуновского движения. В нашей модели мы применим данную концепцию для моделирования волатильности. Тем не менее, существует ещё один важный фактор, который необходимо включить в модель волатильности для её усовершенствования. Для того, чтобы выявить этот фактор и лучше понять особенности поведения волатильности, необходимо проанализировать её распределение. Для этого построим гистограммы для полученных ранее данных по волатильностям:

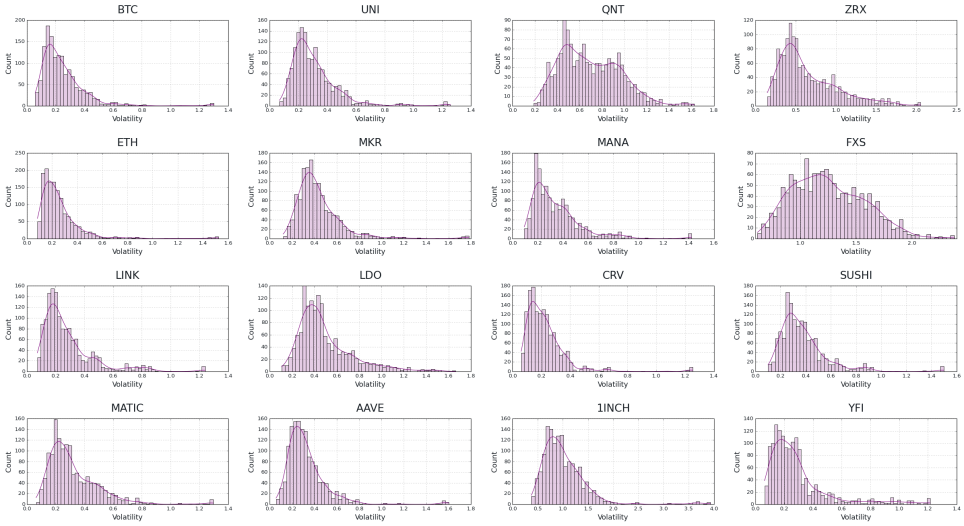


Рисунок 2: Гистограммы аппроксимированных волатильностей для рассматриваемых токенов, построенных по 15-минутному окну

Следует отметить, что в основе большинства стохастических моделей заложено предположение о нормальном распределении волатильности токенов. Даже при использовании модели Орнштейна–Уленбека данное предположение характерно для традиционных финансовых рынков. Однако, исходя из анализа гистограмм, можно заметить, что поведение волатильности криптоактивов ближе к логнормальному распределению, чем к стандартному нормальному. Данная особенность обусловлена спецификой криптовалютных рынков, где значительно чаще наблюдаются

резкие колебания цен и высокие значения волатильности. В связи с этим необходимо учесть данный фактор при построении модели. Одним из решений может стать добавление компоненты волатильности в стохастический процесс, описывающий броуновское движение. Таким образом, в обозначениях для модели волатильности мы получаем следующую стохастическую модель:

$$d\sigma_t = \kappa(\theta - \sigma_t)dt + \gamma\sigma_t dW_t^\sigma, \tag{3}$$

где θ – среднее значение цены, κ – скорость возврата к среднему, γ – коэффициент колебания волатильности, dW_t^σ – приращение броуновского движения, соответствующего процессу σ_t.

Обратимся к анализу гистограмм (Рисунок 2), который демонстрирует, что волатильность характеризуется наличием экстремальных значений, обусловленных значительными колебаниями цен. Эти значения особенно выражены для таких криптоактивов, как UNI, MKR и 1INCH. Следовательно, для более точного описания рынка необходимо учитывать скачки цен на рассматриваемые активы. В целях моделирования таких скачков, предлагается включить компоненту скачков в стохастическое уравнение, описывающее динамику цены актива. Скачки будут моделироваться пуассоновским процессом с интенсивностью λ, а их величина — случайной величиной J, распределённой по нормальному закону с нулевым средним значением и стандартным отклонением ε. Альтернативные подходы к моделированию скачков, такие как метод, описанный в работе [7], также могли бы быть применены. Однако в данной работе нормальное распределение скачков представляется приемлемым и достаточным для описываемой модели. Таким образом, стохастическое уравнение для динамики цен можно записать следующим образом:

$$dX_t = \mu dt + \sigma_t dW_t^X + JdN_t, \tag{4}$$

где X_t – лог-цена актива, μ – тренд, σ_t – волатильность, dW_t^X – инкремент броуновского движения, соответствующего процессу X_t, N_t – пуассоновский процесс с интенсивностью λ, J – размер скачка, который описывается нормальным распределением с нулевым средним и некой дисперсией.

Однако, уравнение в его текущем виде не предоставляет достаточной интерпретации. При работе со случайными процессами мы стремимся наблюдать четкую динамику поведения случайной величины во времени. Тем не менее, в данном виде уравнение не отражает полной картины поведения величины, отвечающей за размер скачков. В связи с этим было предложено новое концептуальное решение, ранее не применявшееся в подобного рода моделях: выделить величину скачка в отдельный процесс

J_t. Такой подход позволяет более точно описать независимые от рынка колебания цен, которые оказывают значительное влияние на динамику актива в конкретные моменты времени, обусловленные срабатыванием пуассоновского процесса. Это нововведение позволяет нам лучше понять вклад скачков в общую динамику цен. Таким образом, можно записать следующее стохастическое уравнение для описания так называемой «нерыночной» компоненты:

$$dJ_t = \varepsilon dW_t^J, \tag{5}$$

где ε – вариация величины скачков, dW_t^J – инкремент броуновского движения, соответствующего процессу J_t. Несмотря на то, что это уравнение отображает динамику изменения скачков, такая запись корректно отображает идею нормального распределения для самой величины скачков в силу свойств винеровского процесса.

Следует подчеркнуть, что в контексте финансовой математики необходимо, чтобы все применяемые случайные процессы являлись мартингалами, поскольку модели, используемые для оценки активов, предполагают отсутствие арбитражных возможностей и справедливую оценку параметров. Однако, сам по себе пуассоновский процесс не является мартингалом. Для того чтобы привести его в соответствие с данными требованиями, необходимо добавить корректирующий инкремент $-\lambda t$, тем самым превращая его в компенсированный пуассоновский процесс

Теорема 1. Пусть $N(t)$ – пуассоновский процесс с интенсивностью λ. Определим *компенсированный пуассоновский процесс* следующим образом:

$$M(t) = N(t) - \lambda t.$$

Тогда процесс $M(t)$ является мартингалом.

Доказательство данной теоремы приведено в учебнике [8]

Таким образом, возникает необходимость записи пуассоновского процесса в виде компенсированного пуассоновского процесса с добавленной компонентой λt. Это позволяет учесть компенсирующий инкремент и сделать процесс мартингалом. В результате, уравнение (4) можно переписать в следующем виде:

$$dX_t = (\mu + \lambda J_t)dt + \sigma_t dW_t^X + J_t d\tilde{N}_t, \tag{6}$$

где $\tilde{N}_t$ – соответствующий компенсированный пуассоновский процесс. Данная запись будет удобной при дальнейших вычислениях.

Прежде чем перейти к окончательной формулировке стохастической системы, важно обратить внимание на возможную корреляцию между винеровскими процессами, описывающими динамику X_t и σ_t. Обозначим

коэффициент корреляции между ними как ρ. Следует также отметить, что винеровский процесс, связанный со скачками J_t, предполагается независимым от процессов X_t и σ_t. Это связано с тем, что J_t представляет нерыночные компоненты, которые не зависят от волатильности и логарифмической цены актива. Данное предположение оправдано, поскольку скачки, описываемые J_t, имеют другую природу и не коррелируют с основными рыночными процессами. Таким образом, объединив выражения (3), (5) и (6) с учётом указанных зависимостей и сделанных предположений, мы получаем следующую стохастическую модель, которая полноценно описывает динамику рассматриваемых процессов:

$$\begin{cases} dX_t = (\mu + \lambda J_t)dt + \sigma_t dW_t^X + J_t d\tilde{N}_t, \\ d\sigma_t = \kappa(\theta - \sigma_t)dt + \gamma\sigma_t dW_t^\sigma, \\ dJ_t = \varepsilon dW_t^J, \\ dW_t^X dW_t^\sigma = \rho, \\ dW_t^X dW_t^J = dW_t^\sigma dW_t^J = 0. \end{cases}$$

Стоит отметить, что по структуре данная модель очень похожа на модель стохастической волатильности Бэйтса, описанную в его работе [1], которая является расширением известной модели Хестона [4], включающим в себя компоненты скачков. Модель Бэйтса описывается следующей стохастической системой:

$$\begin{cases} dS_t = (r - \lambda\mu_J)\, S_t dt + \sqrt{V_t} S_t d\widetilde{W}_t^{(1)} + J S_t d\widetilde{N}_t, \\ dV_t = \kappa\,(\theta - V_t)\, dt + \sigma_v \sqrt{V_t} d\widetilde{W}_t^{(2)}, \\ d\widetilde{W}_t^{(1)} d\widetilde{W}_t^{(2)} = \rho dt, \end{cases}$$

при этом скачки следуют логнормальному распределению:

$$\eta := \ln(1 + J) \sim \mathcal{N}\left(\mu_J, \sigma_J^2\right).$$

Данная модель учитывает как наличие скачков, так и ненормальную структуру распределения волатильности, а также её возврат к среднему значению. Основные отличия заключаются в том, что в модели Бэйтса скачки распределены по логнормальному закону, волатильность имеет гамма-распределение (при предположении, что процесс не вырождается), а величина скачков рассматривается как случайная величина. В нашей же модели величина скачков описывается отдельным случайным процессом. Учитывая структурное сходство предлагаемой модели с моделью Бэйтса, последняя представляется наиболее подходящим кандидатом для сравнения. Поэтому для анализа мы выберем именно её.

3. Построение уравнения для кумулятивной функции распределения

Для дальнейшего анализа необходимо использовать мощный инструмент для оценки модели и сравнения её характеристик. В качестве такого инструмента предлагается рассчитать переходную функцию плотности вероятности. Поскольку речь идёт о функции плотности вероятности, требуется составить одно из уравнений Колмогорова, которое описывает её изменение в контексте уравнений в частных производных. Однако, прежде чем приступить к его построению, необходимо определить, как будет выглядеть дифференциал некоторой гладкой функции для рассматриваемой модели. Таким образом, мы приходим к следующей теореме:

Теорема 2. Пусть $f : \mathbb{R}^4 \to \mathbb{R}$ – некая дважды дифференцируемая функция. Тогда для нашей модели справедлива следующая формула:

$$\begin{aligned} df(t,X,\sigma,J) = \Bigg[& \frac{\partial f}{\partial t} + \mu\frac{\partial f}{\partial X} + \lambda\big[f(t,X+J,\sigma,J) - f(t,X,\sigma,J)\big] + \\ & + \kappa(\theta-\sigma)\frac{\partial f}{\partial \sigma} + \frac{1}{2}\sigma^2\frac{\partial^2 f}{\partial X^2} + \frac{1}{2}\gamma^2\sigma^2\frac{\partial^2 f}{\partial \sigma^2} + \frac{1}{2}\varepsilon^2\frac{\partial^2 f}{\partial X^J} + \\ & + \rho\gamma\sigma^2\frac{\partial^2 f}{\partial X \partial \sigma}\Bigg] dt + \sigma\frac{\partial f}{\partial X}dW_t^X + \gamma\sigma\frac{\partial f}{\partial \sigma}dW_t^\sigma + \varepsilon\frac{\partial f}{\partial J}dW_t^J + \\ & + \big[f(t,X+J,\sigma,J) - f(t,X,\sigma,J)\big]d\tilde{N}_t \end{aligned}$$

В качестве упрощения можно разложить компонету с компенсированным пуассоновским процессом по формуле Тейлора до второго порядка, после чего мы получим, что

$$f(t,X+J,\sigma,J) - f(t,X,\sigma,J) \approx J\frac{\partial f}{\partial X} + J^2\frac{\partial^2 f}{\partial X^2}$$

Доказательство. Рассмотрим систему, которой описывается наша модель. Свернем компенсированный пуассоновский процесс обратно в стандартный пуассоновский процесс:

$$dX_t = \mu dt + \sigma_t dW_t^X + J_t dN_t.$$

Можно отметить следующее: для всех компонентов системы выполняется многомерная лемма Ито, что позволяет вывести практически всю формулу, приведённую в теореме. Исключение составляет компонент с пуассоновским процессом $J_t dN_t$, поскольку классическая лемма Ито не включает пуассоновские процессы. Таким образом, необходимо доказать лемму Ито для пуассоновского процесса. Доказательство этой

леммы в интегральной форме представлено в работах [5] и [8], однако для наших целей требуется дифференциальная форма. Поэтому мы предлагаем собственное доказательство в дифференциальной форме, которое отличается от упомянутых источников.

Доказывать будем для более простого, но в то же время достаточно общего случая, так как результат будет справедлив для нашего сценария без изменений. Постановка задачи следующая: рассмотрим процесс $dY_t = j(Y_t, t)dN_t$. Задав гладкую функцию $g(t, N_t)$, необходимо вычислить дифференциал $dg(t, Y_t)$.

По определению дифференциала, запишем

$$dg(t, Y_t) = g(t+dt, Y_t+dY_t)-g(t, Y_t) = g(t+dt, Y_t+j(Y_t, t)dN_t)-g(t, Y_t) \quad (7)$$

Для пуассоновского процесса известно следующее свойство: компонента dN_t может принимать только два значения: 1 с вероятностью λdt (процесс активировался) и 0 с вероятностью $1 - \lambda dt$ (процесс не активировался). Соответственно, первое слагаемое мы можем представить в виде суммы двух «ортогональных» компонент:

$$\begin{aligned} g(t + dt, Y_t + j(Y_t, t)dN_t) = \\ = g(t + dt, Y_t + j(Y_t, t))dN_t + g(t + dt, Y_t)(1 - dN_t). \end{aligned} \quad (8)$$

Таким образом, если $dN_t = 1$, то скачок размера $j(Y_t, t)$ активируется, а второе слагвемое обнулится; если $dN_t = 0$, то скачок не активировался (или равен нулю), а первое слагаемое обнулится.

Рассмотрим первое слагаемое. Функцию g мы можем разложить по формуле Тейлора по первому аргументу до первой производной:

$$g(t + dt, Y_t + j(Y_t, t))dN_t = g(t, Y_t + j(Y_t, t))dN_t + \frac{\partial g(t, Y_t + j(Y_t, t))}{\partial t} dtdN_t$$

Заметим, что второе слагаемое обнулится в силу того, что $dtdN_t = 0$ (изометрия Ито для пуассоновского процесса), после чего мы получим, что

$$g(t + dt, Y_t + j(Y_t, t))dN_t = g(t, Y_t + j(Y_t, t))dN_t$$

Аналогично можно разложить по Тейлору второе слагаемое в (8) (компонента с первой производной здесь будет равна нулю по той же причине):

$$g(t + dt, Y_t)(1 - dN_t) = g(t, Y_t)(1 - dN_t)$$

Итого, с учетом полученных разложений формула (8) перепишется в виде

$$g(t + dt, Y_t + j(Y_t, t)dN_t) = g(t, Y_t + j(Y_t, t))dN_t + g(t, Y_t)(1 - dN_t)$$

Подставляя эту формулу в уравнение (7), получим

$$dg(t,Y_t) = g(t,Y_t + j(Y_t,t))dN_t + g(t,Y_t)(1-dN_t) - g(t,Y_t) = \\ = \big(g(t,Y_t + j(Y_t,t)) - g(t,Y_t)\big)dN_t.$$

Теперь вернемся опять к записи в форме компенсированного пуассоновского процесса, переписав дифференциал в следующем виде:

$$dg(t,Y_t) = \big(g(t,Y_t + j(Y_t,t)) - g(t,Y_t)\big)d(N_t - \lambda t + \lambda t) = \\ = \lambda\big(g(t,Y_t + j(Y_t,t)) - g(t,Y_t)\big)dt + \big(g(t,Y_t + j(Y_t,t)) - g(t,Y_t)\big)d\tilde{N}_t.$$

В итоге, мы получили лемму Ито в дифференциальном виде для компенсированного пуассоновского процесса. Соединяя вместе стандартную многомерную лемму Ито в доказанной только что лемму Ито для компенсированного пуассоновского процесса, мы получим, что

$$df(t,X,\sigma,J) = \Bigg[\frac{\partial f}{\partial t} + \mu\frac{\partial f}{\partial X} + \lambda\big[f(t,X+J,\sigma,J) - f(t,X,\sigma,J)\big] + \\ + \kappa(\theta-\sigma)\frac{\partial f}{\partial \sigma} + \frac{1}{2}\sigma^2\frac{\partial^2 f}{\partial X^2} + \frac{1}{2}\gamma^2\sigma^2\frac{\partial^2 f}{\partial \sigma^2} + \frac{1}{2}\varepsilon^2\frac{\partial^2 f}{\partial X^J} + \\ + \rho\gamma\sigma^2\frac{\partial^2 f}{\partial X \partial \sigma}\Bigg]dt + \sigma\frac{\partial f}{\partial X}dW_t^X + \gamma\sigma\frac{\partial f}{\partial \sigma}dW_t^\sigma + \varepsilon\frac{\partial f}{\partial J}dW_t^J + \\ + \big[f(t,X+J,\sigma,J) - f(t,X,\sigma,J)\big]d\tilde{N}_t.$$

□

Теперь необходимо записать уравнение для переходной функции плотности вероятности $p(t,X,\sigma,J)$. Однако следует отметить один важный аспект: эволюция переходной плотности описывается уравнением Фоккера-Планка (известным также как прямое уравнение Колмогорова). При записи уравнения Фоккера-Планка для интервала времени $\tilde{t} \le t \le T$, необходимо также учесть граничное условие. В нашем случае оно будет задано следующим образом: в момент времени $\tilde{t}$

$$p(t_0,X,\sigma,J) = \delta(X-\tilde{X})\cdot\delta(\sigma-\tilde{\sigma})\cdot\delta(J-\tilde{J}),$$

где δ – дельта-функция Дирака, а $\tilde{X},\tilde{\sigma},\tilde{J}$ – некие начальные параметры.

Здесь возникает основная проблема: уравнение Фоккера-Планка для предложенной модели является слишком сложным для аналитического решения. Тем не менее, ограничение с дельта-функцией можно эффективно обработать аналитически. Однако, при численном решении уравнения Фоккера-Планка возникает необходимость в численной аппроксимации дельта-функции, так как ее невозможно воспроизвести

в чистом виде для использования в вычислительных алгоритмах. Среди существующих методов численной аппроксимации дельта-функции наиболее известен подход, основанный на сеточных методах. В этом случае дельта-функция представляется в виде треугольника, где боковые вершины располагаются в соседних узлах сетки, а третья вершина — в граничной точке. Высота треугольника подбирается таким образом, чтобы его площадь была равна единице. Основная идея метода заключается в том, что по мере уменьшения шага разбиения треугольник сжимается, сохраняя площадь равной единице, постепенно приближаясь к истинной дельта-функции. Однако данный метод не представляется возможным в нашем случае, так как для более точного анализа предполагается численное решение параметрического уравнения в частных производных. Это означает, что постоянные параметры модели также будут рассматриваться как меняющиеся величины в определённом диапазоне. Таким образом, функция плотности вероятности будет зависеть от 14 переменных, включая исходные переменные и изменяющиеся параметры. Применение сеточной аппроксимации в данном случае неприемлемо ввиду ограничений вычислительных ресурсов. Для адекватного разбиения интервалов, которое достаточно точно аппроксимировало бы дельта-функцию (например, минимум 100 узлов), сеточный метод потребовал бы порядка 10^{28} итераций, включающих решение систем линейных уравнений. Это делает задачу вычислительно невыполнимой в разумные сроки.

Учитывая вышеописанные соображения, необходимо избежать непосредственного использования дельта-функции. Но стоит отметить, что преимуществом дельта-функции является её простая и удобная запись при интегрировании. Именно поэтому возникает желание перейти к интегральной форме, которая позволит сохранить эти свойства. Однако, если в уравнении Фоккера-Планка мы будем брать интеграл от дельта-функции, то аналогично потребуется интегрирование функции плотности вероятности. Такая логика приводит нас к следующему решению, позволяющему избежать прямого взаимодействия с дельта-функцией — это переход от функции плотности вероятности к кумулятивной функции распределения. Этот подход позволит сохранить аналитическую простоту, избегая при этом сложностей, связанных с численной аппроксимацией дельта-функции.

Кумулятивная функция распределения для нашей модели определяется следующим образом:

$$F(t, X, \sigma, J; t', \tilde{X}, \tilde{\sigma}, \tilde{J}) := Prob\left[\, X(t') \leq \tilde{X}; \sigma(t') \leq \tilde{\sigma}; J(t') \leq \tilde{J} \,\right] =$$

$$= \int\limits_{-\infty}^{\tilde{X}} \int\limits_{-\infty}^{\tilde{\sigma}} \int\limits_{-\infty}^{\tilde{J}} p(t, X, \sigma, J; t', X', \sigma', J') dX' d\sigma' dJ',$$

где соответственно $Prob$ – вероятность.

Записать уравнение Фоккера-Планка для кумулятивной функции распределения не представляется возможным, так как оно справедливо исключительно для функции плотности вероятности. В связи с этим целесообразно перейти к уравнению Фейнмана-Каца (также известному как обратное уравнение Колмогорова), применимому к нашей модели:

Теорема 3. Формула Фейнмана-Каца для кумулятивной функции распределения в нашей модели записывается следующим образом:

$$\frac{\partial F}{\partial t} + \mu \frac{\partial F}{\partial X} + \lambda J \frac{\partial F}{\partial X} + \lambda J^2 \frac{\partial^2 F}{\partial X^2} + \kappa(\theta - \sigma) \frac{\partial F}{\partial \sigma} + \\ + \frac{1}{2}\sigma^2 \frac{\partial^2 F}{\partial X^2} + \frac{1}{2}\gamma^2 \sigma^2 \frac{\partial^2 F}{\partial \sigma^2} + \frac{1}{2}\varepsilon^2 \frac{\partial^2 F}{\partial X^J} + \rho\gamma\sigma^2 \frac{\partial^2 F}{\partial X \partial \sigma} = 0$$

Доказательство. Выпишем еще раз формулу Ито для нашей модели из Теоремы 1, подставив в нее кумулятивную функцию распределения:

$$dF(t, X, \sigma, J) = \left[\frac{\partial F}{\partial t} + \mu \frac{\partial F}{\partial X} + \lambda J \frac{\partial F}{\partial X} + \lambda J^2 \frac{\partial^2 F}{\partial X^2} + \kappa(\theta - \sigma) \frac{\partial F}{\partial \sigma} + \right. \\ \left. + \frac{1}{2}\sigma^2 \frac{\partial^2 F}{\partial X^2} + \frac{1}{2}\gamma^2 \sigma^2 \frac{\partial^2 F}{\partial \sigma^2} + \frac{1}{2}\varepsilon^2 \frac{\partial^2 F}{\partial X^J} + \rho\gamma\sigma^2 \frac{\partial^2 F}{\partial X \partial \sigma} \right] dt + \\ + \sigma \frac{\partial F}{\partial X} dW_t^X + \gamma\sigma \frac{\partial F}{\partial \sigma} dW_t^\sigma + \varepsilon \frac{\partial F}{\partial J} dW_t^J + \\ + \left[\lambda J \frac{\partial F}{\partial X} + \lambda J^2 \frac{\partial^2 F}{\partial X^2} \right] d\tilde{N}_t$$

В силу того, что функция $F(t, X, \sigma, J)$ является мартингалом (доказательство есть в учебнике Шрива [8]) и учитывая, что процессы W_t^X, W_t^σ, W_t^J, $\tilde{N}_t$ являются мартингалами, множитель при dt должен обнуляться. Это условие непосредственно приводит к уравнению Фейнмана-Каца, представленному в формулировке теоремы. □

При этом необходимо учитывать граничное условие. В отличие от уравнения Фоккера-Планка, уравнение Фейнмана-Каца имеет обратный ход, то есть граничное условие в уравнении Фейнана-Каца имеет противоположную границу, то есть для точки T при $\tilde{t} \leq t \leq T$. Поэтому

граничное условие запишется в следующей форме:

$$F(T, X, \sigma, J; T, \tilde{X}, \tilde{\sigma}, \tilde{J}) = \int_{-\infty}^{\tilde{X}} \int_{-\infty}^{\tilde{\sigma}} \int_{-\infty}^{\tilde{J}} \delta(X - X') \cdot \delta(\sigma - \sigma') \cdot \delta(J - J') dX' d\sigma' dJ'.$$

А значение этого интеграла мы уже можем выписать в явном виде, воспользовавшись определением дельта-функции. Получим

$$F(T, X, \sigma, J; T, \tilde{X}, \tilde{\sigma}, \tilde{J}) = \mathbb{I}(X \leq \tilde{X}) \cdot \mathbb{I}(\sigma \leq \tilde{\sigma}) \cdot \mathbb{I}(J \leq \tilde{J}),$$

где $\mathbb{I}$ – индикаторная функция: функция равна 1, если индикаторное условие выполнено, иначе равна нулю. Так, мы перешли от дельта-функции к индикаторной функции, которую можно задать явным способом без каких-либо аппроксимаций

Таким образом, мы пришли к следующей задаче: необходимо найти решение для следующего уравнения в частных производных на интервале $\tilde{t} \leq t \leq T$ со следующими граничными условием:

$$\frac{\partial F}{\partial t} + \mu \frac{\partial F}{\partial X} + \lambda J \frac{\partial F}{\partial X} + \lambda J^2 \frac{\partial^2 F}{\partial X^2} + \kappa(\theta - \sigma) \frac{\partial F}{\partial \sigma} + \\ \frac{1}{2}\sigma^2 \frac{\partial^2 F}{\partial X^2} + \frac{1}{2}\gamma^2 \sigma^2 \frac{\partial^2 F}{\partial \sigma^2} + \frac{1}{2}\varepsilon^2 \frac{\partial^2 F}{\partial X^J} + \rho\gamma\sigma^2 \frac{\partial^2 F}{\partial X \partial \sigma} = 0$$

$$F(T, X, \sigma, J; T, \tilde{X}, \tilde{\sigma}, \tilde{J}) = \mathbb{I}(X \leq \tilde{X}) \cdot \mathbb{I}(\sigma \leq \tilde{\sigma}) \cdot \mathbb{I}(J \leq \tilde{J}).$$

После того как будет найдено численное решение уравнения для кумулятивной функции распределения, значение искомой функции плотности вероятности можно будет вычислить методом численного дифференцирования. Это достигается с помощью разностной схемы для третьей производной по параметрам $\tilde{X}, \tilde{\sigma}, \tilde{J}$. Данный подход позволяет получить значение функции плотности вероятности, что и является нашей конечной целью.

4. Численное решение уравнения Фейнмана-Каца

Как упоминалось ранее, стандартные численные методы для решения нашего уравнения не подходят из-за проблемы, известной как «проклятие размерности». По этой причине для решения уравнения мы будем использовать нейронные сети. Данный подход является оптимальным, поскольку позволяет работать с высокоразмерными задачами и не требует фиксированного разбиения пространства. В частности, мы будем применять методологию Deep Galerkin Method (DGM), предложенную Sirignano и Spiliopoulos [6], которая также использована в работе [3]. Мы

будем следовать оригинальной модели, описанной в [6], с небольшими модификациями.

Скорость и качество обучения нейронных сетей зависят от специфики задачи, поскольку каждая структура сети оптимизирована для определённого класса задач. Например, сверточные нейронные сети (CNN) наиболее эффективны для задач распознавания изображений, в то время как сети типа LSTM (Long Short-Term Memory) лучше подходят для работы с временными рядами, где требуется моделирование прогнозов или решение задач оптимизации. В нашем случае мы воспользуемся модифицированной структурой нейронной сети, предложенной Sirignano и Spiliopoulos [6]. В своей работе они отмечают, что данная структура, являющаяся адаптацией LSTM, не только хорошо работает с последовательными данными, но и эффективно справляется с функциями, демонстрирующими «резкие повороты» в результате наложения граничных условий. Это особенно важно для нашей модели, где граничное условие представлено индикаторной функцией, имеющей ступенчатую форму.

Выбранная структура сетей описывается в виде следующей рекуррентной формы:

$$S^1 = \tanh\left(W^1\vec{x} + b^1\right),$$

$$Z^\ell = \tanh\left(U^{z,\ell}\vec{x} + W^{z,\ell}S^\ell + b^{z,\ell}\right), \quad \ell = 1,\dots,L$$

$$G^\ell = \tanh\left(U^{g,\ell}\vec{x} + W^{g,\ell}S^1 + b^{g,\ell}\right), \quad \ell = 1,\dots,L$$

$$R^\ell = \tanh\left(U^{r,\ell}\vec{x} + W^{r,\ell}S^\ell + b^{r,\ell}\right), \quad \ell = 1,\dots,L,$$

$$H^\ell = \tanh\left(U^{h,\ell}\vec{x} + W^{h,\ell}\left(S^\ell \odot R^\ell\right) + b^{h,\ell}\right), \quad \ell = 1,\dots,L,$$

$$S^{\ell+1} = \left(1 - G^\ell\right) \odot H^\ell + Z^\ell \odot S^\ell, \quad \ell = 1,\dots,L,$$

$$f(t,x;\Theta) = WS^{L+1} + b,$$

где $\vec{x}$ – вектор входных параметров для нашего уравнения, L – количество скрытых слоев, M – количество узлов в слое, D – количество входных параметров, $\odot$ – операция покомпонетного умножения векторов. Параметры модели обозначим за Θ, где

$$\Theta = \Big\{W^1, b^1, \left(U^{z,\ell}, W^{z,\ell}, b^{z,\ell}\right)_{\ell=1}^{L}, \left(U^{g,\ell}, W^{g,\ell}, b^{g,\ell}\right)_{\ell=1}^{L},$$
$$\left(U^{r,\ell}, W^{r,\ell}, b^{r,\ell}\right)_{\ell=1}^{L}, \left(U^{h,\ell}, W^{h,\ell}, b^{h,\ell}\right)_{\ell=1}^{L}, W, b\Big\}.$$

Параметры в Θ обладают следующими размерностями:

$$W^1 \in \mathbb{R}^{M\times(D+1)},\ b^1 \in \mathbb{R}^M,$$
$$U^{z,\ell} \in \mathbb{R}^{M\times(D+1)},\ W^{z,\ell} \in \mathbb{R}^{M\times M},\ b^{z,\ell} \in \mathbb{R}^M,$$
$$U^{g,\ell} \in \mathbb{R}^{M\times(D+1)},\ W^{g,\ell} \in \mathbb{R}^{M\times M},\ b^{g,\ell} \in \mathbb{R}^M,$$
$$U^{r,\ell} \in \mathbb{R}^{M\times(D+1)},\ W^{r,\ell} \in \mathbb{R}^{M\times M},\ b^{r,\ell} \in \mathbb{R}^M,$$
$$U^{h,\ell} \in \mathbb{R}^{M\times(D+1)},\ W^{h,\ell} \in \mathbb{R}^{M\times M},\ b^{h,\ell} \in \mathbb{R}^M,$$
$$W \in \mathbb{R}^{1\times M},\ b \in \mathbb{R}.$$

Следует отметить, что в качестве функции активации выбран гиперболический тангенс. Это обусловлено тем, что данная функция является гладкой, и для нашей задачи важно, чтобы функция распределения также обладала свойством гладкости. Перечисленные в Θ параметры будут инициализироваться однородным распределением по Ксавье. Для дальнейшей работы были выбраны следующие параметры: $L = 4, M = 64$.

Также закрепим, что в процессе обучения будут оптимизироваться не только переменные $t, X, \sigma, J, \tilde{X}, \tilde{\sigma}, \tilde{J}$, но и все остальные параметры модели: $\mu, \kappa, \theta, \gamma, \varepsilon, \rho, \lambda$. Таким образом, входной вектор будет содержать 14 элементов, то есть $D = 14$.

Теперь необходимо сформулировать функцию потерь. Следует использовать такую функцию, которая учитывает как соответствие самому уравнению, так и выполнение граничных условий. Определим её в общем виде для некоторой функции $u = u(t, \vec{x})$, предполагая, что имеем уравнение в частных производных следующего вида:

$$\frac{\partial u}{\partial t}(t, \vec{x}) + \mathcal{L}u(t, \vec{x}) = 0, \quad (t, \vec{x}) \in [t_0, T] \times \Omega,$$

$$u(t = T, \vec{x}) = u_T(\vec{x}).$$

где $\mathcal{L}$ – некий дифференциальный оператор, действующий на переменную $\vec{x}$, Ω – область определения для переменной $\vec{x}$, а граничное условие в момент времени T задается некоторой функцией u_T. Тогда определим функцию потерь следующим образом:

$$Loss_1(f) = \left\| \frac{\partial f}{\partial t}(t, \vec{x}; \Theta) + \mathcal{L}f(t, \vec{x}; \Theta) \right\|^2_{[0,T]\times\Omega,\nu},$$

$$Loss_2(f) = \|f(u, \vec{x}; \Theta) - g(t, \vec{x})\|^2_{[0,T]\times\partial\Omega,\nu},$$
$$Loss(f) = \alpha Loss_1(f) + Loss_2(f).$$

В данной формулировке суть заключается в том, что мы стремимся приблизить функцию f, чтобы минимизировать отклонение от истинного значения функции. Аналогичная логика применяется для функции g и граничного условия. В конструкции функции потерь для обеих компонент используется норма в пространстве L^2 по распределению ν, то есть

$$\|f(y)\|^2_{\Lambda,\nu} = \int_\Lambda |f(y)|^2 \nu(y) dy,$$

где $\nu(y)$ – функция плотности распределения, которое соответствует распределению, по которому инициализируются параметры, а Λ – область, в которой задаются переменные. Это делается для того, чтобы учесть концентрацию сгенерированных значений в нужных областях. Однако в нашем случае параметры инициализируются с равномерным распределением, поэтому плотность рассматривается как некоторая константа. Кроме того, в модель был добавлен параметр α, который отвечает за вес ошибки дифференциального оператора относительно ошибки, связанной с граничным условием. Согласно работе [3], оптимальное значение для параметра α равно 100, однако мы будем использовать значение $\alpha = 128$, чтобы сохранить пропорцию между количеством узлов в слоях, что также влияет на размер выборки для алгоритма.

Теперь, собрав все элементы воедино, представим следующий алгоритм для численного решения:

1) На каждом n-ом шаге генерируется батч из M многомерных точек $(t_n, \vec{x}_n)$, которые инициализируются в области $[0, T] \times \Omega$.

2) Для каждой такой выборки вычисляется средня квадратичная ошибка на основе построенной ранее функции потерь:

$$MSE\left(\Theta_n, \vec{x}_n, t_n\right) = \frac{\alpha}{M} \sum_{i=1}^{M} \left(\frac{\partial}{\partial t} f\left(t_{ni}, \vec{x}_{ni}; \Theta_n\right) + \mathcal{L} f\left(t_{ni}, \vec{x}_{ni}; \Theta_n\right) \right)^2 + \\ + \frac{1}{M} \sum_{i=1}^{M} \left(f\left(T, \vec{x}_{ni}; \Theta_n\right) - u_T\left(\vec{x}_{ni}\right) \right)^2$$

3) На основе функции среднеквадратичной ошибки, по точкам $(t_n, \vec{x}_n)$ обновляются параметры, заложенные в Θ. Для обновления параметров используется алгоритм адаптивной оценки для моментов (ADAM) с темпом обучения $lr = 0.0001$

4) Все описанные выше шаги выполняются итеративно до тех пор, пока значение ошибки не достигнет установленного минимального порога, который в нашем случае равен $e = 0.0001$.

Теперь необходимо определить, какие параметры будут подаваться на вход алгоритма. Поскольку в дальнейшем планируется калибровка коэффициентов модели, нас не устраивает обучение на фиксированных коэффициентах, как это предполагается в уравнениях. Таким образом, на вход, помимо дифференцируемых параметров $X, \sigma, J, \tilde{X}, \tilde{\sigma}, \tilde{J}$, мы также подадим постоянные коэффициенты модели: $\mu, \kappa, \theta, \gamma, \varepsilon, \rho, \lambda$. Эти коэффициенты будут инициализированы в пределах выбранных нами интервалов. Определим следующие диапазоны для данных параметров:

$$X, \tilde{X} \in [-7.0, 7.0]; \quad \sigma, \tilde{\sigma} \in [0.0, 2.0]; \quad J, \tilde{J} \in [-2.0, 2.0];$$
$$\mu \in [-1.0, 1.0]; \quad \kappa \in [0.5, 1.5]; \quad \theta \in [0.2, 1.1]; \quad \gamma \in [0.0, 0.8];$$
$$\varepsilon \in [0.0, 1.0]; \quad \rho \in [-0.5, 0.5]; \quad \lambda \in [0.0, 3.0]; \quad t \in [0.0, 1.0].$$

Модель запускается на 100,000 эпох. Конечно, ввиду большого количества изменяющихся параметров, следовало бы выбрать большее количество итераций, что улучшило бы качество прогнозов. Однако, исходя из ограничений моих вычислительных ресурсов, которые позволяют выполнить 100,000 итераций за приблизительно 27 часов, было принято решение остановиться на данном количестве эпох. Этого оказалось достаточно для получения адекватных результатов.

Для многомерной задачи подобного рода сложно представить визуализацию функции плотности распределения. Тем не менее, мы можем, как минимум, построить тепловую карту (heatmap) и визуализировать изменения плотности «сверху».

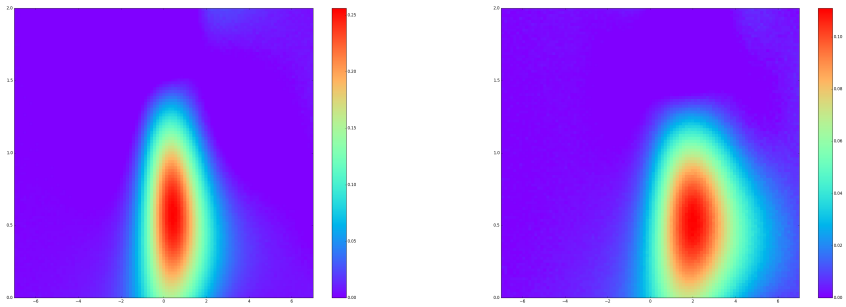


Рисунок 3: Heatmap-ы для вычисленной функции плотности вероятности. Оси x соответствует величина $\tilde{X}$, оси y – величина $\tilde{\sigma}$. Параметры рисунка слева: $X = 0.25$, $\sigma = 0.4$, $J = 0.5$, $\tilde{J} = 0.9$, $\mu = 0.2$, $\kappa = 0.3$, $\theta = 0.4$, $\gamma = 0.3$, $\epsilon = 1.0$, $\rho = -0.25$, $\lambda = 0.25$; параметры рисунка слева: $t = 0.0$, $X = 1.0$, $\sigma = 0.4$, $J = 1.0$, $\tilde{J} = 0.9$, $\mu = 0.2$, $\kappa = 0.5$, $\theta = 0.3$, $\gamma = 0.3$, $\epsilon = 0.6$, $\rho = 0.3$, $\lambda = 0.1$;

На Рисунке 3 видно, что если «разрезать» графики, то есть зафиксировать одну из осевых переменных, полученная функция демонстрирует

поведение, характерное для функции плотности распределения. Также следует отметить, что перед запуском оптимизатора для нашей модели была проведена проверка на простом примере стандартного геометрического броуновского движения. В этом тестовом случае оптимизатор успешно справился с решением уравнения Фейнмана-Каца. Таким образом, исходя как из визуальной проверки, так и из проверки на более простой модели, можно считать результаты достоверными.

Таким образом, этап решения уравнения Фейнмана-Каца был успешно завершён, что позволяет нам переходить к калибровке коэффициентов модели.

5. Калибровка параметров и сравнение с моделью Бэйтса

Пройдя сложный путь поиска решения уравнения Фейнмана-Каца, логично воспользоваться полученной плотностью распределения для калибровки параметров модели. Для этого мы применим один из наиболее надёжных методов математической статистики, специально предназначенный для поиска оптимальных параметров модели — метод максимального правдоподобия. Суть данного метода заключается в том, что при наличии выборки мы вычисляем функцию правдоподобия, представляющую собой кумулятивное произведение условных вероятностей, после чего находим её максимальное значение. Аргументы, при которых достигается максимальное значение функции, будут являться оптимальными оценками для модели. Однако к функции правдоподобия необходимо относиться с особой осторожностью.

Калибровка параметров будет проводиться для каждого токена по отдельности. Напомним, что у нас имеются ежеминутные данные за один день, что даёт временные ряды для цен и волатильностей на $1440 - 15 = 1425$ наблюдений (вычитаем 15, так как волатильность вычисляется по 15-минутному окну). В данной методологии мы будем исходить из того, что волатильность, оценённая как стандартное отклонение по скользящему окну, хотя и не является абсолютно точной оценкой, но даёт достаточно адекватные результаты. Мы разделим данные на обучающую выборку (1000 наблюдений) и тестовую выборку (оставшиеся 425 наблюдений). Обучающая выборка будет использована для калибровки постоянных коэффициентов модели $\mathbf{z} = (\mu, \kappa, \theta, \gamma, \varepsilon, \rho, \lambda)$, а тестовая — для фиксации оптимальных коэффициентов и сравнения модели с моделью Бэйтса.

Перед построением функции правдоподобия необходимо отметить следующее: изначально уравнение Фейнмана-Каца решалось для кумулятивной функции распределения с введением дополнительных

параметров $\tilde{t}, \tilde{X}, \tilde{\sigma}, \tilde{J}$ для существующих переменных t, X, σ, J. Эта функция по определению описывает вероятность того, что, находясь в момент времени t с определёнными значениями X, σ, J, к моменту времени $\tilde{t}$ значения переменных не превысят $\tilde{X}, \tilde{\sigma}, \tilde{J}$. Однако при переходе от функции распределения к функции плотности вероятности интерпретация переменных с тильдами меняется: мы переходим от вероятности не превысить определённые значения к вероятности достижения этих значений. Таким образом, функция $p(t, X, \sigma, J; \tilde{t}, \tilde{X}, \tilde{\sigma}, \tilde{J})$ выражает вероятность того, что переменные X, σ, J в момент времени t примут значения $\tilde{X}, \tilde{\sigma}, \tilde{J}$ к моменту времени $\tilde{t}$. Это важно учитывать при дальнейшем анализе.

Таким образом, функция правдоподобия может быть записана в следующем виде:

$$LH(\mathbf{z}) = \prod_{i=1}^{n} p\big[(X_i, \sigma_i, J_i) \mid \big\{(X_k, \sigma_k, J_k)\big\}_{k=0}^{i-1};\ \mathbf{z}\big],$$

где соответственно $p\big[(X_i, \sigma_i, J_i) \mid \big\{(X_k, \sigma_k, J_k)\big\}_{k=0}^{i-1};\ \mathbf{z}\big]$ – условная вероятность: вероятность того, что лог-цена, волатильность и размер скачка в момент времени i будут равны (X_i, σ_i, J_i) при условии, что те же параметры в предыдущие моменты времени равнялись соответственно $\big\{(X_k, \sigma_k, J_k)\big\}_{k=0}^{i-1}$. Всё это предполагается при фиксированных значениях постоянных параметров модели, $\mathbf{z}$. В силу того, что функция плотности вероятности была построена только для единичного временного интервала и что лучшей оценкой для переменной в момент времени i является та же переменная в момент времени $i-1$ (в силу мартингального свойства), будет корректно записать функцию правдоподобия для нашего случая таким образом:

$$LH(\mathbf{z}) = \prod_{i=1}^{n} p\big[(X_i, \sigma_i, J_i) \mid (X_{i-1}, \sigma_{i-1}, J_{i-1});\ \mathbf{z}\big],$$

Однако следует напомнить, что граничное условие для уравнения Фейнмана-Каца задается на противоположной границе временного интервала, что приводит к обратному ходу времени, начиная с момента T. В связи с этим возникает необходимость «перевернуть» функцию правдоподобия, учитывая данный обратный ход:

$$LH(\mathbf{z}) = \prod_{i=1}^{n} p\big[(X_{i-1}, \sigma_{i-1}, J_{i-1}) \mid (X_i, \sigma_i, J_i);\ \mathbf{z}\big],$$

то есть в соответствии с граничным условием для обратного уравнения Колмогорова, мы должны рассматривать вероятность параметров в

прошлом при условии известных значений в будущем, при этом подразумевается, что постоянные параметры модели заданы вектором $\mathbf{z}$).

Теперь необходимо разобраться с определением оставшихся переменных. Очевидно, что в качестве временного параметра мы выбираем $t = 0$. Процесс X_t является наблюдаемым, поэтому значения X_{i-1} и X_i мы получаем напрямую из тренировочных данных по токенам. Что касается процесса волатильности, на данном этапе мы не производили новых вычислений, поэтому значения σ_{i-1} и σ_i берем из ранее вычисленных волатильностей, полученных с помощью стандартного отклонения цен по скользящему окну. Однако процесс J_t вызывает определённые трудности, так как не вполне очевидно, как корректно выделить скачки исходя из динамики цены.

Мы можем сделать упрощённое предположение, что значения J_{i-1} и J_i равны нулю (предполагая, что в среднем скачки не наблюдаются). Однако в этом случае теряется смысл введения скачков в модель, если мы избегаем их учета. Поэтому предлагается следующая аппроксимация: предположим, что значение скачка в момент времени $i-1$ вычисляется как разница между логарифмическими ценами X_{i-1} и X_i. Идея заключается в том, чтобы задать допустимые пороговые значения изменения лог-цены (симметричные относительно нуля, так как скачки в нашей модели распределены нормально с нулевым средним). Если изменение лог-цены превышает порог, это будет интерпретироваться как активация скачка в соответствующий момент времени. В таком случае значение скачка будет равно разнице между ближайшей границей и фактическим изменением лог-цены. Если же порог не был превышен, то скачка не было, и значение J_{i-1} равно нулю.

Почему мы можем поступить таким образом? В данной модели и в других временных рядах изменения лог-цен величины скачков могут быть настолько малы, что их невозможно выделить из динамики лог-цены. Для нашей текущей задачи идентифицировать малые скачки практически невозможно, но это не является нашей основной целью. Мы стремимся моделировать экстремальные изменения цены, а малые скачки являются побочным эффектом моделирования. Поэтому целесообразно разделить все скачки на наблюдаемые и ненаблюдаемые, сосредоточив внимание на крупных скачках, а относительно малые значения можно игнорировать без существенной потери точности.

Для моделирования наблюдаемых скачков, как уже было отмечено, необходимо определить границы допустимых изменений лог-цены. Это можно сделать либо вручную, анализируя графики, либо задать границы в виде формулы. Если задавать границы в виде формулы, нам необходимо учитывать зависимость от волатильности. Однако, поскольку волатильность аппроксимирована на основе самой лог-

цены, скачок лог-цены вызовет скачок волатильности, что приведет к некорректному заданию границ. Поэтому более разумно задать фиксированный граничный интервал $[-b,\ b]$, где $b > 0$ — параметр, который может варьироваться для разных токенов.

Таким образом, модель значений скачков будет записана следующим уравнением:

$$J_{i-1} = \operatorname{sign}(\Delta X_{i-1}) \cdot \max\big\{|\Delta X_{i-1}| - b, 0\big\},$$

где $\Delta X_{i-1} = X_{i-1} - X_i$ – приращение логарифмической цены в момент времени i. Таким образом, если абсолютное значение изменения лог-цены не превышает заданную границу, то максимум второго множителя достигается при нулевом значении скачка, в результате чего данный скачок классифицируется как несущественный, и его значение обнуляется (в упомянутом ранее предположении, что несущественный скачок является частью процесса X_t). В противном случае, при пересечении границы, с помощью функции знака определяется направление скачка.

Прежде чем перейти к обсуждению реализации оптимизатора, необходимо учесть следующее: поскольку мы работаем с относительно большой тренировочной выборкой и вероятностями, перемножение множества вероятностей порядка тысячи раз неизбежно приведет к обнулению функции правдоподобия. Это явление называется underflow, и его необходимо избежать. Для этого мы перейдём от обычной функции правдоподобия к её логарифму: благодаря монотонности логарифма, оптимальные решения для исходной и логарифмической функций будут достигаться при одном и том же наборе параметров. Более того, мы сразу перейдём к отрицательной функции правдоподобия, поскольку в рамках кода задача заключается в минимизации, а не максимизации. Таким образом, мы будем рассматривать следующую отрицательную логарифмическую функцию правдоподобия:

$$LLH^{-}(\mathbf{z}) = \sum_{i=1}^{n} -\ln p\big[(X_{i-1}, \sigma_{i-1}, J_{i-1}) \mid (X_i, \sigma_i, J_i);\ \mathbf{z}\big].$$

Наконец, можно перейти к описанию реализации оптимизатора. Здесь возникает серьёзная проблема: использование нейронных сетей для поиска решения переходной функции плотности вероятности (в нашем случае, с использованием библиотеки PyTorch для языка Python) накладывает ограничения на использование известных библиотек для оптимизации, таких как SciPy или OR-Tools. Это связано с тем, что PyTorch использует свои собственные тензоры для представления данных, которые не воспринимаются сторонними библиотеками. Более того, такие библиотеки не поддерживают концепции PyTorch, такие как

вычисление градиентов, что делает невозможным запуск оптимизатора на основе этих инструментов. Можно было бы рассмотреть метод Монте-Карло, его использование для задач оптимизации не является корректным. Данный метод лучше подходит для задач, где приближение к решению достигается за счёт большого количества симуляций, таких как нахождение интегралов или оценка средних значений. Однако в задаче поиска оптимальной точки на гиперплоскости размерности 7 метод Монте-Карло неэффективен, поскольку его использование можно сравнить с поиском иголки в стоге сена. В результате остаётся единственный вариант — использовать библиотеку PyTorch для решения задачи оптимизации.

Однако и здесь возникает проблема: модели оптимизации в PyTorch не позволяют задать явные границы для оптимизируемых параметров. Поскольку других подходящих инструментов нет, необходимо нивелировать эту проблему путём введения дополнительных условий. Мы решаем это путём включения штрафов за выход за границы в нашу отрицательную логарифмическую функцию правдоподобия, которая будет одновременно выступать в роли функции потерь для оптимизатора.

Важно отметить, что параметры будут калиброваться в пределах тех же интервалов, которые прежде использовались при решении параметрического уравнения Фейнмана-Каца. Если параметры выйдут за эти границы, поведение переходной функции плотности вероятности станет непредсказуемым, что негативно скажется на процессе оптимизации. Выпишем ещё раз эти интервалы:

$$\mu \in [-1.0, 1.0]; \quad \kappa \in [0.5, 1.5]; \quad \theta \in [0.2, 1.1]; \quad \gamma \in [0.0, 0.8];$$

$$\varepsilon \in [0.0, 1.0]; \quad \rho \in [-0.5, 0.5]; \quad \lambda \in [0.0, 3.0].$$

Также введем следующие обозначения: $\mathbf{c_{min}}, \mathbf{c_{max}}$ – вектора минимальных и максимальных границ для вектора параметров $\mathbf{z}$.

Теперь вернёмся к введению штрафов. Идея заключается в том, чтобы добавлять штрафы к функции потерь при выходе параметров за установленные границы. Построим штрафную функцию следующим образом:

$$Penalty(\mathbf{z_n}) = \pi \left|\left| \max\{\mathbf{z_n} - \mathbf{c_{max}}; \mathbf{0}\} + \max\{\mathbf{c_{min}} - \mathbf{z_n}; \mathbf{0}\} \right|\right|_1,$$

где $\mathbf{z_n}$ – вектор параметров на n-ой итерации, $\mathbf{0}$ – нулевой вектор, операция максимума берется покомпонентно, $\|\cdot\|_1$ – норма в пространстве l_1, которая вычисляется как сумма абсолютных значений компонент вектора, π – штрафной коэффициент, который регулирует силу штрафа. Так, предложенная штрафная функция показывает, насколько сильно алгоритм отдалился от границы; если же вектор параметров находится

внутри установленных границ, то размер штрафа равен нулю. Далее в качестве штрафного коэффициента будем использовать $\pi = 10$.

В итоге осталось объединить нашу отрицательную логарифмическую функцию правдоподобия со штрафной функцией. Сделав это, получим окончательную форму функции потерь:

$$Loss(\mathbf{z_n}) = \sum_{i=1}^{n} - \ln p\big[(X_{i-1}, \sigma_{i-1}, J_{i-1}) \mid (X_i, \sigma_i, J_i);\ \mathbf{z_n}\big] + Penalty(\mathbf{z_n}).$$

Для решения данной задачи мы применили алгоритм стохастического градиентного спуска (SGD) с темпом обучения $lr = 0.001$. В качестве начальных параметров $\mathbf{z_0}$ были взяты середины интервалов, задающих границы параметров: $\mathbf{z_0} = \frac{1}{2}(\mathbf{c_{max}} + \mathbf{c_{min}})$.

Теперь у нас есть все необходимое для сравнения результатов с моделью Бэйтса. Для этой модели были воспроизведены все шаги, выполненные для нашей модели. Единственное существенное отличие заключается в природе скачков: в то время как в процессе решения уравнения Фейнмана-Каца в нашей модели скачки выделены в отдельный процесс, в модели Бэйтса параметры логнормального распределения скачков являются одними из постоянных коэффициентов модели. Соответственно, величина скачка моделируется как случайная величина, распределённая по логнормальному закону с соответствующими параметрами.

В результате была составлена Таблица 1, содержащая итоговые результаты оптимизации логарифмической функции правдоподобия, включая максимальные значения функции для модели Бэйтса. Стоит отметить, что функция правдоподобия, помимо использования для калибровки коэффициентов модели, также служит показателем качества интерпретации рыночных процессов. Как было упомянуто ранее, функция правдоподобия строится как произведение условных вероятностей, что по сути отражает вероятность точного предсказания временного ряда. Таким образом, функция правдоподобия может интерпретироваться как ответ на вопрос: «насколько точно мы предсказываем следующее значение временного ряда?», а лучшая предсказательная способность, в свою очередь, указывает на лучшую интерпретацию актива или рынка, на котором он представлен. По результатам, представленным в таблице, можно заметить, что максимальные значения функции правдоподобия для нашей модели в среднем несколько выше. Однако, эта информация не даёт окончательной сравнительной оценки: необходимо проанализировать поведение функций правдоподобия на тестовой выборке с уже зафиксированными оптимальными коэффициентами.

	μ	κ	θ	γ	ε	ρ	λ	LLH_Our	LLH_Bates
LINK	-0.4792	1.0616	0.2172	0.1813	0.8323	0.4379	1.6761	-1439.5780	-1650.1839
BTC	-0.7391	1.2236	0.2046	0.1587	0.8148	0.1976	1.2809	-1537.4155	-1419.7142
CRV	-0.2301	1.1125	0.2354	0.4477	0.8766	0.4780	0.8428	-1567.0087	-1507.2024
ETH	-0.7755	1.1199	0.2104	0.0198	0.8490	0.1639	0.8450	-1591.1027	-1560.5366
UNI	-0.4072	1.1098	0.2451	0.7735	0.6434	-0.2716	1.5069	-1653.7539	-1744.5925
AAVE	-0.3200	1.0446	0.2675	0.3351	0.9205	0.1190	0.8333	-1666.4852	-1737.6021
MKR	-0.2887	0.7957	0.2032	0.5857	0.9971	0.0233	1.9902	-1482.6937	-1776.7665

Таблица 1. Итоговые значениоткалиброванных параметров на примере семи токенов в результате оптимизации логарифмической функции правдоподобия (первые 7 столбцов) и максимальные значения функции для нашей модели и для модели Бэйтса (последние 2 столбца)

Значения логарифмической функции правдоподобия являются отрицательными, поскольку в исходной версии функции правдоподобия (до применения логарифмов) мы перемножали вероятности, значения которых лежат в диапазоне от 0 до 1, следовательно, логарифмы этих вероятностей всегда будут отрицательными. Также стоит отметить, что наш оптимизатор выявил несколько интересных особенностей: например, известно, что токены BTC и ETH обладают относительно высокой корреляцией между собой, что проявилось как в схожести значений откалиброванных параметров, так и в максимальных значениях функции правдоподобия для этих активов. В целом, результаты выглядят удовлетворительными, но окончательные выводы можно будет сделать только после анализа на тестовой выборке при оптимальных параметрах.

После аналогичного построения Таблицы 2 для результатов на тестовой выборке с использованием откалиброванных параметров мы получили гораздо более обнадёживающие результаты.

	LLH_our	LLH_Bates
LINK	-894.723091	-1119.705687
BTC	-931.342194	-960.669007
CRV	-1139.196365	-1324.843547
ETH	-923.042812	-931.730988
UNI	-793.611184	-1181.945911
AAVE	-803.573028	-1099.465584
MKR	-734.850167	-727.403466
mean	-888.619834	-1049.394884

Таблица 2. Сравнение значений логарифмической функции правдоподобия на тестовой выборке для нашей модели и модели Бэйтса. Последняя строка в таблице отвечает за среднее значение функций для семи активов

В Таблице 2 наблюдаются более высокие значения функций правдоподобия, что объясняется меньшим размером тестовой выборки по сравнению с тренировочной. Тем не менее, по результатам этой таблицы видно, что в среднем наша модель гораздо лучше предсказывает поведение рынка по сравнению с моделью Бэйтса. Особого внимания заслуживает случай токена CRV, для которого были получены относительно более низкие значения функции правдоподобия. Это связано с тем, что на тестовой выборке было зарегистрировано множество скачков, которые, хотя и значимы, но не превышают ранее установленные границы для наблюдаемых скачков. В результате максимальное значение функции правдоподобия для CRV оказалось значительно ниже. Однако, несмотря на аналогичное моделирование скачков в модели Бэйтса, наша модель всё же демонстрирует более точную интерпретацию подобных случаев.

6. Заключение

В итоге нам удалось разработать новую модель ценообразования и выписать для неё систему стохастических уравнений, которая точно описывает поведение рынка криптоактивов в сравнении с моделью Бэйтса, имеющей схожую структуру. В данной работе мы не только учли такие ключевые особенности, как логнормальное распределение волатильности, её возврат к среднему значению и наличие пуассоновских скачков, характерных для криптоактивов, но также предложили принципиально новую концепцию стохастической модели, в которой величина скачков описывается отдельным стохастическим процессом. Это позволяет более точно выделить данную компоненту в динамике изменения цен, что даёт как и гибкость в её моделировании, так и лучшую интерпретацию скачков. Кроме того, мы успешно составили и решили параметрическое уравнение Фейнмана-Каца с помощью нейронных сетей, что позволило нам точно определить значение функции плотности вероятности без привязки к разбиению. После получения функции плотности вероятности мы использовали метод максимального правдоподобия для калибровки модели на исторических данных, а затем провели сравнение результатов на тестовой выборке. Сравнение с такой сложной моделью, как модель Бэйтса, показало, что наша модель в среднем лучше интерпретирует динамику цены, что является важным достижением.

В дальнейшем данную модель можно усовершенствовать и применять не только к крипторынкам, но и к традиционным финансовым рынкам, таким как опционы или процентные ставки. В частности, модель можно обобщить, например, перейдя от обычного пуассоновского процесса к составному пуассоновскому процессу, что позволит более точно моделировать скачки. Также можно рассмотреть возможность добавления

корреляции между винеровским процессом, описывающим скачки, и винеровскими процессами, описывающими цену и волатильность. Кроме того, при составлении уравнения Фейнмана-Каца можно не прибегать к разложению разности функций со скачком и без него в ряд Тейлора до второй производной, что также может повысить точность модели.

Список литературы

[1] Bates, D., *Jumps and Stochastic Volatility: Exchange Rates Processes Implicit in Deutsche Mark Options*, The Review of Financial Studies 9, 1996.

[2] Cornelis W Oosterlee, Lech A Grzelak, *Mathematical Modeling and Computation in Finance*, World Scientific Publishing Europe Ltd, 2020.

[3] Haozhe Su, M.V. Tretyakov, David P. Newton, *Deep Learning of Transition Probability Densities for Stochastic Asset Models with Applications in Option Pricing*, arXiv:2105.10467, 2023.

[4] Heston, S., *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, The Review of Financial Studies 6, 1993.

[5] *Notes on Stochastic Finance, Chapter 20: Stochastic Calculus for Jump Processes*, Nanyang Technological University.

[6] Justin Sirignano, Konstantinos Spitiopoulos, *A deep learning algorithm for solving partial differential equations*, arXiv:1708.07469, 2018.

[7] S. G. Kou, *A Jump-Diffusion Model for Option Pricing*, Department of Industrial Engineering and Operations Research, Columbia University, New York, 2002.

[8] Steven E. Shreve, *Stochastic Calculus for Finance II, Continuous-Time Models*, Springer Science + Business Media, Inc., 2004.

Jump diffusion model with mean-reverting lognormal volatility

Chestnov R.V.

Currently, there are numerous stochastic models built for different market concepts. However, almost all of these models are based and tested for traditional markets, while at present the cryptoasset market is gaining tremendous momentum in terms of volume and market capitalisation. For this reason, the idea arises to create such a model,

which would be built specifically for the cryptoasset market, taking into account all its peculiarities and behavioural patterns. In this paper, we will try to look at some of the features that are noticeable for cryptoassets and build a stochastic model that takes them into account, and after that, we will compare it with another model that is similar in structure. It is notable right away that, similar to how traditional models are used for cryptomarkets, our model for cryptoassets will interpret the traditional market just as well. Moreover, even better than in the opposite case, because the main idea of our model is to take into account various critical events that occur with this or that asset, but by calibrating the model in the right way, we can ignore them.

Key words: Financial mathematics, stochastic calculus, cryptoassets, neural networks, Feynman-Kac equation, Bates model, likelihood function.

References

[1] Bates, D., *Jumps and Stochastic Volatility: Exchange Rates Processes Implicit in Deutsche Mark Options*, The Review of Financial Studies 9, 1996.

[2] Cornelis W Oosterlee, Lech A Grzelak, *Mathematical Modeling and Computation in Finance*, World Scientific Publishing Europe Ltd, 2020.

[3] Haozhe Su, M.V. Tretyakov, David P. Newton, *Deep Learning of Transition Probability Densities for Stochastic Asset Models with Applications in Option Pricing*, arXiv:2105.10467, 2023.

[4] Heston, S., *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, The Review of Financial Studies 6, 1993.

[5] *Notes on Stochastic Finance, Chapter 20: Stochastic Calculus for Jump Processes*, Nanyang Technological University.

[6] Justin Sirignano, Konstantinos Spitiopoulos, *A deep learning algorithm for solving partial differential equations*, arXiv:1708.07469, 2018.

[7] S. G. Kou, *A Jump-Diffusion Model for Option Pricing*, Department of Industrial Engineering and Operations Research, Columbia University, New York, 2002.

[8] Steven E. Shreve, *Stochastic Calculus for Finance II, Continuous-Time Models*, Springer Science + Business Media, Inc., 2004.

Часть 2
Специальные вопросы теории интеллектуальных систем

Математические основы прогнозирования временных рядов

А. М. Миронов[1]

В статье излагаются основные понятия и методы прогнозирования временны́х рядов. Рассматриваются различные алгоритмы смешивающего прогнозирования, и приводятся оценки качества этих алгоритмов.

Ключевые слова: временные ряды, алгоритмы прогнозирования, смешивающие алгоритмы прогнозирования

Введение

Основным объектом исследования в данной статье являются алгоритмы прогнозирования временных рядов (называемые ниже просто алгоритмами прогнозирования), которые основаны на следующей идее: пусть заданы несколько алгоритмов прогнозирования $A_1, \ldots, A_N$, искомый алгоритм прогнозирования A (называемый смешивающим алгоритмом) должен использовать результаты работы алгоритмов $A_1, \ldots, A_N$, качество прогнозирования смешивающего алгоритма A (т.е. доля правильных прогнозов этого алгоритма) должно быть близким к качеству наилучшего из алгоритмов $A_1, \ldots, A_N$.

Цель настоящей работы заключается в систематизации изложения основных подходов к смешивающему прогнозированию. Содержание работы имеет следующий вид. Сначала рассматриваются простейшие алгоритмы смешивающего прогнозирования: алгоритм большинства, алгоритм взвешенного большинства, алгоритм оптимального распределения потерь. Далее рассматриваются более сложные алгоритмы смешивающего прогнозирования: алгоритм следования за возмущённым лидером, агрегирующий алгоритм Вовка. Кроме того, рассматривается алгоритм усиления классификаторов (бустинг), природа которого сходна природе смешивающих алгоритмов. Затем рассматриваются понятие прогнозной стратегии и примеры детерминированной и вероятностной прогнозных стратегий. Содержание статьи основано на материале из книги [1], в настоящем тексте представлены новые, более простые доказательства соответствующих теорем из [1].

[1] *Миронов Андрей Михайлович* — доцент каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: amironov66@gmail.com.

Mironov Andrew Mikhaylovich — associate professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

1. Задача прогнозирования временны́х рядов

Под **временны́м рядом** понимается последовательность $y = (y_1, y_2, \ldots)$ элементов некоторого множества Y, называемых **исходами**. Мы будем рассматривать случай, когда $Y = \{0, 1\}$ или $[0, 1]$ (вместо 0 м.б. -1).

Прогнозируемый временной ряд y может быть бесконечным или иметь конечную длину, которую мы будем обозначать символом T, т.е. во втором случае $y = (y_1, \ldots, y_T)$.

Задача прогнозирования временного ряда y заключается в построении **прогнозирующего алгоритма (ПА)** A, который на каждом шаге прогнозирования $t = 1, 2, \ldots$ выдаёт значение $\gamma_t \in Y$, называемое **прогнозом** временного ряда y на шаге t. После того, как A выдал γ_t, становится известным значение $y_t \in Y$ **исхода** в момент t.

На каждом шаге прогнозирования t, который выполняет ПА A, определена **потеря** $l_t \in [0, 1]$ ПА A, связанная с несовпадением прогноза γ_t и исхода y_t. Будем считать, что $l_t = 0$ тогда и только тогда, когда $\gamma_t = y_t$. Как правило, потеря l_t является значением некоторой **функции потерь** λ на паре (γ_t, y_t), например $l_t = [\![\gamma_t \neq y_t]\!]$. Напомним, что для каждого логического утверждения φ запись $[\![\varphi]\!]$ обозначает число 1, если φ истинно, и 0, если φ ложно. Если $y = (y_1, \ldots, y_T)$, то будем называть **кумулятивной потерей** ПА A величину $L_T \stackrel{\text{def}}{=} \sum_{t=1}^{T} l_t$.

2. Смешивающее прогнозирование

Один из методов построения прогнозирующих алгоритмов заключается в следующем. Пусть имеется несколько ПА A_1, …, A_N. Для каждого $i = 1, \ldots, N$ в каждый момент времени $t = 1, \ldots, T$ ПА A_i выдаёт прогноз γ_t^i. Будем обозначать записью L_T^i кумулятивную потерю ПА A_i. Используя алгоритмы $A_1, \ldots, A_N$ можно построить ПА A, называемый **смешивающим** ПА. На каждом шаге прогнозирования t прогноз γ_t, выдаваемый алгоритмом A, определяется как некоторая функция от прогнозов $\gamma_t^1, \ldots, \gamma_t^N$, называемая **функцией смешивания**.

Алгоритмы $A_1, \ldots, A_N$, участвующие в определении смешивающего ПА A, будем называть **экспертами**. Экспертов можно сравнивать по качеству их прогнозов: эксперт A_i лучше эксперта A_j, если $L_T^i < L_T^j$. Будем обозначать экспертов просто их номерами $1, \ldots, N$, и множество экспертов $\{1, \ldots, N\}$ будем обозначать символом I.

Величина $R_T = L_T - \min_{i \in I} L_T^i$ называется **регретом** смешивающего ПА A. Данная величина выражает собой отличие кумулятивной потери ПА A от кумулятивной потери наилучшего эксперта. При построении смешивающих ПА функция смешивания должна выбираться так, чтобы регрет смешивающего ПА был как можно меньше.

3. Алгоритм большинства

В этом пункте излагается простейший смешивающий ПА, который может использоваться лишь в ситуации, когда среди экспертов из множества $I = \{1, \ldots, N\}$ существует эксперт i_0, в каждый момент времени выдающий правильный прогноз, т.е. такой, что $\forall\, t \geq 1 \quad \gamma_t^{i_0} = y_t$. Этот алгоритм называется **алгоритмом большинства (Majority Algorithm, MA)**. Прогнозы данного ПА определяются следующим образом:

$$\forall\, t \geq 1 \quad \gamma_t := [\![|\{i \in B_t \mid \gamma_t^i = 1\}| \geq \tfrac{|B_t|}{2}]\!], \tag{1}$$

где для каждого конечного множества X запись $|X|$ обозначает число элементов в X, и $\forall\, t \geq 1$ множество B_t определяется следующим образом:

$$B_t = \{i \in I \mid \forall\, t' = 1, \ldots, t-1 \;\; \gamma_{t'}^i = y_{t'}\}. \tag{2}$$

Будем говорить, что ПА A **делает ошибку** на шаге t, если $\gamma_t \neq y_t$.

Теорема 1.
MA делает не более $\log_2 N$ ошибок.

Доказательство.
Нетрудно видеть, что последовательность множеств (2) обладает свойством $B_1 \supseteq B_2 \supseteq \ldots$, и каждое из множеств B_t непусто, т.к. $i_0 \in B_t$.
Если MA делает ошибку на шаге t, т.е. $\gamma_t \neq y_t$, то

- либо $\gamma_t = 1$ и $y_t = 0$, в этом случае, согласно (1),

$$|\{i \in B_t \mid \gamma_t^i = 1\}| \geq \tfrac{|B_t|}{2}, \tag{3}$$

  согласно (2), $B_{t+1} = \{i \in B_t \mid \gamma_t^i = 0\}$, и из (3) следует $|B_{t+1}| \leq \frac{|B_t|}{2}$,

- либо $\gamma_t = 0$ и $y_t = 1$, в этом случае, согласно (1),

$$|\{i \in B_t \mid \gamma_t^i = 1\}| < \tfrac{|B_t|}{2}, \tag{4}$$

  согласно (2), $B_{t+1} = \{i \in B_t \mid \gamma_t^i = 1\}$, и из (4) следует $|B_{t+1}| < \frac{|B_t|}{2}$.

В обоих случаях $|B_{t+1}| \leq \frac{|B_t|}{2}$.
Пусть MA делает k ошибок, и t – момент, в который делается k–я ошибка, тогда, по установленному выше,

$$N \geq |B_1| \geq 2^k |B_{t+1}|. \tag{5}$$

Учитывая $|B_{t+1}| \geq 1$, из (5) получаем $N \geq 2^k$, или $k \leq \log_2 N$. ■

4. Алгоритм взвешенного большинства

В этом пункте излагается смешивающий алгоритм, называемый **алгоритмом взвешенного большинства (Weighted Majority Algorithm, WMA)**, впервые он был изложен в работе [2]. Данный алгоритм может использоваться в том случае, когда среди экспертов из $I = \{1, \ldots, N\}$ может не быть эксперта, выдающего в каждый момент времени правильный прогноз.

В каждый момент времени t данный алгоритм сопоставляет каждому эксперту $i \in I$ некоторое число $w_t^i \in [0, 1]$, называемое **весом** этого эксперта. В начальный момент $t = 1$ вес каждого эксперта равен 1. Считаем, что потери имеют вид $l_t = |\gamma_t - y_t|$, $l_t^i = |\gamma_t^i - y_t|$.

Прогнозы данного ПА и изменения весов определяются следующим образом: выбирается параметр $\varepsilon \in (0, 1)$, и

$$\forall\, t = 1, \ldots, T \quad \begin{cases} \gamma_t := [\![\sum_{i:\gamma_t^i=0} w_t^i \le \sum_{i:\gamma_t^i=1} w_t^i]\!] \\ w_{t+1}^i := w_t^i(1 - \varepsilon l_t^i). \end{cases} \tag{6}$$

Теорема 2.
Для кумулятивных потерь WMA верно неравенство

$$L_T \le \tfrac{2}{1-\varepsilon} \min_{i \in I} L_T^i + \tfrac{2}{\varepsilon} \ln N \tag{7}$$

(т.е. WMA ошибается примерно не более чем в $\frac{2}{1-\varepsilon}$ раз, чем наилучший эксперт).

Доказательство.
Будем использовать следующие обозначения:

$$M = L_T, \quad m = \min_{i \in I} L_T^i, \quad |\vec{w}_t| = \sum_{i \in I} w_t^i.$$

Пусть i – номер наилучшего эксперта.
w_t^i корректируется $\le m$ раз, поэтому

$$|\vec{w}_T| \ge w_T^i \ge (1 - \varepsilon)^m. \tag{8}$$

Нетрудно проверить (это делается так же, как в доказательстве предыдущей теоремы), что если WMA делает ошибку на шаге t, то

$$\sum_{i:\gamma_t^i \ne y_t} w_t^i \ge \sum_{i:\gamma_t^i = y_t} w_t^i. \tag{9}$$

Прибавив к обеим частям (9) слагаемое $\sum_{i:\gamma_t^i \ne y_t} w_t^i$, получаем

$$\sum_{i:\gamma_t^i \ne y_t} w_t^i \ge \tfrac{|\vec{w}_t|}{2}. \tag{10}$$

Из (10) и из определения w^i_{t+1} в (6) следует, что если WMA делает ошибку на шаге t, то

$$|\vec{w}_{t+1}| = \sum_{i\in I} w^i_t(1-\varepsilon l^i_t) = \\ = |\vec{w}_t| - \varepsilon \sum_{i:\gamma^i_t \neq y_t} w^i_t \leq |\vec{w}_t|(1-\tfrac{\varepsilon}{2}),$$

т.е. если WMA делает ошибку на шаге t, то $\frac{|\vec{w}_{t+1}|}{|\vec{w}_t|} \leq 1-\frac{\varepsilon}{2}$.

Из определения весов в (6) следует, что для каждого $t = 1, \ldots, T-1$ $\frac{|\vec{w}_{t+1}|}{|\vec{w}_t|} \leq 1$. Следовательно,

$$\frac{|\vec{w}_T|}{|\vec{w}_1|} = \prod_{t=1}^{T-1} \frac{|\vec{w}_{t+1}|}{|\vec{w}_t|} \leq (1-\tfrac{\varepsilon}{2})^M,$$

откуда, учитывая (8) и равенство $|\vec{w}_1| = N$, получаем неравенство

$$\frac{(1-\varepsilon)^m}{N} \leq (1-\tfrac{\varepsilon}{2})^M,$$

логарифмируя которое, и учитывая неравенство

$$\ln(1+x) \leq x \quad \text{при } x \in (-1,1) \tag{11}$$

получаем:

$$m\ln(1-\varepsilon) - \ln N \leq M\ln(1-\tfrac{\varepsilon}{2}) \leq -\tfrac{\varepsilon}{2}M$$

откуда следует неравенство

$$\tfrac{\varepsilon}{2}M \leq m\ln\tfrac{1}{1-\varepsilon} + \ln N. \tag{12}$$

Применяя (11) для $x = \frac{\varepsilon}{1-\varepsilon}$, получаем соотношения

$$\ln\tfrac{1}{1-\varepsilon} = \ln(1+\tfrac{\varepsilon}{1-\varepsilon}) \leq \tfrac{\varepsilon}{1-\varepsilon} \tag{13}$$

Из (12) и (13) следует неравенство

$$\tfrac{\varepsilon}{2}M \leq m\tfrac{\varepsilon}{1-\varepsilon} + \ln N,$$

которое эквивалентно доказываемому неравенству (7). ■

5. Алгоритм оптимального распределения потерь

В этом пункте рассматривается другая постановка задачи построения смешивающего ПА: как и выше, задано множество экспертов $I = \{1, \ldots, N\}$, но для каждого шага прогнозирования t вместо прогнозов экспертов γ^i_t известны лишь потери $l^i_t \in [0,1]$, которые несут эксперты на шаге t. Требуется построить ПА, кумулятивные потери которого были бы как можно ближе к кумулятивным потерям наилучшего из этих экспертов (т.е. к $\min_{i\in I} L^i_T$).

Будем использовать понятие **вероятностного распределения (ВР)** на множестве $I = \{1, \ldots, N\}$, которое представляет собой произвольный вектор $\vec{p} = (p^1, \ldots, p^N)$ неотрицательных действительных чисел, удовлетворяющих условию $\sum_{i \in I} p^i = 1$. Множество всех ВР на I будем обозначать записью $I^{\triangle}$. Вектор из $I^{\triangle}$, все компоненты которого совпадают (т.е. равны $\frac{1}{N}$) будем называть **равномерно распредёленным (р.р.)**.

Также будем использовать следующее обозначение – если $\vec{w}$ – вектор неотрицательных действительных чисел вида $(w^1, \ldots, w^N)$, то $norm(\vec{w})$ – это ВР $(p^1, \ldots, p^N)$, где

$$\forall\, i \in I \;\; p^i = \frac{w^i}{|\vec{w}|}, \quad \text{где } |\vec{w}| = \sum_{i=1}^{N} w^i.$$

Предлагается следующее решение описанной выше задачи: на каждом шаге прогнозирования t определяется ВР $\vec{p}_t = (p_t^1, \ldots, p_t^N) \in I^{\triangle}$, и прогноз искомого алгоритма A на шаге t полагается равным прогнозу эксперта i, номер которого выбран из I случайным образом, в соответствии с распределением $\vec{p}_t$ (т.е. с вероятностью p_t^1 выбран эксперт 1, с вероятностью p_t^2 выбран эксперт 2, и т.д.). Потери ПА A в момент t совпадают с потерями выбранного эксперта i в момент t.

Нетрудно видеть, что математическое ожидание l_t потери ПА A в момент t совпадает со **скалярным произведением** $\langle \vec{p}_t, \vec{l}_t \rangle = \sum_{i \in I} p_t^i l_t^i$, где $\vec{l}_t = (l_t^1, \ldots, l_t^N)$.

$\forall\, t = 1, \ldots, T$ определяем $\vec{p}_t$ как $norm(\vec{w}_t)$, где

$$\begin{array}{l} \vec{w}_1 \text{ – р.р.} \\ \forall\, t = 1, \ldots, T-1, \forall\, i \in I \;\; w_{t+1}^i := w_t^i \beta^{l_t^i} \end{array} \tag{14}$$

где $\beta \in (0, 1)$ – параметр.

Описанный выше алгоритм называется **алгоритмом оптимального распределения потерь**, и обозначается записью $Hedge(\beta)$. Впервые он был изложен в [3].

Лемма 1.

Средняя кумулятивная потеря $L_T = \sum_{t=1}^{T} l_t$ данного алгоритма удовлетворяет неравенству

$$\ln |\vec{w}_{T+1}| \leq -(1 - \beta) L_T. \tag{15}$$

Доказательство.

Докажем эквивалентное неравенство:

$$|\vec{w}_{T+1}| \leq e^{-(1-\beta)L_T}. \tag{16}$$

Согласно определению (14), $\forall\, t = 1, \ldots, T$

$$|\vec{w}_{t+1}| = \sum_{i \in I} w_{t+1}^i = \sum_{i \in I} w_t^i \beta^{l_t^i} \leq \sum_{i \in I} w_t^i (1 - (1-\beta) l_t^i) \tag{17}$$

(в (17) используем неравенство

$$\beta^{l_t^i} \leq 1 - (1-\beta) l_t^i, \tag{18}$$

которое следует из выпуклости функции $y = \beta^x$).

Правая часть (17) равна

$$|\vec{w}_t| - (1-\beta) \sum_{i \in I} w_t^i l_t^i = |\vec{w}_t| (1 - (1-\beta) l_t) \tag{19}$$

Из неравенства $1 + x \leq e^x$ $(\forall\, x \in \mathbb{R})$, где $\mathbb{R}$ обозначает множество действительных чисел, следует неравенство

$$1 - (1-\beta) l_t \leq e^{-(1-\beta) l_t} \tag{20}$$

Из (17), (19) и (20) следует, что $\forall\, t = 1, \ldots, T$

$$|\vec{w}_{t+1}| \leq |\vec{w}_t| e^{-(1-\beta) l_t} \tag{21}$$

Перемножая неравенства (21) для $t = 1, \ldots, T$, производя сокращения, и учитывая $|\vec{w}_1| = 1$, получаем искомое неравенство (16). ■

Перепишем (15) в виде

$$L_T \leq -\tfrac{1}{1-\beta} \ln |\vec{w}_{T+1}|. \tag{22}$$

$\forall\, i \in I$ из неравенства $w_{T+1}^i \leq |\vec{w}_{T+1}|$ следует неравенство

$$-\tfrac{1}{1-\beta} \ln |\vec{w}_{T+1}| \leq -\tfrac{1}{1-\beta} \ln w_{T+1}^i \tag{23}$$

Из (14) следует, что

$$w_{T+1}^i = w_1^i \beta^{L_T^i} = \tfrac{1}{N} \beta^{L_T^i}. \tag{24}$$

Из (22), (23) и (24) следует, что

$$L_T \leq -\tfrac{1}{1-\beta} (\ln \tfrac{1}{N} + L_T^i \ln \beta). \tag{25}$$

Поскольку $\forall\, i \in I$ верно (25), то получаем соотношение

$$L_T \leq \tfrac{1}{1-\beta} \ln \tfrac{1}{\beta} \min_{i \in I} L_T^i + \tfrac{\ln N}{1-\beta}. \tag{26}$$

Неравенство (26) означает, что средние кумулятивные потери ПА $Hedge(\beta)$ не превосходят кумулятивных потерь наилучшего эксперта,

умноженных на константу $\frac{1}{1-\beta}\ln\frac{1}{\beta}$, к которым добавлен регрет (т.е. ошибка обучения) $\frac{\ln N}{1-\beta}$.

Теорема 3.
Если в ПА $Hedge(\beta)$ значение параметра β равно $\frac{1}{1+\sqrt{\frac{2}{T/\ln N}}}$, то

$$L_T \le \min_{i\in I} L_T^i + \sqrt{2T\ln N} + \ln N. \tag{27}$$

Доказательство.
Сначала докажем утверждение: если действительные числа L, L', R, R' удовлетворяют неравенствам $L' > L \ge 0$, $R' \ge R > 0$, и $\beta = \frac{1}{1+\sqrt{\frac{2}{L'/R'}}}$, то

$$\frac{1}{1-\beta}\ln\frac{1}{\beta}L + \frac{1}{1-\beta}R \le L + \sqrt{2L'R'} + R. \tag{28}$$

Обозначим $\sigma = \sqrt{2R'/L'}$.
Имеет место неравенство

$$\ln\frac{1}{\beta} \le \frac{1-\beta^2}{2\beta} \tag{29}$$

т.к. производная функции $f(\beta) = \frac{1-\beta^2}{2\beta} - \ln\frac{1}{\beta}$ равна $-\frac{(\beta-1)^2}{2\beta^2}$, и поскольку $f(1) = 0$, то если бы неравенство (29) было бы неверно, т.е. $f(\beta) < 0$, то, по теореме Лагранжа, $\exists\,\beta' \in (0,1) : f'(\beta') > 0$, что невозможно (производная функции f отрицательна во всех точках интервала $(0,1)$).
Из (29) следует, что $\frac{1}{1-\beta}\ln\frac{1}{\beta} \le \frac{1+\beta}{2\beta}$, поэтому

$$\begin{aligned}
&\text{левая часть (28)} \ \le \tfrac{1+\beta}{2\beta}L + \tfrac{1}{1-\beta}R = \\
&= \tfrac{1}{2}(1+\tfrac{1}{\beta})L + \tfrac{1}{1-\beta}R = L + \tfrac{1}{2}L\sigma + \tfrac{1}{1-\frac{1}{1+\sigma}}R \le \\
&\le L + \sqrt{\tfrac{L'R'}{2}} + R + \tfrac{R}{\sigma} \le \text{правая часть (28)}\ .
\end{aligned}$$

(мы используем равенство $\frac{1}{1-\frac{1}{1+\sigma}} = 1 + \frac{1}{\sigma}$).

Рассматривая (28) для $L = \min_{i\in I} L_T^i$, $L' = T$, $R = R' = \ln N$, и учитывая (26), получаем (27). ■

6. Бустинг

В этом параграфе рассматривается задача построения сильных алгоритмов машинного обучения. Для ее описания приведем необходимые определения.

Пусть заданы множества X и Y, элементы которых называются **объектами** и **ответами** соответственно, как правило, $Y = \{0, 1\}$. **Обучающей выборкой (ОВ)** будем называть совокупность S вида

$$S = \{(x^i, y^i, p^i) \mid i \in I\} \tag{30}$$

где $I = \{1, \ldots, N\}$, $\forall\, i \in I \;\; x^i \in X$, $y^i \in Y$, $(p^1, \ldots, p^N) \in I^{\triangle}$. Запись $|S|$ обозначает число компонентов в S (т.е. N).

Каждая тройка (x, y, p) из ОВ S интерпретируется как утверждение о том, что объекту x соответствует ответ y с мерой уверенности p.

Под **алгоритмом машинного обучения (АМО)** понимаем алгоритм, получающий на вход ОВ S, и выдающий функцию $h : X \to Y$, называемую **классификатором**. **Ошибка** классификатора h на ОВ S – это число

$$Err(h, S) = \textstyle\sum_{i \in I} p^i [\![h(x^i) \neq y^i]\!].$$

АМО называется

- **сильным**, если для каждой ОВ S и $\forall \varepsilon, \delta \in (0, 1)$ он выдает с вероятностью $> 1 - \delta$ за время, полиномиально зависящее от $\frac{1}{\varepsilon}, \frac{1}{\delta}, |S|$, классификатор h, такой, что $Err(h, S) \leq \varepsilon$,
- **слабым**, если для каждой ОВ S $\exists\, \varepsilon \in (0, \frac{1}{2})$: $\forall\, \delta \in (0, 1)$ верно то же свойство, что и для сильного АМО.

Ниже решается следующая задача: пусть имеется слабый АМО, требуется на базе него построить сильный АМО. Метод преобразования слабого АМО в сильный АМО называется **бустингом**. Излагаемый ниже бустинг называется **AdaBoost** (**адаптивное усиление**). Впервые он был изложен в [3]. Говоря неформально, в основе данного бустинга лежит выделение таких элементов ОВ S, на которых классификатор h, получаемый при помощи слабого АМО делает наибольшую ошибку, и коррекция h на именно этих элементах. Входными данными для алгоритма AdaBoost являются ОВ (30) и слабый АМО $WeakLearn$.

Алгоритм AdaBoost имеет следующий вид. Выбирается натуральное число T, и выполняется нижеследующая последовательность из T шагов. На каждом шаге $t = 1, \ldots, T$ определяются следующие объекты:

- вектора $\vec{w}_t = (w_t^1, \ldots, w_t^N)$ и $\vec{p}_t = (p_t^1, \ldots, p_t^N)$, где

  $$\vec{p}_t = norm(\vec{w}_t),$$
  $$\vec{w}_1 = (p^1, \ldots, p^N) \;\; (p^i \text{ – компоненты исходной ОВ } S,$$

- классификатор h_t, получаемый применением исходного слабого АМО $WeakLearn$ к ОВ $S(\vec{p}_t)$, где $S(\vec{p}_t)$ получается из S заменой в каждой входящей в неё тройке (x^i, y^i, p^i) компоненты p^i на p_t^i,

- $\varepsilon_t := Err(h_t, S(\vec{p}_t))$ $(< \frac{1}{2})$, $\beta_t := \frac{\varepsilon_t}{1-\varepsilon_t}$,
- $w^i_{t+1} := w^i_t \beta_t^{l^i_t}$, где $l^i_t = [\![h_t(x^i) = y^i]\!]$.

Затем определяется искомый классификатор h:

$$h(x) = [\![\langle \vec{q}, \vec{h}(x) \rangle \geq \tfrac{1}{2}]\!] \tag{31}$$

где $\vec{q} = norm(\ln \frac{1}{\beta_1}, \ldots, \ln \frac{1}{\beta_T}), \vec{h}(x) = (h_1(x), \ldots, h_T(x))$.

Теорема 4.
Ошибка результирующего классификатора (31) удовлетворяет неравенству

$$Err(h, S) \leq 2^T \prod_{t=1}^T \sqrt{\varepsilon_t(1-\varepsilon_t)}. \tag{32}$$

Доказательство.
Согласно определениям, $\forall\, t = 1, \ldots, T$ верны равенства

$$\varepsilon_t = \textstyle\sum_{i \in I} p^i_t (1 - l^i_t) = 1 - \sum_{i \in I} p^i_t l^i_t = 1 - \frac{1}{|\vec{w}_t|} \sum_{i \in I} w^i_t l^i_t$$

Следовательно, $\sum_{i \in I} w^i_t l^i_t = |\vec{w}_t|(1 - \varepsilon_t)$, откуда, учитывая неравенство (18) для $\beta = \beta_t$, получаем:

$$\begin{array}{l} |\vec{w}_{t+1}| = \sum_{i \in I} w^i_t \beta_t^{l^i_t} \leq \sum_{i \in I} w^i_t (1 - (1 - \beta_t) l^i_t) = \\ = |\vec{w}_t| - (1 - \beta_t) \sum_{i \in I} w^i_t l^i_t = \\ = |\vec{w}_t| - (1 - \beta_t) |\vec{w}_t| \sum_{i \in I} p^i_t l^i_t = \\ = |\vec{w}_t|(1 - (1 - \beta_t)(1 - \varepsilon_t)) = |\vec{w}_t| 2\varepsilon_t. \end{array} \tag{33}$$

Таким образом, $\forall\, t = 1, \ldots, T$

$$|\vec{w}_{t+1}| \leq |\vec{w}_t| 2\varepsilon_t. \tag{34}$$

Перемножая неравенства (34) для $t = 1, \ldots, T$, и учитывая $|\vec{w}_1| = 1$, получаем:

$$|\vec{w}_{T+1}| \leq 2^T \prod_{t=1}^T \varepsilon_t. \tag{35}$$

Отметим, что $\forall\, i \in I$ из $h(x_i) \neq y_i$ следует, что

$$\prod_{t=1}^T \beta_t^{l^i_t} \geq (\prod_{t=1}^T \beta_t)^{1/2} \tag{36}$$

Действительно, $l^i_t = 1 - |h_t(x_i) - y_i|$, и

- если $y_i = 0$ и $h(x_i) = 1$, то $\forall\, t = 1, \ldots, T$

$$\begin{array}{l} \beta_t^{l^i_t} = \beta_t^{1 - |h_t(x_i) - y_i|} = \beta_t^{1 - h_t(x_i)} \\ \sum_{t=1}^T \ln \frac{1}{\beta_t} h_t(x_i) \geq \frac{1}{2} \sum_{t=1}^T \ln \frac{1}{\beta_t} \end{array}$$

  откуда следует (36) для данного случая, и

- если $y_i = 1$ и $h(x_i) = 0$, то $\forall\, t = 1, \ldots, T$

$$\beta_t^{l_t^i} = \beta_t^{1-|h_t(x_i)-y_i|} = \beta_t^{h_t(x_i)}$$
$$\sum_{t=1}^{T} \ln \frac{1}{\beta_t} h_t(x_i) < \frac{1}{2} \sum_{t=1}^{T} \ln \frac{1}{\beta_t}$$

откуда следует (36) для данного случая.

Учитывая (36), получаем:

$$|\vec{w}_{T+1}| \geq \sum_{i:h(x_i)\neq y_i} w_{T+1}^i = \sum_{i:h(x_i)\neq y_i} p^i \prod_{t=1}^{T} \beta_t^{l_t^i} \geq$$
$$\geq (\sum_{i:h(x_i)\neq y_i} p^i)(\prod_{t=1}^{T} \beta_t)^{1/2} = Err(h, S)(\prod_{t=1}^{T} \beta_t)^{1/2},$$

откуда, учитывая (35), получаем (32). ■

Следствие 1.

Пусть $\forall\, t = 1, \ldots, T$ ошибка ε_t классификатора h_t из алгоритма AdaBoost удовлетворяет условию $\varepsilon_t \leq \frac{1}{2} - \gamma_t$, где $\gamma_t > 0$. Тогда ошибка результирующего классификатора (31) удовлетворяет условию

$$Err(h, S) \leq e^{-2\sum_{t=1}^{T} \gamma_t^2}. \tag{37}$$

Доказательство.

В данном случае правая часть (32) равна

$$\prod_{t=1}^{T} \sqrt{1 - 4\gamma_t^2} = e^{\sum_{t=1}^{T} \frac{1}{2} \ln(1-4\gamma_t^2)} \ ,$$

откуда, учитывая неравенство $\ln(1 - 4\gamma_t^2) \leq -4\gamma_t^2$, получаем (37). ■

В частности, если $\forall\, t = 1, \ldots, T\ \gamma_t = \gamma$, то (37) будет иметь вид

$$Err(h, S) \leq e^{-2T\gamma^2}$$

откуда получаем оценку на число итераций AdaBoost, достаточных для выполнения условия $Err(h, S) < \varepsilon$:

$$T > \frac{1}{2\gamma^2} \ln \frac{1}{\varepsilon} \ .$$

7. Алгоритм следования за возмущённым лидером

В этом пункте рассматривается другой подход к решению задачи прогнозирования, описанной в пункте 5. ПА, построенный в соответствии с данным подходом, называется **алгоритмом следования за возмущённым лидером (Follow the Perturbed Leader, FPL)**, описания данного алгоритма и его разновидностей впервые было изложено

в работах [4], [5], [6], [7]. Данный алгоритм является вероятностной модификацией обычного ПА A следования за лидером, который имеет следующий вид: на каждом шаге прогнозирования t определяется лидер, т.е. такой эксперт i, кумулятивные потери L^i_{t-1} которого минимальны, и на шаге t прогноз A полагается равным прогнозу лидера i. Потери ПА A в момент t совпадают с потерями эксперта i в момент t. Такой ПА может привести к потерям, существенно превышающим потери каждого из экспертов. Например, пусть число экспертов равно двум, и последовательности их потерь на шагах $1, \ldots, 7$ имеют вид

$$\begin{aligned} l^1_{1,\ldots,7} &= (0.5,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1), \\ l^2_{1,\ldots,7} &= \quad (0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0). \end{aligned}$$

Последовательности соответствующих кумулятивных потерь имеют вид

$$\begin{aligned} L^1_{1,\ldots,7} &= (0.5,\ 0.5,\ 1.5,\ 1.5,\ 2.5,\ 2.5,\ 3.5), \\ L^2_{1,\ldots,7} &= \quad (0,\quad 1,\quad 1,\quad 2,\quad 2,\quad 3,\quad 3). \end{aligned}$$

Нетрудно видеть, что в данном случае лидерами на шагах $2, \ldots, 7$ являются соответственно 2, 1, 2, 1, 2, 1, и каждый раз, следуя за лидером на текущем шаге, ПА следования за лидером будет нести потерю 1, и его кумулятивные потери на шагах $2, \ldots, 7$ будет иметь вид

$$L_{2,\ldots,7} = (1,\ 2,\ 3,\ 4,\ 5,\ 6)$$

т.е. его кумулятивные потери на каждом шаге примерно вдвое больше кумулятивных потерь каждого из экспертов.

Излагаемый ниже ПА FPL отличается от детерминированного ПА следования за лидером лишь в изменении понятия лидера: на каждом шаге t лидером среди экспертов $1, \ldots, N$ является тот эксперт i (называемый **возмущённым лидером**), у которого минимальной является величина

$$L^i_{t-1} - \tfrac{1}{\varepsilon_t}\xi^i,$$

где ε_t – параметр, и $\xi^1, \ldots, \xi^N$ – независимые одинаково распределенные **случайные величины (СВ)** , с экспоненциальным законом распределения, т.е. их плотность имеет вид $p(x) = e^{-x}$ $(x \geq 0)$.

$\forall\, t = 1, \ldots, T$ обозначим

$$\begin{aligned} i_t &= \text{ СВ } \arg\min_{i \in I}(L^i_{t-1} - \tfrac{1}{\varepsilon_t}\xi^i) \\ l_t &= \mathbf{E} l^{i_t}_t = \textstyle\sum_{i \in I} l^i_t \mathbf{P}\{i_t = i\},\ \ L_T = \sum_{t=1}^T l_t, \end{aligned}$$

где $\mathbf{P}\{\varphi\}$ обозначает вероятность события φ, а $\mathbf{E}\xi$ обозначает математическое ожидание СВ ξ.

Теорема 5.
Если параметр ε_t из ПА FPL имеет вид $\sqrt{\frac{2\ln N}{t}}$, то

$$L_T \le \min_{i\in I} L_T^i + 3\sqrt{2T\ln N}. \tag{38}$$

Доказательство.
$\forall\, t = 1,\dots,T$ обозначим

$$\begin{aligned} i'_t &= \text{ СВ } \arg\min_{i\in I}(L_t^i - \tfrac{1}{\varepsilon_t}\xi^i) \\ l'_t &= \mathbf{E} l_t^{i'_t} = \textstyle\sum_{i\in I} l_t^i \mathbf{P}\{i'_t = i\},\ \ L'_T = \sum_{t=1}^T l'_t. \end{aligned}$$

Неравенство (38) следует из доказываемых ниже соотношений (39) и (49).

1) Докажем неравенства

$$L_T \le L'_T + \textstyle\sum_{t=1}^T \varepsilon_t \le L'_T + 2\sqrt{2T\ln N}. \tag{39}$$

Второе неравенство следует из определения ε_t и свойства

$$\textstyle\sum_{t=1}^T \frac{1}{\sqrt{t}} \le 1 + \int_1^T \frac{dt}{\sqrt{t}} < 2\sqrt{T}, \tag{40}$$

а первое неравенство следует из свойства

$$\forall\, t = 1,\dots,T \quad l_t - l'_t \le \varepsilon_t l_t \ \ (\le \varepsilon_t,\ \text{т.к. } l_t \in [0,1]). \tag{41}$$

(41) следует из неравенств

$$l'_t \ge e^{-\varepsilon_t} l_t \ge (1-\varepsilon_t) l_t. \tag{42}$$

Второе неравенство в (42) следует из свойства

$$\forall\, x \in \mathbb{R}\ \ e^{-x} \ge 1 - x,$$

а первое неравенство в (42) можно переписать в виде

$$\textstyle\sum_{i\in I} l_t^i \mathbf{P}\{i_t = i\} \le e^{\varepsilon_t} \sum_{i\in I} l_t^i \mathbf{P}\{i'_t = i\} \tag{43}$$

(43) следует из свойства $\forall\, i \in I$

$$\mathbf{P}\{i_t = i\} \le e^{\varepsilon_t} \mathbf{P}\{i'_t = i\}. \tag{44}$$

(44) следует из соответствующих неравенств для условных вероятностей: $\forall\, c_1,\dots,c_N \ge 0$

$$\begin{aligned} &\mathbf{P}\{i_t = i \mid \forall\, j \ne i\ \ \xi^j = c_j\} \le \\ &\le e^{\varepsilon_t} \mathbf{P}\{i'_t = i \mid \forall\, j \ne i\ \ \xi^j = c_j\} \end{aligned} \tag{45}$$

Докажем (45). Обозначим условие $\forall\, j \neq i \;\; \xi^j = c_j$ символом φ, и определим

$$m_i = \min_{j\neq i}(L^j_{t-1} - \tfrac{1}{\varepsilon_t}c_j),$$
$$m'_i = \min_{j\neq i}(L^j_t - \tfrac{1}{\varepsilon_t}c_j) = \min_{j\neq i}(L^j_{t-1} + l^j_t - \tfrac{1}{\varepsilon_t}c_j).$$

Нетрудно видеть, что $m_i \leq m'_i$.

Используя введенные выше обозначения, неравенство (45) можно переписать в виде неравенства условных вероятностей:

$$\begin{gathered}\mathbf{P}\{L^i_{t-1} - \tfrac{1}{\varepsilon_t}\xi^i \leq m_i \mid \varphi\} \leq \\ \leq e^{\varepsilon_t}\mathbf{P}\{L^i_{t-1} + l^i_t - \tfrac{1}{\varepsilon_t}\xi^i \leq m'_i \mid \varphi\}.\end{gathered} \tag{46}$$

Если в неравенстве в правой части (46) заменить l^i_t на 1, а m'_i на m_i, то данное неравенство усилится, поэтому (46) следует из неравенства

$$\begin{gathered}\mathbf{P}\{L^i_{t-1} - \tfrac{1}{\varepsilon_t}\xi^i \leq m_i \mid \varphi\} \leq \\ \leq e^{\varepsilon_t}\mathbf{P}\{L^i_{t-1} + 1 - \tfrac{1}{\varepsilon_t}\xi^i \leq m_i \mid \varphi\},\end{gathered} \tag{47}$$

которое эквивалентно неравенству

$$\begin{gathered}\mathbf{P}\{\xi^i \geq \varepsilon_t(L^i_{t-1} - m_i) \mid \varphi\} \leq \\ \leq e^{\varepsilon_t}\mathbf{P}\{\xi^i \geq \varepsilon_t(L^i_{t-1} - m_i + 1) \mid \varphi\}.\end{gathered} \tag{48}$$

(48) обосновывается следующими свойствами экспоненциально распределенной СВ ξ: $\forall\, a, b \geq 0$

$$\mathbf{P}\{\xi \geq a\} = e^{-a},$$
$$\mathbf{P}\{\xi \geq a + b\} = e^{-b}\mathbf{P}\{\xi \geq a\}.$$

2) Докажем, что

$$L'_T \leq \min_{i\in I} L^i_T + \tfrac{\ln N}{\varepsilon_T}. \tag{49}$$

Будем использовать следующие обозначения:

$$\begin{gathered}\vec{l_t} := (l^1_t, \ldots, l^N_t),\;\; \vec{L}_t := (L^1_t, \ldots, L^N_t), \\ \vec{\xi} := (\xi^1, \ldots, \xi^N), \\ \tilde{l}_t = \vec{l_t} - \vec{\xi}(\tfrac{1}{\varepsilon_t} - \tfrac{1}{\varepsilon_{t-1}}),\;\; \tilde{L}_t = \vec{L}_t - \vec{\xi}\tfrac{1}{\varepsilon_t}\end{gathered} \tag{50}$$

(полагаем $\varepsilon_0 = 1$).

Нетрудно доказать, что $\tilde{L}_T = \tilde{L}_{T-1} + \tilde{l}_T$.

Пусть $E = \{\vec{e}_1, \ldots, \vec{e}_N\} \subseteq \mathbb{R}^N$, где $\forall\, i = 1, \ldots, N$ $\vec{e}_i$ имеет вид $(0, \ldots, 1, \ldots 0)$ (единица – на i-м месте).

$\forall \vec{l} = (l^1, \ldots, l^N) \in \mathbb{R}$ обозначим

$$M(\vec{l}) = \arg\min_{\vec{e}_i \in E} \langle \vec{e}_i, \vec{l} \rangle,$$

Нетрудно видеть, что

$$\begin{array}{l} \langle M(\vec{l}), \vec{l} \rangle = \min_{i \in I} l^i, \\ l_t^{i'_t} = \langle M(\tilde{L}_t), \vec{l}_t \rangle, \quad L'_T = \mathbf{E} \sum_{t=1}^T \langle M(\tilde{L}_t), \vec{l}_t \rangle, \\ \langle M(\tilde{L}_{T-1}), \tilde{L}_{T-1} \rangle \leq \langle M(\tilde{L}_T), \tilde{L}_{T-1} \rangle \end{array} \tag{51}$$

(неравенство в третьей строчке (51) следует из того, что его левая часть – минимальная компонента $\tilde{L}_{T-1}$, а правая – некоторая компонента $\tilde{L}_{T-1}$).

Индукцией по T докажем неравенство

$$\sum_{t=1}^T \langle M(\tilde{L}_t), \tilde{l}_t \rangle \leq \langle M(\tilde{L}_T), \tilde{L}_T \rangle. \tag{52}$$

При $T = 1$ (52) имеет вид $\langle M(\tilde{l}_1), \tilde{l}_1 \rangle \leq \langle M(\tilde{l}_1), \tilde{l}_1 \rangle$.

Индуктивный переход (от $T-1$ к T): используя индуктивное предположение, и учитывая неравенство в третьей строчке (51), получаем:

$$\begin{array}{l} \text{левая часть (52)} \leq \\ \leq \langle M(\tilde{L}_{T-1}), \tilde{L}_{T-1} \rangle + \langle M(\tilde{L}_T), \tilde{l}_T \rangle \leq \\ \leq \langle M(\tilde{L}_T), \tilde{L}_{T-1} \rangle + \langle M(\tilde{L}_T), \tilde{l}_T \rangle = \langle M(\tilde{L}_T), \tilde{L}_T \rangle = \\ = \text{ правая часть (52)}. \end{array}$$

Используя определение $\tilde{l}_t$ (см. третью строчку в (50)), неравенство (52) можно переписать так:

$$\begin{array}{l} \sum_{t=1}^T \langle M(\tilde{L}_t), \vec{l}_t \rangle \leq \\ \leq \langle M(\tilde{L}_T), \tilde{L}_T \rangle + \sum_{t=1}^T \langle M(\tilde{L}_t), \vec{\xi} \rangle (\frac{1}{\varepsilon_t} - \frac{1}{\varepsilon_{t-1}}). \end{array} \tag{53}$$

Из определения $\tilde{L}_T$, следует неравенство

$$\begin{array}{l} \langle M(\tilde{L}_T), \tilde{L}_T \rangle \leq \langle M(\vec{L}_T), \vec{L}_T - \vec{\xi} \frac{1}{\varepsilon_T} \rangle = \\ = \min_{i \in I} L_T^i - \langle M(\vec{L}_T), \xi \rangle \frac{1}{\varepsilon_T}. \end{array} \tag{54}$$

Т.к. $\langle M(\vec{L}_T), \xi \rangle = \xi^k$ для некоторого k, и $\mathbf{E}\xi^k = 1$, то

$$\mathbf{E} \langle M(\vec{L}_T), \xi \rangle \frac{1}{\varepsilon_T} = \frac{1}{\varepsilon_T} \mathbf{E} \xi^k = \frac{1}{\varepsilon_T}. \tag{55}$$

Оценим второй член в (53):

$$\sum_{t=1}^{T}\langle M(\tilde{L}_t),\vec{\xi}\rangle(\tfrac{1}{\varepsilon_t}-\tfrac{1}{\varepsilon_{t-1}}\rangle)\le$$
$$\le\sum_{t=1}^{T}\max_{i\in I}\xi^i(\tfrac{1}{\varepsilon_t}-\tfrac{1}{\varepsilon_{t-1}})\le\tfrac{1}{\varepsilon_T}\max_{i\in I}\xi^i.$$

Нетрудно доказать, что

$$\mathbf{E}\max_{i=1,\ldots,N}\xi^i\le 1+\ln N. \tag{56}$$

Действительно, поскольку СВ $\xi^1,\ldots,\xi^N$ независимы, и функция распределения показательно распределённой СВ имеет вид $1-e^{-x}$, то функция распределения СВ $\max_{i=1,\ldots,N}\xi^i$ имеет вид $(1-e^{-x})^N$, поэтому её плотность равна $N(1-e^{-x})^{N-1}$, и следовательно её мат. ожидание равно

$$N\int_0^\infty(1-e^{-x})^{N-1}e^{-x}xdx. \tag{57}$$

Обозначим (57) записью a_N. Поскольку

$$a_N=N\int_0^\infty(1-e^{-x})^{N-1}e^{-x}xdx=$$
$$=N\int_0^\infty(1-e^{-x})(1-e^{-x})^{N-2}e^{-x}xdx=$$
$$=\tfrac{N}{N-1}a_{N-1}-N\int_0^\infty e^{-x}(1-e^{-x})^{N-2}e^{-x}xdx=$$
$$=\tfrac{N}{N-1}a_{N-1}-\tfrac{N}{N-1}\int_0^\infty e^{-x}xd(1-e^{-x})^{N-1}=$$
(применяем интегрирование по частям)
$$=\tfrac{N}{N-1}a_{N-1}+\tfrac{N}{N-1}\int_0^\infty(1-e^{-x})^{N-1}de^{-x}x=$$
$$=\tfrac{N}{N-1}a_{N-1}+\tfrac{N}{N-1}\int_0^\infty(1-e^{-x})^{N-1}e^{-x}(1-x)dx=$$
$$=\tfrac{N}{N-1}a_{N-1}+\tfrac{1}{N-1}(1-a_N),$$

откуда получаем: $a_N=a_{N-1}+\frac{1}{N}$, следовательно

$$\mathbf{E}\max_{i=1,\ldots,N}\xi^i=a_N=1+\tfrac{1}{2}+\ldots+\tfrac{1}{N}, \tag{58}$$

откуда следует (56), поэтому

$$\mathbf{E}\sum_{t=1}^{T}\langle M(\tilde{L}_t),\vec{\xi}\rangle(\tfrac{1}{\varepsilon_t}-\tfrac{1}{\varepsilon_{t-1}})\le$$
$$\le\frac{\mathbf{E}\max_{i=1,\ldots,N}\xi^i}{\varepsilon_T}\le\frac{1+\ln N}{\varepsilon_T} \tag{59}$$

Таким образом, согласно второй строчке в (51), а также (53), (54), (55) и (59)

$$L'_T=\mathbf{E}\sum_{t=1}^{T}\langle M(\tilde{L}_t),\vec{l}_t\rangle\le\min_{i\in I}L^i_T-\tfrac{1}{\varepsilon_T}+\frac{1+\ln N}{\varepsilon_T}$$

откуда следует (49). ■

8. Агрегирующий алгоритм В.Г.Вовка

В этом пункте описывается агрегирующий алгоритм В.Г.Вовка, существенной особенностью которого является зависимость регрета $R_T = L_T - \min_{i \in I} L_T^i$ только от количества экспертов N и независимость регрета от величины периода наблюдения T. Данный алгоритм был впервые описан в [8].

8.1. Смешиваемые функции потерь

Напомним некоторые введённые ранее понятия и обозначения:

- $Y = \{0, 1\}$ – множество **исходов**,
- $\Gamma = [0, 1]$, или $[-1, 1]$, или $Y^\triangle$ – множество **прогнозов**,
- $\eta > 0$ – **параметр обучения**, $\beta = e^{-\eta}$,
- $\lambda : Y \times \Gamma \to \mathbb{R}_{\geq 0}$ – **функция потерь (ФП)**, она предполагается непрерывной по второму аргументу,
- $I = 1, \ldots, N$ – множество **экспертов**,
- $\forall\, t \geq 1$, $\forall\, i \in I$
  - γ_t^i – **прогноз** эксперта i на шаге t,
  - y_t – **исход** на шаге t,
  - $l_t^i = \lambda(\gamma_t^i, y_t)$ – **потери** эксперта i на шаге t,
    $L_t^i = \sum_{t'=1}^t l_{t'}^i$ – кумулятивные потери эксперта i на шаге t,
  - $m_t = \log_\beta \sum_{i \in I} \beta^{l_t^i} p_{t-1}^i$ – **средние потери** на шаге t,
    $M_t = \sum_{t'=1}^t m_{t'}$ – кумулятивные средние потери.

Алгоритм обучения: это построение последовательностей $\vec{w}_0, \vec{w}_1, \ldots$ и $\vec{p}_0, \vec{p}_1, \ldots$ векторов из $\mathbb{R}^I$, где

- $\vec{w}_0 = \vec{p}_0 \in I^\triangle$,
- $\forall\, t \geq 1, \forall\, i \in I \;\; w_t^i = \beta^{l_t^i} w_{t-1}^i = \beta^{L_t^i} p_0^i$,
- $\forall\, t \geq 1 \;\; \vec{p}_t = norm(\vec{w}_t) \in I^\triangle$.

Отметим, что $M_t = \log_\beta \sum_{i \in I} \beta^{L_t^i} p_0^i$. Действительно,

$$
\begin{aligned}
&m_t = \log_\beta \textstyle\sum_{i \in I} \beta^{l_t^i} p_{t-1}^i = \log_\beta \sum_{i \in I} \beta^{l_t^i} \frac{w_{t-1}^i}{\sum_{j \in I} w_{t-1}^j} = \\
&= \log_\beta \frac{\sum_{i \in I} \beta^{l_t^i} w_{t-1}^i}{\sum_{j \in I} w_{t-1}^j} = \log_\beta \frac{\sum_{i \in I} \beta^{L_t^i} p_0^i}{\sum_{j \in I} \beta^{L_{t-1}^j} p_0^i} = \\
&= \log_\beta \textstyle\sum_{i \in I} \beta^{L_t^i} p_0^i - \log_\beta \sum_{i \in I} \beta^{L_{t-1}^i} p_0^i
\end{aligned}
$$

откуда непосредственно следует доказываемое равенство.

Примеры ФП:

$$\lambda(\gamma, y) = \begin{cases} c(y-\gamma)^2 \text{ (квадратичная, } c \text{ – константа)}, \\ c|y-\gamma| \text{ (абсолютная, } c \text{ – константа)}, \\ [\![y \neq \gamma]\!] \text{ (простая)}, \\ -\ln \gamma(y) \text{ (логарифмическая, } \Gamma = Y^{\triangle}). \end{cases}$$

В последнем случае можно отождествить распределение

$$\gamma = (\gamma(1), \gamma(0)) \in Y^{\triangle}$$

с числом $\gamma(1) \in [0,1]$, которое будем обозначать тем же символом γ, и значение $\lambda(\gamma, y)$ в данном случае можно записать в виде $-\ln|1-y-\gamma|$.

ФП λ называется **смешиваемой ФП (СФП)**, если

$$\mathcal{U}_\lambda = \bigcup_{\gamma\in\Gamma}[0, \beta^{\lambda(\gamma,0)}] \times [0, \beta^{\lambda(\gamma,1)}] \tag{60}$$

является выпуклым подмножеством $\mathbb{R}^2$ (это будет, например, когда ФП λ – квадратичная или логарифмическая.

В настоящей статье рассматривается задача вычисления прогнозов в том случае, когда ФП λ является СФП.

Теорема 6.

Если λ – СФП, то $\forall\, t \geq 1 \;\; \exists\, \gamma_t^* \in \Gamma : \forall\, y \in Y$

$$\lambda(\gamma_t^*, y) \leq m_t. \tag{61}$$

Доказательство.

Совокупность точек

$$\{(\beta^{(\lambda(\gamma_t^i,0)}, \beta^{\lambda(\gamma_t^i,1)}) \mid i \in I\} \tag{62}$$

принадлежит выпуклому множеству $\mathcal{U}_\lambda$.

Согласно определению, m_t – выпуклая комбинация вида

$$m_t = \log_\beta \textstyle\sum_{i\in I} \beta^{\lambda(\gamma_t^i, y)} p_{t-1}^i.$$

Т.к. $\mathcal{U}_\lambda$ выпукло, то выпуклая комбинация

$$\textstyle\sum_{i\in I}(\beta^{(\lambda(\gamma_t^i,0)}, \beta^{\lambda(\gamma_t^i,1)} p_{t-1}^i \tag{63}$$

точек из множества (62) тоже принадлежит $\mathcal{U}_\lambda$.

Построим луч с началом в $0 = (0,0)$, проходящий через точку (63). Этот луч пересекает границу

$$\{(\beta^{\lambda(\gamma,0)}, \beta^{\lambda(\gamma,1)}) \mid \gamma \in \Gamma\}$$

множества $\mathcal{U}_\lambda$ в некоторой точке

$$u = (\beta^{\lambda(\gamma_t^*,0)}, \beta^{\lambda(\gamma_t^*,1)}). \tag{64}$$

Поскольку точка (63) принадлежит отрезку $[0, u]$, то её абсцисса и ордината не превосходят абсциссы и ординаты соответственно точки u, т.е.

$$\sum_{i\in I} \beta^{\lambda(\gamma_t^i,0)} p_{t-1}^i \le \beta^{\lambda(\gamma_t^*,0)},$$
$$\sum_{i\in I} \beta^{\lambda(\gamma_t^i,1)} p_{t-1}^i \le \beta^{\lambda(\gamma_t^*,1)},$$

т.е. верно утверждение

$$\forall\, y \in Y \quad \sum_{i\in I} \beta^{\lambda(\gamma_t^i,y)} p_{t-1}^i \le \beta^{\lambda(\gamma_t^*,y)}. \tag{65}$$

Логарифмируя неравенство в (65), получаем (61). ■

L_T^{AA} обозначает кумулятивную потерю $\sum_{t=1}^T \lambda(\gamma_t^*, y_t)$, где γ_t^* – прогнозы, определяемые в доказательстве теоремы 6 (AA является аббревиатурой словосочетания «агрегирующий алгоритм»). Из теоремы 6 следует неравенство

$$L_T^{AA} \le M_T.$$

Отметим, что если $\vec{p}_0$ – р.р., то $\forall\, i \in I$

$$L_t^{AA} \le M_t = \log_\beta(\textstyle\sum_{i\in I} \beta^{L_t^i} \frac{1}{N}) \le \log_\beta(\beta^{L_t^i} \frac{1}{N}),$$

поэтому

$$L_T^{AA} \le \min_{i\in I} L_T^i + \tfrac{\ln N}{\eta}. \tag{66}$$

8.2. Смешиваемость квадратичной функции потерь

Будем считать, что $\Gamma = [-1, 1]$, $Y = \{-1, 1\}$ (м.б. и $[-1, 1]$). В этом случае $\mathcal{U}_\lambda$ имеет вид

$$\textstyle\bigcup_{\gamma\in\Gamma} [0, \beta^{\lambda(\gamma,-1)}] \times [0, \beta^{\lambda(\gamma,1)}]$$

Лемма 2.

ФП $\lambda(\gamma, y) = (y - \gamma)^2$ является η–смешиваемой тогда и только тогда, когда $\eta \le \frac{1}{2}$.

Доказательство.

Нетрудно доказать, что множество $\mathcal{U}_\lambda$ выпукло тогда и только тогда, когда его граница – кривая

$$\hat{\mathcal{U}}_\lambda \stackrel{\text{def}}{=} \{(\beta^{\lambda(\gamma,-1)}, \beta^{\lambda(\gamma,1)}) \mid \gamma \in [-1, 1]\} =$$
$$= \{(e^{-\eta(-1-\gamma)^2}, e^{-\eta(1-\gamma)^2}) \mid \gamma \in [-1, 1]\} \tag{67}$$

обладает следующим свойством: при увеличении γ от -1 до 1 абсцисса соответствующей точки кривой уменьшается и кривая поворачивает налево, что эквивалентно свойству

$$\forall\,\gamma\in[-1,1]\quad\begin{cases}\gamma_x<0,\\ y_{xx}=\gamma_x\frac{x_\gamma y_{\gamma\gamma}-x_{\gamma\gamma}y_\gamma}{x_\gamma^2}\le 0.\end{cases}\tag{68}$$

Из первого неравенства в (68) следует, что второе неравенство в (68) равносильно неравенству

$$x_\gamma y_{\gamma\gamma}\ge x_{\gamma\gamma}y_\gamma.\tag{69}$$

Нетрудно видеть, что

$$\begin{aligned}x_\gamma&=-2\eta(1+\gamma)e^{-\eta(1+\gamma)^2}\\ x_{\gamma\gamma}&=2\eta(-1+2\eta(1+\gamma)^2)e^{-\eta(1+\gamma)^2}\\ y_\gamma&=2\eta(1-\gamma)e^{-\eta(1-\gamma)^2}\\ y_{\gamma\gamma}&=2\eta(-1+2\eta(1-\gamma)^2)e^{-\eta(1-\gamma)^2}\end{aligned}$$

откуда следует, что (69) можно переписать в виде

$$-(1+\gamma)(-1+2\eta(1-\gamma)^2)\ge(1-\gamma)(-1+2\eta(1+\gamma)^2).$$

Последнее неравенство эквивалентно утверждению

$$\eta(1-\gamma^2)\le\tfrac12,$$

которое должно быть верным для каждого $\gamma\in[-1,1]$, что равносильно неравенству $\eta\le\frac12$. ■

Рассмотрим задачу вычисления оптимального прогноза γ_t^* для квадратичной ФП в случае $\eta=\frac12$.

Точка (63) в данном случае имеет вид (A,B), где

$$A=\textstyle\sum_i\beta^{\lambda(\gamma_t^i,-1)}p_{t-1}^i,\quad B=\sum_i\beta^{\lambda(\gamma_t^i,1)}p_{t-1}^i,$$

и точка (64) ищется из уравнения

$$\frac{B}{A}=\frac{\beta^{\lambda(\gamma_t^*,1)}}{\beta^{\lambda(\gamma_t^*,1)}}=\beta^{(1-\gamma_t^*)^2-(-1-\gamma_t^*)^2}=\beta^{-4\gamma_t^*}.$$

Поэтому $\gamma_t^*=\frac14\log_\beta\frac{A}{B}$.

8.3. Супермартингалы

8.3.1. Понятие супермартингала

Пусть $Y = \{0, 1\}$, $\Gamma = [0, 1]$. Ниже запись u_n обозначает произвольную последовательность из $(\Gamma \times Y)^n$ $(n \geq 0)$:

$$u_n = ((\gamma_1, y_1), \ldots, (\gamma_n, y_n)).$$

Последовательность u_0 пуста и обозначается ε.

Супермартингал (СМ) – это семейство функций

$$Q = \{Q_n : (\Gamma \times Y)^n \to \mathbb{R}_{\geq 0} \mid n \geq 0\} \tag{70}$$

таких, что

- $Q_0(\varepsilon) \leq 1$,
- $\forall\, n \geq 0, \forall\, u_n \in (\Gamma \times Y)^n, \forall\, y \in Y$ функция

  $$Q_{n+1}(u_n, (\cdot, y)) : \Gamma \to \mathbb{R}_{\geq 0}$$

  непрерывна, и верно свойство

$$\begin{aligned} \forall\, \gamma \in \Gamma \quad & \gamma Q_{n+1}(u_n, (\gamma, 1)) + \\ & + (1 - \gamma) Q_{n+1}(u_n, (\gamma, 0)) \leq Q_n(u_n). \end{aligned} \tag{71}$$

Теорема 7.
Пусть задан СМ Q вида (70).
Тогда $\forall\, n \geq 0$, $\forall\, u_n \in (\Gamma \times Y)^n$ $\exists \gamma^*$:

$$\forall\, y \in Y \quad Q_n(u_n) \geq Q_{n+1}(u_n, (\gamma^*, y)). \tag{72}$$

Доказательство.
Определим функцию $f_{u_n} : \Gamma \times Y \to \mathbb{R}_{\geq 0}$:

$$f_{u_n}(\gamma, y) = Q_{n+1}(u_n, (\gamma, y)) - Q_n(u_n).$$

f_{u_n} непрерывна по γ, и из (71) следует, что

$$\forall\, \gamma \in \Gamma \quad \gamma f_{u_n}(\gamma, 1) + (1 - \gamma) f_{u_n}(\gamma, 0) \leq 0 \tag{73}$$

поэтому $f_{u_n}(1, 1) \leq 0$ и $f_{u_n}(0, 0) \leq 0$.

Докажем, что

$$\exists\, \gamma^* : f_{u_n}(\gamma^*, 0) \leq 0 \text{ и } f_{u_n}(\gamma^*, 1) \leq 0. \tag{74}$$

Отметим, что из (74) следует (72).

- Если $f_{u_n}(1,0) \le 0$, то $\gamma^* = 1$, и если $f_{u_n}(0,1) \le 0$, то $\gamma^* = 0$,
- иначе $f_{u_n}(1,0) > 0$ и $f_{u_n}(0,1) > 0$, в этом случае рассмотрим непрерывную функцию

$$f(\gamma) = f_{u_n}(\gamma, 1) - f_{u_n}(\gamma, 0).$$

Поскольку $f(0) > 0$, $f(1) < 0$, и f непрерывна, то

$$\exists\, \gamma^* \in (0,1) : f(\gamma^*) = 0,$$

т.е. $f_{u_n}(\gamma^*, 1) = f_{u_n}(\gamma^*, 0)$. По (73), отсюда следует (74). ■

8.3.2. Пример супермартингала

Определим $\forall\, i \in I = \{1, \dots, N\}$

$$R_n^i(u_n) = \textstyle\sum_{t=1}^n (\lambda(\gamma_t, y_t) - \lambda(\gamma_t^i, y_t)).$$

В некоторых случаях $Q_n^i(u_n) = e^{\eta R_n^i} = e^{\eta(L_n - L_n^i)}$ будет СМ. Неравенство в (71) равносильно неравенству

$$\gamma e^{\eta(\lambda(\gamma,1) - \lambda(\gamma_{n+1}^i, 1))} + (1-\gamma) e^{\eta(\lambda(\gamma,0) - \lambda(\gamma_{n+1}^i, 0))} \le 1 \tag{75}$$

1) Если $\lambda(\gamma, y) = -\ln|1 - y - \gamma|$, то (75) имеет вид

$$\gamma e^{\eta(-\ln\gamma + \ln\gamma_{n+1}^i)} + (1-\gamma) e^{\eta(-\ln(1-\gamma) + \ln(1-\gamma_{n+1}^i))} \le 1 \tag{76}$$

что равносильно неравенству

$$\gamma^{1-\eta}(\gamma_{n+1}^i)^\eta + (1-\gamma)^{1-\eta}(1-\gamma_{n+1}^i)^\eta \le 1. \tag{77}$$

Если $\eta = \frac{1}{2}$, то (77) следует из неравенства Коши- Буняковского для векторов

$$(\gamma^{1-\eta}, (1-\gamma)^{1-\eta}), \quad ((\gamma_{n+1}^i)^\eta, (1-\gamma_{n+1}^i)^\eta).$$

Левая часть (77) – скалярное произведение этих векторов, а их норма в случае $\eta = \frac{1}{2}$ равна 1.

2) $\lambda(\gamma, y) = (y-\gamma)^2 : y \in \{0,1\}, \gamma \in [0,1], \eta \in (0,2]$.

В этом случае (75) равносильно неравенству

$$\gamma e^{\eta((\gamma-1)^2 - (1-\gamma_{n+1}^i)^2)} + (1-\gamma) e^{\eta(\gamma^2 - (\gamma_{n+1}^i)^2)} \le 1. \tag{78}$$

Представим γ^i_{n+1} в виде $\gamma + x$ и перепишем (78) в виде

$$\gamma e^{2\eta(1-\gamma)x} + (1-\gamma)e^{-2\eta\gamma x} \le e^{\eta x^2} \tag{79}$$

(79) вытекает из следующего утверждения: если значения СВ ξ лежат в отрезке $[a,b]$, то $\forall\, s \in \mathbb{R}$

$$\ln \mathbf{E} e^{s\xi} \le s\mathbf{E}\xi + \tfrac{s^2(b-a)^2}{8}. \tag{80}$$

Если СВ ξ принимает значение 1 с вероятностью γ и значение 0 с вероятностью $1-\gamma$, то полагая $a=0$, $b=1$ получаем: (80) имеет вид

$$\gamma e^{s(1-\gamma)} + (1-\gamma)e^{-s\gamma} \le e^{s^2/8}.$$

Если $s := 2\eta x$, то Л.Ч. (79) $\le e^{\eta^2 x^2/2} \le$ Пр.Ч. (79), ибо $\eta \le 2$.

8.3.3. Применение теоремы 7

Пусть $Y = \{0,1\}$, $\Gamma = [0,1]$, ФП $\lambda(\gamma, y)$ – η-смешиваемая, $w_0 \in I^{\triangle}$, где $I = \{1,\dots,N\}$, построены прогнозы $\gamma_1,\dots,\gamma_{T-1}$ и имеются прогнозы экспертов $\gamma^i_1,\dots,\gamma^i_T$ $(i \in I)$.

(66) будет выполнено, если $\forall\, t$ верно (65), что равносильно следующему: $\forall\, y \in \{0,1\}$ верно неравенство

$$\textstyle\sum_{i\in I} w^i_{t-1} \ge \sum_{i\in I} w^i_{t-1} e^{-\eta(\lambda(\gamma^i_t,y)-\lambda(\gamma_t,y))}$$

которое эквивалентно неравенству

$$\textstyle\sum_{i\in I} p^i_0 e^{-\eta L^i_{t-1}} \ge \sum_{i\in I} p^i_0 e^{-\eta L^i_{t-1}} e^{\eta(\lambda(\gamma_t,y)-\lambda(\gamma^i_t,y))},$$

после домножения обоих частей которого на $e^{\eta L_{t-1}}$ получаем

$$\textstyle\sum_{i\in I} p^i_0 Q^i_{t-1} \ge \sum_{i\in I} p^i_0 Q^i_{t-1} e^{\eta(\lambda(\gamma_t,y)-\lambda(\gamma^i_t,y))} = \sum_{i\in I} p^i_0 Q^i_t(y).$$

Таким образом утверждение (65) равносильно тому, что последовательность $\{\sum_{i\in I} p^i_0 Q^i_t \mid t \ge 1\}$ не возрастает с ростом t.

Нетрудно доказать, что семейство функций

$$\{\textstyle\sum_{i\in I} p^i_0 Q^i_n \mid n \ge 1\}$$

тоже является СМ, и поэтому для него верна теорема 7, т.е. $\forall\, t > 0$, $\forall\, u \in (\Gamma \times Y)^{t-1}$ $\exists\, \gamma_t : \forall\, y \in Y$

$$\textstyle\sum_{i\in I} p^i_0 Q^i_t(u,(\gamma_t,y)) \le \sum_{i\in I} p^i_0 Q^i_{t-1}(u) \le 1. \tag{81}$$

$\forall\, t \ge 1$ из свойства

$$\textstyle\sum_{i\in I} p^i_0 Q^i_t = \sum_{i\in I} p^i_0 e^{\eta(L_t - L^i_t)} \le 1$$

следует, что $\forall\, i \in I$ $\ p^i_0 e^{\eta(L_t - L^i_t)} \le 1$, откуда следует (66), если $p^i_0 = 1/N$.

9. Прогнозные стратегии

В этом пункте рассматривается задача прогнозирования временного ряда в ситуации, когда эксперты отсутствуют. Для определения качества алгоритма прогнозирования в данном случае используется понятие калибруемости алгоритма, впервые введённое в [10]. Излагаемый в данном пункте вероятностный алгоритм вычисления калибруемых прогнозов впервые был описан в [11], см. также [9] и [6].

9.1. Понятие прогнозной стратегии

Пусть множество исходов Y имеет вид $\{0,1\}$. Будем обозначать произвольную последовательность из $Y^n = \underbrace{Y \times \ldots \times Y}_{n}$ записью y^n, и последний элемент последовательности y^n – записью y_n. Обозначим записью Y^* множество $\bigcup_{n\geq 0} Y^n$, где Y^0 состоит из пустой последовательности y^0 (которая обозначается ε). Символом y будем обозначать неограниченную последовательность элементов множества Y, и если y – такая последовательность,то $\forall\, n \geq 1$ записи y^n и y_n обозначают префикс последовательности y длины n и n-й элемент y соответственно.

Прогнозная стратегия (ПС) – это функция

$$f : Y^* \to [0,1], \text{ где } f(\varepsilon) = 1 \text{ и } \forall\, n \geq 0, \forall\, y^n \in Y^n$$
$$f(y^n) = \mathbf{P}\{y^n = \text{ последовательность первых } n \text{ исходов}\}$$

Для каждой ПС f и $\forall\, n \geq 0$ верно равенство

$$\textstyle\sum_{y^n \in Y^n} f(y^n) = 1.$$

Будем обозначать $f(y_n \mid y^{n-1}) = \frac{f(y^n)}{f(y^{n-1})}$.

Пусть $\mathcal{F}$ – некоторый класс ПС. $\forall\, y \in Y^*$ будем обозначать записью $\mathcal{F}(y)$ число $\sum_{\varphi\in\mathcal{F}} \varphi(y)$.

Ниже будем опускать «$\in \mathcal{F}$» в $\sum_{\varphi\in\mathcal{F}}, \sup_{\varphi\in\mathcal{F}}, \inf_{\varphi\in\mathcal{F}}$.

Потери ПС f определяются следующим образом: $\forall\, n \geq 1$

$$l_n^f(y) = -\ln f(y_n \mid y^{n-1}),$$
$$L_n^f(y) = \textstyle\sum_{i=1}^n l_i^f(y) = -\sum_{i=1}^n \ln f(y_i \mid y^{i-1}) = -\ln f(y^n).$$

Регрет ПС f относительно класса ПС $\mathcal{F}$:

$$R_n^f(y) = L_n^f(y) - \inf_\varphi L_n^\varphi(y) = \sup_\varphi \ln \tfrac{\varphi(y^n)}{f(y^n)} = \ln \tfrac{\sup_\varphi \varphi(y^n)}{f(y^n)}$$
$$R_n(y) = \inf_\varphi R_n^\varphi(y).$$

9.2. Примеры прогнозных стратегий

9.2.1. Смешивающая и минимаксная прогнозные стратегии

Пусть задан конечный класс $\mathcal{F}$ ПС. **Смешивающая ПС** для класса $\mathcal{F}$ определяется следующим образом:

$$f(y_n) = \tfrac{1}{|\mathcal{F}|}\mathcal{F}(y^n).$$

Нетрудно видеть, что

$$R_n^f(y) = \sup_\varphi \ln \tfrac{\varphi(y^n)}{f(y^n)} = \sup_\varphi \ln \tfrac{\varphi(y^n)}{\frac{1}{|\mathcal{F}|}\mathcal{F}(y^n)} =$$
$$= \ln|\mathcal{F}| + \sup_\varphi \ln \tfrac{\varphi(y^n)}{\mathcal{F}(y^n)} \le \ln|\mathcal{F}|.$$

Минимаксная ПС для класса $\mathcal{F}$ определяется следующим образом:

$$f(y^n) = \frac{\sup_\varphi \varphi(y^n)}{\sum_{u^n \in Y^n} \sup_\varphi \varphi(u^n)}.$$

Ниже будем опускать «$\in Y^n$» в записях $\sum_{u^n \in Y^n}$.

Минимаксный регрет определяется следующим образом:

$$V_n^f = \sup_{y^n} R_n^f(y) = \sup_{y^n} \ln \tfrac{\sup_\varphi \varphi(y^n)}{f(y^n)}$$
$$V_n = \inf_{\varphi\in\mathcal{F}} V_n^\varphi.$$

Нетрудно видеть, что

$$R_n^f(y) = \ln \tfrac{sup_\varphi \varphi(y^n)}{f(y^n)} = \ln \frac{\sup_\varphi \varphi(y^n)}{\sup_\varphi \varphi(y^n)} = \tag{82}$$
$$= \ln \textstyle\sum_{u^n} \sup_\varphi \varphi(u^n).$$

Отметим, что правая часть (82) не зависит от y и f, поэтому можно обозначить её R_n или V_n.

ПС f называется **оптимальной**, если $\forall\, n \ge 0\ V_n^f = V_n$.

Докажем оптимальность минимаксной ПС f, т.е. следующее свойство: для каждой ПС $f' \ne f\ V_n^{f'} \ge V_n$:

- если $\forall\, y^n \in Y^n\ f'(y^n) = f(y^n)$, то

$$V_n^{f'} = \sup_{y^n} \ln \tfrac{\sup_\varphi \varphi(y^n)}{f'(y^n)} = \sup_{y^n} \ln \tfrac{\sup_\varphi \varphi(y^n)}{f(y^n)} = V_n^f,$$

- иначе, если $\exists\, y^n \in Y^n : f'(y^n) \ne f(y^n)$, то, поскольку

$$\textstyle\sum_{y^n} f'(y^n) = \sum_{y^n} f(y^n) = 1,$$

то $\exists\, y^n : f'(y^n) < f(y^n)$, поэтому

$$R_n^{f'}(y) = \ln \tfrac{\sup_\varphi \varphi(y^n)}{f'(y^n)} > \ln \tfrac{\sup_\varphi \varphi(y^n)}{f(y^n)} = V_n,$$

откуда следует: $V_n^{f'} = \sup_{y^n} R_n^{f'}(y) > V_n$. ■

Отметим, что

$$V_n = \ln \sum_{u^n} \sup_\varphi \varphi(u^n) \le \ln \sum_{u^n} \sum_\varphi \varphi(u^n) =$$
$$= \ln \sum_\varphi \sum_{u^n} \varphi(u^n) = \ln \sum_\varphi 1 \le \ln |\mathcal{F}|.$$

9.2.2. Прогнозная стратегия Лапласа

ПС Лапласа, применяется в ситуации, когда $\forall\, n \ge 1$ y_n генерируется независимо, и $\forall\, n \ge 1$ $\mathbf{P}\{y_n = 1\}$ равно одному и тому же числу p.

Ниже $\forall\, n \ge 0$ и $\forall\, y^n \in Y^n$ n_1 и n_2 обозначают число единиц и нулей соответственно в последовательности y^n.

Нетрудно доказать, что вероятность того, что y^n является последовательностью первых n исходов, равна $p^{n_1}(1-p)^{n_2}$.

Определим класс ПС $\mathcal{F}$ как класс функций, каждая из которых соответствует некоторому числу $p \in [0,1]$ и сопоставляет последовательности $y^n \in Y^n$ вероятность $p^{n_1}(1-p)^{n_2}$ того, что y^n – последовательность первых n исходов.

ПС Лапласа f имеет следующий вид:

$$f(y^n) \stackrel{\text{def}}{=} \int_0^1 p^{n_1}(1-p)^{n_2} dp, \tag{83}$$

т.е. $f(y^n)$ равно мат. ожиданию вероятности того, что y^n является последовательностью первых n исходов. Нетрудно доказать, что данное значение равно $\frac{1}{(n+1)C_n^{n_1}}$. Это доказывается обратной индукцией по n_1:

- для $n_1 = n$ имеем $\int_0^1 p^n dp = \frac{1}{n+1}$, что верно, и
- если верно

  $$\int_0^1 p^{n_1+1}(1-p)^{n_2-1} dp = \frac{1}{(n+1)C_n^{n_1+1}} =: A$$

  то, интегрируя по частям, получаем:

  $$\int_0^1 p^{n_1}(1-p)^{n_2} dp = \frac{n-n_1}{n_1+1} A = \frac{1}{(n+1)C_n^{n_1}}.$$

Нетрудно видеть, что

$$f(y_{n+1} = 1 \mid y^n) = \frac{f(y^n 1)}{f(y^n)} = \frac{1}{(n+2)C_{n+1}^{n_1+1}} \Big/ \frac{1}{(n+1)C_n^{n_1}} = \frac{n_1+1}{n+2},$$
$$f(y_{n+1} = 0 \mid y^n) = \frac{n_2+1}{n+2}.$$

$$\forall\, y \;\; R_n^f(y) = \ln \frac{\sup_{0 \le p \le 1} p^{n_1}(1-p)^{n_2}}{\int_0^1 p^{n_1}(1-p)^{n_2} dp} = \ln \frac{(\frac{n_1}{n})^{n_1}(\frac{n_2}{n})^{n_2}}{\frac{1}{(n+1)C_n^{n_1}}} \le \ln(n+1).$$

Для оптимальной ПС оценка V_n имеет следующий вид:

$$V_n = \tfrac{1}{2}\ln n + \tfrac{1}{2}\ln\tfrac{\pi}{2} + \varepsilon_n \;\; (\text{где } \varepsilon_n \to 0). \tag{84}$$

Действительно,

$$\begin{array}{l} V_n = \ln \sum_{u^n} \sup_{\varphi} \varphi(u^n) = \\ = \ln \sum_{u^n} \sup_{0\le p\le 1} p^{n_1}(1-p)^{n_2} = \\ = \ln \sum_{u^n} (\frac{n_1}{n})^{n_1} (\frac{n_2}{n})^{n_2} = \\ = \ln \sum_{n_1=0}^{n} C_n^{n_1} (\frac{n_1}{n})^{n_1} (\frac{n_2}{n})^{n_2}. \end{array} \tag{85}$$

Из формулы Стирлинга

$$\sqrt{2\pi n}(\tfrac{n}{e})^n e^{\frac{1}{12n}} \le n! \le \sqrt{2\pi n}(\tfrac{n}{e})^n e^{\frac{1}{12n+\varepsilon}}$$

(где $\varepsilon \in (0,1)$ – некоторая константа) следует неравенство

$$\tfrac{1}{\sqrt{2\pi}}\sqrt{\tfrac{n}{n_1 n_2}} e^{\frac{1}{12n}} \le C_n^{n_1}(\tfrac{n_1}{n})^{n_1}(\tfrac{n_2}{n})^{n_2} \le \tfrac{1}{\sqrt{2\pi}}\sqrt{\tfrac{n}{n_1 n_2}} e^{\frac{1}{12n+1}}$$

из которого, учитывая (85), получаем верхнюю оценку

$$V_n \le \ln\left((1+o(1))\sqrt{\tfrac{n}{2\pi}} e^{\frac{1}{12n+1}} \sum_{n_1=1}^{n-1} \tfrac{1}{\sqrt{n_1 n_2}}\right)$$

Однако

$$\sum_{n_1=1}^{n-1} \tfrac{1}{\sqrt{n_1 n_2}} = \sum_{n_1=1}^{n-1} \tfrac{1}{n} \tfrac{1}{\sqrt{\frac{n_1}{n}(1-\frac{n_1}{n})}} \approx \int_0^1 \tfrac{dx}{\sqrt{x(1-x)}} = \pi,$$

поэтому $V_n \le \ln((1+o(1))\sqrt{\frac{n\pi}{2}} =$ правая часть (84).

Нижняя оценка для n устанавливается аналогично. ■

Из (84) следует, что для оптимальной ПС f, т.е. такой ПС f, что $V_n = V_n^f$, будет выполнено свойство

$$\forall\, y\ L_n^f(y) - \inf_{\varphi} L_n^{\varphi}(y) \le \text{ Пр.Ч. (84)},$$

или: $\forall\, y,\ \forall\, \varphi \in \mathcal{F}\ \ L_n^f(y) \le L_n^{\varphi}(y) + \text{ Пр.Ч. (84)}$.

9.3. Детерминированное прогнозирование

В этом и следующем пунктах мы рассмотрим задачу прогнозирования временного ряда в условиях, когда эксперты отсутствуют. Мы рассмотрим детерминированный и вероятностный алгоритмы прогнозирования.

Ниже используется следующее обозначение: если a – некоторый объект, и m – сообщение, то запись $a!m$ обозначает действие, которое заключается в том, что объект a посылает в окружающую среду сообщение m.

Алгоритм детерминированного прогнозирования (АДП) заключается в выполнении следующих действий:

$$\forall\, n \ge 1 \begin{cases} \text{прогнозист } !\ \gamma_n \in [0,1] \\ \text{природа } !\ y_n \in [0,1] \end{cases}$$

Будем говорить, что АДП **калибруется**, если для каждой последовательности исходов $(y_1, y_2, \ldots)$ и каждого связного подмножества $I \subseteq [0,1]$ последовательность прогнозов $(\gamma_1, \gamma_2, \ldots)$ обладает следующим свойством:

$$\frac{\sum_{i=1}^{n} [\![\gamma_i \in I]\!](y_i - \gamma_i)}{\sum_{i=1}^{n} [\![\gamma_i \in I]\!]} \to 0 \text{ при } \sum_{i=1}^{n} [\![\gamma_i \in I]\!] \to \infty. \tag{86}$$

АДП не калибруется, для обоснования этого определим

$$I_1 = [0, \tfrac{1}{2}], \quad I_2 = [\tfrac{1}{2}, 1], \quad \forall\, n \geq 1 \quad y_n = [\![\gamma_n < \tfrac{1}{2}]\!]$$

откуда следует, что $\forall\, n \geq 1\ |y_n - \gamma_n| \geq \frac{1}{2}$.

Нетрудно установить, что (86) нарушается когда $I = I_1$ или $I = I_2$. Действительно, в один из отрезков I_1, I_2 попадает бесконечное число точек γ_i, и свойство $\sum_{i=1}^{n} [\![\gamma_i \in I]\!] \to \infty$ для этого отрезка верно, однако модуль дроби в (86) больше или равен $\frac{1}{2}$.

9.4. Вероятностное прогнозирование

Алгоритм вероятностного прогнозирования (АВП) имеет следующий вид:

$$\forall\, n \geq 1 \begin{cases} \text{прогнозист } !\ p_n \in [0,1]^{\triangle} \\ \text{природа } !\ y_n \in \{0,1\} \\ \text{ГСЧ } !\ \gamma_n \sim p_n \end{cases} \tag{87}$$

где ГСЧ – генератор случайных чисел, третье действие в (87) заключается в случайном порождении значения $\gamma_n \in [0,1]$ в соответствии с распределением p_n.

Свойство **калибруемости** АВП имеет следующий вид:

- для каждого $\delta > 0$, и
- для каждой последовательности исходов $(y_1, y_2, \ldots)$

последовательность прогнозов $(\gamma_1, \gamma_2, \ldots)$, которую порождает АВП, удовлетворяет условию: $\forall\, I \subseteq [0,1]$

$$\models \left(\lim_{n\to\infty} \sup \left| \tfrac{1}{n} \sum_{i=1}^{n} [\![\gamma_i \in I]\!](y_i - \gamma_i) \right| \right) \leq \delta \tag{88}$$

где обозначение $\models A$ имеет следующий смысл: событие A выполняется с вероятностью 1.

Теорема 8.
Существует калибруемый АВП.

Доказательство.
Представим доказательство в виде последовательности этапов.

1) $\forall\, k \geq 1$ обозначим $V_k = \{v_0, \ldots, v_k\}$, где $v_i = \frac{i}{k}$.

$\forall\, c \in [0,1]$ определён отрезок $[v_{i-1}, v_i]$, который содержит c.

Число c можно представить в виде суммы

$$c = \lambda v_{i-1} + (1-\lambda)v_i, \quad \text{где } \lambda \in [0,1].$$

$$\forall\, v \in V_k \text{ определим } w_v(c) \stackrel{\text{def}}{=} \begin{cases} \lambda & \text{при } v = v_{i-1}, \\ 1-\lambda & \text{при } v = v_i, \\ 0, & \text{иначе.} \end{cases}$$

Таким образом, $c = \sum_{v \in V_k} w_v(c) v$.

2) Определим индуктивно последовательность $c_1, c_2, \ldots$ чисел из $[0,1]$ следующим образом. Полагаем $c_1 = 0$.

Пусть для некоторого n определены числа

$$c_1, \ldots, c_{n-1}. \tag{89}$$

$\forall\, v \in V_k$ будем использовать обозначение

$$\mu_n(v) = \textstyle\sum_{i=1}^{n} w_v(c_i)(y_i - c_i), \tag{90}$$

где $c_1, \ldots, c_{n-1}$ – числа из (89), и y_n, c_n – переменные.

Из (90) следует, что

$$\begin{aligned} &\mu_n(v)^2 = \mu_{n-1}(v) + w_v(c_n)(y_n - c_n))^2 = \\ &= \mu_{n-1}(v)^2 + 2\mu_{n-1}(v) w_v(c_n)(y_n - c_n) + w_v(c_n)^2 (y_n - c_n)^2, \end{aligned}$$

поэтому $\sum_{v \in V_k} \mu_n(v)^2 = A + 2(y_n - c_n)B + C$, где

$$\begin{cases} A = \sum_{v \in V_k} \mu_{n-1}(v)^2 \\ B = \sum_{v \in V_k} w_v(c_n)\mu_{n-1}(v) \\ C = (y_n - c_n)^2 \sum_{v \in V_k} w_v(c_n)^2 \end{cases}$$

Заметим:

$$\begin{aligned} &B = \textstyle\sum_{v \in V_k} w_v(c_n) \sum_{i=1}^{n-1} w_v(c_i)(y_i - c_i) = \\ &= \textstyle\sum_{i=1}^{n-1} \Big(\sum_{v \in V_k} w_v(c_n) w_v(c_i) \Big)(y_i - c_i) = \\ &= \textstyle\sum_{i=1}^{n-1} \langle \vec{w}(c_n), \vec{w}(c_i) \rangle (y_i - c_i) = \\ &= \textstyle\sum_{i=1}^{n-1} K(c_n, c_i)(y_i - c_i), \end{aligned}$$

где $\vec{w}(c) = (w_{v_0}(c), \ldots w_{v_k}(c))$, $K(c_n, c_i) = \langle \vec{w}(c_n), \vec{w}(c_i) \rangle$.

Определяем c_n следующим образом:

- если уравнение

$$B(c) = \sum_{i=1}^{n-1} K(c, c_i)(y_i - c_i) = 0$$

имеет корень в $[0, 1]$, то полагаем c_n равным этому корню,

- иначе $c_n = 1$ или 0, если $\forall\, c \in [0, 1]\ B(c) > 0$ или $B(c) < 0$ соответственно.

3) Отметим, что $2(y_n - c_n)B \le 0$ и $C \le \sum_{v \in V_k} w_v(c_n) = 1$.

Таким образом,

$$\begin{array}{l} \sum_{v \in V_k} \mu_n(v)^2 = A + 2(y_n - c_n)B + C \le \\ \le \sum_{v \in V_k} \mu_{n-1}(v)^2 + 1, \end{array}$$

откуда, учитывая равенство $\mu_0(v) = 0$, получаем:

$$\sum_{v \in V_k} \mu_n(v)^2 \le n. \tag{91}$$

4) Определим $p_n \in V_k^{\triangle} : \forall\, v \in V_k\ \ p_n(v) = w_v(c_n)$.

$\forall\, i = 1, \ldots, n$ рассмотрим СВ

$$\xi_i = [\![p_i \in I]\!](y_i - p_i),$$

где $I \subseteq [0, 1]$ – заданное подмножество. Используя ξ_i, перепишем условие (88) в виде

$$\models \left(\lim_{n \to \infty} \sup \left| \tfrac{1}{n} \sum_{i=1}^{n} \xi_i \right| \right) \le \delta. \tag{92}$$

Нетрудно видеть, что

$$\mathbf{E}\xi_i = \sum_{v \in V_k} w_v(c_i)[\![v \in I]\!](y_i - v). \tag{93}$$

По усиленному закону больших чисел,

$$\models \lim_{n \to \infty} \left(\tfrac{1}{n} \sum_{i=1}^{n} \xi_i - \tfrac{1}{n} \sum_{i=1}^{n} \mathbf{E}\xi_i \right) = 0. \tag{94}$$

Поскольку $\forall\, i = 1, \ldots, n$

$$\begin{array}{l} |\mathbf{E}\xi_i - \sum_{v \in V_k} w_v(c_i)[\![v \in I]\!](y_i - c_i)| = \\ = |\sum_{v \in V_k} w_v(c_i)[\![v \in I]\!](y_i - v) - \\ - \sum_{v \in V_k} w_v(c_i)[\![v \in I]\!](y_i - c_i)| = \\ = |\sum_{v \in V_k} w_v(c_i)[\![v \in I]\!](c_i - v)| < \delta \end{array} \tag{95}$$

то

$$|\sum_{i=1}^{n} \mathbf{E}\xi_i| \le |\sum_{i=1}^{n} \sum_{v \in V_k} w_v(c_i)[\![v \in I]\!](y_i - c_i)| + \delta n. \tag{96}$$

Обозначим записями $\vec{\mu}_n$ и $\vec{I}$ вектора

$$(\mu_n(v_0), \ldots, \mu_n(v_k)) \text{ и } ([\![v_0 \in I]\!], \ldots, [\![v_k \in I]\!])$$

соответственно. Из (91) следует, что $||\vec{\mu}_n|| \leq \sqrt{n}$.

Используя неравенство Коши-Буняковского, оценим первое слагаемое в правой части (96):

$$\begin{array}{l} |\sum_{i=1}^{n} \sum_{v \in V_k} w_v(c_i)[\![v \in I]\!](y_i - c_i)| = \\ = |\sum_{v \in V_k} (\sum_{i=1}^{n} w_v(c_i)(y_i - c_i)[\![v \in I]\!]| = \\ = |\sum_{v \in V_k} \mu_n(v)[\![v \in I]\!]| = \\ = |\langle \vec{\mu}_n, \vec{I} \rangle| \leq ||\vec{\mu}_n|| \cdot ||\vec{I}|| \leq \sqrt{n}\sqrt{k+1} \end{array} \tag{97}$$

Из (4.15) и (4.16) следует:

$$|\sum_{i=1}^{n} \mathbf{E}\xi_i| \leq \sqrt{n}\sqrt{k+1} + \delta n. \tag{98}$$

Из (98) и (94) получаем соотношение (92). ■

Нетрудно доказать более сильное утверждение: существует АВП, такой, что для каждой последовательности исходов $(y_1, y_2, \ldots)$ последовательность прогнозов $(\gamma_1, \gamma_2, \ldots)$ обладает свойством:

$$\forall\, I \subseteq [0,1] \quad \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \xi_i = 0.$$

Для обоснования этого утверждения в процессе конструирования чисел c_n нужно в определенные моменты времени n_s $(s \geq 1)$ изменять δ, т.е. вместо фиксированного δ рассматривать последовательность δ_s $(s \geq 1)$, стремящуюся к 0.

10. Заключение

В работе были изложены основные понятия смешивающего прогнозирования, и приведены доказательства основных свойств изложенных алгоритмов смешивающего прогнозирования. Развитие изложенных результатов может заключаться, например, путем нового определения меры качества алгоритма прогнозирования, и построения алгоритма смешивающего прогнозирования, оптимального относительно новой меры качества. Например, в качестве такой меры качества можно выбрать долю ошибочных предсказаний алгоритма смешивающего прогнозирования не на всём периоде наблюдения, а на некоторой его части, на которой предсказания экспертов имеют высокую точность.

Список литературы

[1] Вьюгин В.В., *Математические основы машинного обучения и прогнозирования*, МЦНМО, Москва, 2018, 384 pp.

[2] Littlestone N., Warmuth M., "The weighted majority algorithm", *Information and Computation*, **108** (1994), 212–261

[3] Freund Y., Schapire R.E., "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", *Journal of Computer and System Sciences*, **55** (1997), 119–139

[4] Hannan J., "Approximation to Bayes risk in repeated plays", *Contributions to the Theory of Games (ed. by M.Dresher, A.W.Tucker, and P. Wolfe)*, **3** (1957), 97–139

[5] Kalai A., Vempala S., "Efficient algorithms for online decisions", *Journal of Computer and System Sciences*, **71** (2005), 291–307

[6] G. Lugosi, N. Cesa-Bianchi, *Prediction, Learning and Games*, Cambridge University Press, New York, 2006

[7] M. Hutter, J. Poland, "Adaptive online prediction by following the perturbed leader", *Journal of Machine Learning Research*, **6** (2005), 639–660

[8] V. Vovk, "Aggregating strategies", *Proceedings of the 3rd Annual Workshop on Computational Learning Theory (M. Fulk and J.Case, editors)*, 1990, 371–383

[9] Cover Thomas M., Thomas Joy A., *Elements of Information Theory, 2nd ed.*, John Wiley and Sons, Inc., 2006, 748 pp.

[10] A.P. Dawid, "Calibration-based empirical probability", *Ann. Statist.*, **13** (1985), 1251–1285

[11] A. Chernov, Y. Kalnishkan, F. Zhdanov, V. Vovk, "Supermartingales in Prediction with Expert Advice", *Theoretical Computer Science*, **411**:29-30 (2010), 2647–2669

Mathematical foundations of time series prediction

Mironov A.M.

The article outlines the basic concepts and methods of prediction time series. Various mixing prediction algorithms are considered and assessments of the quality of these algorithms are provided.

Keywords: time series, prediction algorithms, mixing prediction

References

[1] Vyugin V.V., *Mathematical foundations of machine learning and prediction*, MCNMO, Moskva, 2018 (In Russian), 384 pp.

[2] Littlestone N., Warmuth M., "The weighted majority algorithm", *Information and Computation*, **108** (1994), 212–261

[3] Freund Y., Schapire R.E., "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", *Journal of Computer and System Sciences*, **55** (1997), 119–139

[4] Hannan J., "Approximation to Bayes risk in repeated plays", *Contributions to the Theory of Games (ed. by M.Dresher, A.W.Tucker, and P. Wolfe)*, **3** (1957), 97–139

[5] Kalai A., Vempala S., "Efficient algorithms for online decisions", *Journal of Computer and System Sciences*, **71** (2005), 291–307

[6] G. Lugosi, N. Cesa-Bianchi, *Prediction, Learning and Games*, Cambridge University Press, New York, 2006

[7] M. Hutter, J. Poland, "Adaptive online prediction by following the perturbed leader", *Journal of Machine Learning Research*, **6** (2005), 639–660

[8] V. Vovk, "Aggregating strategies", *Proceedings of the 3rd Annual Workshop on Computational Learning Theory (M. Fulk and J.Case, editors)*, 1990, 371–383

[9] Cover Thomas M., Thomas Joy A., *Elements of Information Theory, 2nd ed.*, John Wiley and Sons, Inc., 2006, 748 pp.

[10] A.P. Dawid, "Calibration-based empirical probability", *Ann. Statist.*, **13** (1985), 1251–1285

[11] A. Chernov, Y. Kalnishkan, F. Zhdanov, V. Vovk, "Supermartingales in Prediction with Expert Advice", *Theoretical Computer Science*, **411**:29-30 (2010), 2647–2669

Об индексе ассоциативности конечных квазигрупп

К. Д. Царегородцев[1]

В статье рассматриваются результаты, связанные с оценками числа ассоциативных троек в произвольных квазигруппах и в квазигруппах из некоторых классов. Приведены результаты исследований, описывающие количество ассоциативных троек в квазигруппах, задаваемых правильными семействами булевых функций малых размеров.

Ключевые слова: ассоциативная тройка, квазигруппа, правильное семейство булевых функций.

1. Введение

Квазигруппы — одни из базовых структур в алгебре. Таблицы умножения квазигрупп, более известные под названием «латинские квадраты», с древнейших времен и по настоящее время используются в различных областях математики [1]: при планировании статистических экспериментов, в играх и головоломках, в теории кодирования и криптографии. Из общих обзоров криптографических приложений квазигрупп можно отметить следующие источники:

- статья [2], в которой приводятся примеры кодов аутентификации, шифров и однонаправленных функций на основе квазигрупповых преобразований, а также недавний обзор [3], затрагивающий тематику построения симметричных криптопримитивов на основе квазигрупповых операций;
- монография [4], в которой довольно подробно освещена тематика использования квазигрупп в криптографии; в частности, в работе рассматриваются следующие темы: поточные шифры и их криптоанализ, хэш-функции и односторонние функции, схемы разделения секрета; а также смежная тематика теории кодирования (в частности, рекурсивные МДР-коды);
- монография [1] и статья [5], посвященные общим обзорам тематики латинских квадратов, их использованию в докомпьютерный этап развития криптографии и современным приложениям.

[1] *Царегородцев Кирилл Денисович* — старший специалист-исследователь Лаборатории Криптографии, АО «НПК «Криптонит» e-mail: kirill94_12@mail.ru.
Tsaregorodtsev Kirill Denisovich — senior researcher, JSRPC "Kryptonite".

Для того, чтобы некоторые криптографические примитивы, основанные на квазигрупповом умножении, были стойкими к криптоанализу, необходимо, чтобы в квазигруппе было как можно меньше ассоциативных троек, то есть, чтобы квазигрупповая операция была как можно менее ассоциативна. Так, например, большое количество ассоциативных троек может быть использовано при нахождении коллизий и вторых прообразов для некоторых хэш-функций, построенных на основе квазигруппового умножения [6]. Следовательно, с практической точки зрения интересны следующие вопросы:

- каково минимально возможное (и достижимое) число ассоциативных троек для квазигрупп заданного размера?
- можно ли построить классы квазигрупп с заданным малым числом ассоциативных троек?
- можно ли найти квазигруппы с малым числом ассоциативных троек и компактным описанием (в частности, для которых не нужно было бы хранить всю таблицу умножения в компьютере, а вычислять результат квазигрупповой операции более эффективно)?

Указанные вопросы, а также тесно связанные с ними (например, каково *минимально возможное* число неассоциативных троек в неассоциативной квазигруппе заданного порядка?) изучались с 1980-х годов и в отрыве от практических приложений (см. работы [7, 8, 9, 10, 11], а также задачу 1.1 в [1]). Таким образом, сформулированные вопросы интересны как с точки зрения практики, так и чисто теоретически. В данной работе мы рассматриваем большинство полученных на данный момент результатов по количеству ассоциативных троек в квазигруппах, а также приводим результаты исследований, описывающих количество ассоциативных троек в квазигруппах, задаваемых правильными семействами булевых функций малых размеров.

2. Предварительные сведения

Приведем стандартные определения из теории квазигрупп (более подробно см., например, [1, 12]).

Определение 1. Квазигруппой $(Q, \circ)$ называется множество Q с заданной на нем бинарной операцией $\circ\colon Q \times Q \to Q$, удовлетворяющей следующему условию: для любых $a, b \in Q$ найдутся единственные элементы $x, y \in Q$ — решения уравнений $a \circ x = b$, $y \circ a = b$.

Далее мы будем рассматривать конечные квазигруппы $|Q| < \infty$, для краткости слово «конечный» будем опускать. Также иногда будем писать

«квазигруппа Q» без явного упоминания операции $\circ$, если она понятна из контекста.

Замечание 1. *Пусть $(Q, \circ)$ — квазигруппа, тогда для каждого $a \in Q$ можно задать операции левого L_a и правого R_a сдвига:*

$$L_a \colon Q \to Q,\ L_a(x) = a \circ x,$$
$$R_a \colon Q \to Q,\ R_a(y) = y \circ a.$$

Операции L_a и R_a задают биективные отображения на множестве Q.

Определение 2. Латинский квадрат размера k — это квадратная таблица размера $k \times k$, заполненная некоторыми k различными элементами таким образом, что в каждой строке и в каждом столбце каждый элемент встречается ровно один раз.

Определение 3. Пусть $(Q, \circ)$ — квазигруппа, $Q = \{q_1, \ldots, q_k\}$. Таблицей умножения Q будем называть квадратную таблицу размера $k \times k$, заполненную элементами $q \in Q$ таким образом, что на пересечении i-й строки и j-го столбца записывается произведение $(q_i \circ q_j) \in Q$.

Замечание 2. *Латинские квадраты являются таблицами умножения квазигруппы. Это следует из того факта, что левые и правые сдвиги являются биекциями.*

Далее мы будем отождествлять квазигруппу с латинским квадратом, задающим ее таблицу умножения.

Определение 4. Пусть $(Q, \circ)$ — квазигруппа. Ее изотопом называется квазигруппа $(Q_{\alpha\beta\gamma}, *)$ с операцией $*$, заданной на том же множестве Q по правилу $a * b = \gamma^{-1}(\alpha(a) \circ \beta(b))$, где $\alpha, \beta, \gamma \in \mathcal{S}_Q$ — биекции на Q.

Определение 5. Главным изотопом $Q_{\alpha\beta}$ называется изотоп квазигруппы Q с дополнительным условием $\gamma = \mathsf{id}$, где id — тождественное отображение на Q.

Определение 6. Биекция $\sigma \in \mathcal{S}_Q$ называется ортоморфизмом квазигруппы $(Q, \circ)$, если отображение θ, задаваемое правилом $x \circ \theta(x) = \sigma(x)$ также является биекцией на множестве Q.

Определение 7. Идемпотентом в квазигруппе $(Q, \circ)$ называется элемент $x \in Q$ со свойством $x \circ x = x$.

Определение 8. Ассоциативной тройкой называется тройка элементов квазигруппы $a, b, c \in Q$ таких, что выполнено равенство:

$$(a \circ b) \circ c = a \circ (b \circ c).$$

Определение 9 ([10]). Индексом ассоциативности $a(Q)$ квазигруппы Q называется число ассоциативных троек в ней.

Индекс ассоциативности, как было отмечено выше, является важной характеристикой квазигруппы, которая, в частности, показывает, насколько квазигрупповая операция близка к групповой. В дальнейшем изложении нам понадобятся следующие обозначения:

- $a(Q)$: индекс ассоциативности для квазигруппы Q;
- $b(Q)$: число неассоциативных троек в квазигруппе Q;
- $a(n)$: минимальное число ассоциативных троек среди всех квазигрупп порядка n;
- $a(n, C)$: минимальное число ассоциативных троек среди всех квазигрупп из класса C порядка n;
- $b(n)$: минимальное число неассоциативных троек среди всех неассоциативных квазигрупп порядка n;
- $b(n, C)$: минимальное число неассоциативных троек среди всех квазигрупп из класса C порядка n.

3. Оценки на число ассоциативных троек

Очевидно, что число ассоциативных троек в квазигруппе не может превышать $|Q|^3$ — общего числа всех троек элементов в квазигруппе. Данная оценка достижима при условии что Q — группа. Можно легко получить следующую универсальную для всех квазигрупп оценку.

Утверждение 1 ([13]). *Выполняется следующее двойное неравенство:*

$$n \leq a(n) \leq n^3.$$

Утверждение следует из того факта, что в квазигруппе Q для каждого элемента $x \in Q$ существуют левая и правая единицы $le(x), re(x) \in Q$ со свойством $le(x) \circ x = x = x \circ re(x)$. Тогда для каждого $x \in Q$ тройка $(le(x), x, re(x))$ является ассоциативной:

$$(le(x) \circ x) \circ re(x) = x = le(x) \circ (x \circ re(x)).$$

Одной из первых работ, в которых изучалось число ассоциативных троек в алгебраических структурах, является работа [8], автор которой

исследовал коммутативные группоиды. В [8] показано, что для коммутативного неассоциативного группоида Q порядка n верны оценки:

$$n^2 \leq a(Q) \leq n^3 - 2,$$

причем каждая из границ достижима в классе коммутативных группоидов (при $n \geq 3$). Также в [8] рассмотрены классы коммутативных квазигрупп, изотопных группам, коммутативных медиальных квазигрупп и несколько других классов, для каждого из которых получены похожие оценки ($\Theta(n^2)$ для нижней границы и $\Theta(n^3)$ для верхней).

Работа [7] также посвящена группоидам (а именно, классу группоидов с сокращением, частными случаями которых являются квазигруппы). Следствием результатов из работы [7] является неравенство $b(n) \geq n$ (т.е. число неассоциативных троек в группоидах Q с сокращениями не может быть меньше, чем $|Q|$).

Работы [9, 11] посвящены смежному вопросу: каково минимальное число неассоциативных троек в неассоциативной квазигруппе? В работе [9] был рассмотрен класс квазигрупп, изотопных группам, и на него были расширены некоторые результаты из работы [8]. Общим результатом этих работ является следующее наблюдение.

Утверждение 2 ([9, теорема 5.1]). *Пусть C — класс всех неассоциативных квазигрупп Q, изотопных группам. Тогда:*

$$b(n, C) \geq \begin{cases} 4n^2 - 6n, \ n \geq 3, \ n \text{ нечетно;} \\ 4n^2 - 8n, \ n \text{ четно.} \end{cases}$$

В [11] для исследования величины $b(Q)$ вводится следующая характеристика квазигрупповой операции.

Определение 10. Для квазигруппы $(Q, \circ)$ определим расстояние до множества групп $\mathsf{gdist}(Q)$ как минимум среди чисел $\mathsf{dist}(Q, G)$, где $G = (Q, \cdot)$ — группа, заданная на том же множестве, что и квазигруппа $(Q, \circ)$, а функция dist определена следующим образом:

$$\mathsf{dist}(Q, G) = |\{(x, y) \in Q^2 \mid x \circ y \neq x \cdot y\}|.$$

Утверждение 3 ([11, утверждение 4.1]). *Пусть Q — квазигруппа порядка n, $t = \mathsf{gdist}(Q)$. Тогда выполнены следующие неравенства:*

1) $4tn - 2t^2 - 24t \ \leq \ b(Q) \ \leq \ 4tn$;

2) если $t \geq 24$, *то* $b(Q) \ \geq \ 4tn - 2t^2 - 16t$.

Также в [11] показано, что для всех $n \geq 6$ выполняется неравенство

$$b(n) \leq 16n - 64.$$

Обозначим через $i(Q) = |\{x \in Q \mid x \circ x = x\}|$ — количество идемпотентов (см. определение 7) в квазигруппе Q. Основным результатом работы [14] является связь чисел $i(Q)$ и $a(Q)$.

Утверждение 4 ([14, теорема 1.1]). *Для квазигруппы Q выполняется следующее неравенство:*

$$a(Q) \geq 2n - i(Q).$$

В частности, из утверждения 4 следует, что если в квазигруппе Q порядка n число ассоциативных троек $a(Q)$ также равно n (т.е. достигается нижняя граница на число ассоциативных троек для квазигруппы порядка n), то каждый элемент квазигруппы является идемпотентом.

Замечание 3. *Заметим, что с криптографической точки зрения это требование входит в противоречие с требованием отсутствия подквазигрупп [15, 16] (в частности, подквазигрупп размера 1).*

Дальнейшие продвижения были получены в работе [17]. Обозначим через $\delta_L(Q)$ число элементов $a \in Q$, для которых подстановка L_a (см. замечание 1) не имеет неподвижных точек, через $\delta_R(Q)$ число элементов $a \in Q$, для которых подстановка R_a не имеет неподвижных точек.

Утверждение 5 ([17, теорема 2.5]). *Выполнено следующее неравенство:*

$$a(Q) \geq 2n - i(Q) + \delta_L(Q) + \delta_R(Q).$$

Таким образом, если для квазигруппы Q порядка n достигается минимально возможное число ассоциативных троек $a(Q) = n$, то в Q каждый элемент является идемпотентом (т.е., $i(Q) = n$), и у отображений L_a, R_a нет неподвижных точек.

4. Примеры квазигрупп с заданным числом ассоциативных троек

В работах [10, 13] приведены несколько примеров классов квазигрупп с малым числом ассоциативных троек, что позволяет получить верхние оценки на минимальное число ассоциативных троек $a(n)$.

Так, для случая $n \neq 2 \mod 4$ существует коммутативная группа $(G, +)$ и автоморфизм $\phi \in Aut(G)$ со свойством

$$\forall x \in G \setminus \{0\} \quad \phi(x) \neq x.$$

Если n нечетно, то положим $G = \mathbb{Z}_n$, $\phi(x) = 2x$. В случае $n = 2^m$ рассмотрим группу $G = \mathbb{Z}_2 \times \ldots \times \mathbb{Z}_2$ и автоморфизм

$$\phi(x_1, \ldots, x_m) = (x_1 + x_2, x_3, \ldots, x_m, x_1).$$

Наконец, в случае $n = 2^m \cdot d$, где $m \geq 2$, d нечетное, рассмотрим группу $G = \mathbb{Z}_2 \times \ldots \times \mathbb{Z}_2 \times \mathbb{Z}_d$ и автоморфизм

$$\phi(x_1, \ldots, x_m, z) = (x_1 + x_2, x_3, \ldots, x_m, x_1, 2z).$$

Зададим квазигрупповую операцию $\circ$ на множестве G по правилу:

$$x \circ y = \phi(x + y), \quad x, y \in G.$$

В таком случае все тройки (x, y, x) в $(G, \circ)$ являются ассоциативными:

$$\begin{aligned}(x \circ y) \circ x = \phi\left(\phi(x + y) + x\right) = \phi^2(x) + \phi^2(y) + \phi(x) = \\ = \phi\left(x + \phi(x + y)\right) = x \circ (y \circ x).\end{aligned}$$

Других ассоциативных троек в $(G, \circ)$ нет: если (x, y, z) — ассоциативная тройка, то выполняются следующие равенства:

$$\begin{aligned}(x \circ y) \circ z = \phi^2(x) + \phi^2(y) + \phi(z) = \phi(x) + \phi^2(y) + \phi^2(z) = x \circ (y \circ z) \Rightarrow \\ \Rightarrow \phi^2(x - z) = \phi(x - z).\end{aligned}$$

Поскольку $\phi(x) = x$ только при $x = 0$, мы имеем $x = z$.

Для полученной квазигруппы $Q = (G, \circ)$ верно равенство $a(Q) = n^2$, а следовательно, мы имеем:

$$a(n) \leq n^2, \quad n \neq 2 \mod 4.$$

Для случая $n = 2 \mod 4$ можно построить квазигруппу Q с индексом ассоциативности $a(Q) = 2n^2$. Для этого представим n в виде $n = 2d$, d нечетное. Положим $G = \mathbb{Z}_2 \times \mathbb{Z}_d$ и введем операцию покомпонентного сложения в G. Рассмотрим автоморфизм ϕ группы G, заданный по правилу $\phi(a, b) = (a, 2b)$, и операцию $\circ$ на множестве G:

$$(x_1, x_2) \circ (y_1, y_2) = \phi(x_1, x_2) + (y_1, y_2).$$

Операция $\circ$ задает структуру квазигруппы Q на множестве G. Для ассоциативных троек должно выполняться равенство:

$$(x \circ y) \circ z = \phi^2(x) + \phi(y) + z = \phi(x) + \phi(y) + z = x \circ (y \circ z).$$

Следовательно, любая тройка (x, y, z) с условием $\phi(x) = x$ является ассоциативной. Указанное условие выполняется для элементов

$$x = (x_1, 0), \quad x_1 \in \mathbb{Z}_2,$$

y, z — любые элементы G. Следовательно, $a(n) \leq 2n^2, \ n = 2 \bmod 4$.

В работе [10] приведен пример класса квазигрупп размера n, где $n \geq 6$, $n = 0, 2 \bmod 6$, с количеством ассоциативных троек $a(Q) = n^2 - 3n + 3$. Таким образом, в случае $n \geq 6$, $n = 0, 2 \bmod 6$ мы получаем оценку

$$a(n) \leq n^2 - 3n + 3.$$

В ряде статей [18, 19, 20] были получены примеры классов **максимально неассоциативных** квазигрупп, т.е. квазигрупп, для которых $a(Q) = |Q|$. В [18] была дана конструкция на основе т.н. почтиполей (см., например, [21]), из которой следует, что $a(n) = n$ для $n = 2^{6k} \cdot r^2$, где $k \geq 0$, r нечетное. В частности, $a(p^2) = p^2$ для всех нечетных простых p.

Указанный результат был расширен в [19, 20]. Обозначим через $\nu_p(n)$ степень вхождения p в разложение n на простые сомножители. В статье [20] показано, что для n, удовлетворяющих условиям:

$$\nu_p(n) \neq 1, \ p \in \{3, 5, 7, 11\}, \quad \nu_2(n) \neq 2, 4 \text{ и четно},$$

существует максимально неассоциативная квазигруппа порядка n.

В статье [19] показано, что максимально неассоциативная квазигруппа существует для всех достаточно больших порядков n, которые **не имеют** вид $n = 2p_1$ или $n = 2p_1p_2$, где p_1, p_2 — нечетные простые, $p_1 \leq p_2 < 2p_1$. В частности, существует максимально неассоциативная квазигруппа для простых порядков $p \geq 13$.

5. Оценка среднего числа ассоциативных троек

В работе [22] предложен еще один подход к подсчету числа ассоциативных троек в квазигруппах. Как известно (см., например, [23]), ассоциативные тройки можно рассматривать как неподвижные точки коммутатора отображений $[L_a, R_b]$, где $[x, y] = x^{-1}y^{-1}xy$: если (a, x, b) — ассоциативная тройка, то выполняется условие

$$(a \circ x) \circ b = R_b(L_a(x)) = a \circ (x \circ b) = L_a(R_b(x)),$$

то есть x является неподвижной точкой коммутатора: $[L_a, R_b](x) = x$.

В работе [22] предложено оценивать среднее число ассоциативных троек в квазигруппе, где усреднение берется по всем главным изотопам. Обозначим через $Q_{\alpha\beta}$ главный изотоп Q, заданный операцией

$$a * b = \alpha(a) \circ \beta(b).$$

Утверждение 6 ([22, утверждение 2.1]). *Для $n \geq 2$ выполнено следующее равенство:*

$$\frac{1}{(n!)^2} \sum_{\alpha,\beta \in \mathcal{S}_Q} a(Q_{\alpha\beta}) = \frac{n^3}{n-1}.$$

Идея доказательства состоит в подсчете числа неподвижных точек всех коммутаторов $[L_a, R_b]$ для всех главных изотопов, что, в свою очередь, сводится к задаче подсчета суммы $\sum_{\phi,\psi \in \mathcal{S}_Q} |Fix([\phi, \psi])|$, где $Fix(\pi) = \{x \in Q \mid \pi(x) = x\}$ — множество неподвижных точек подстановки π.

Следующее утверждение следует из предыдущего.

Утверждение 7. *Для $n \geq 2$ выполнено следующее равенство:*

$$\frac{1}{(n!)^3} \sum_{\alpha,\beta,\gamma \in \mathcal{S}_Q} a(Q_{\alpha\beta\gamma}) = \frac{n^3}{n-1}.$$

Таким образом, для каждой квазигруппы среднее число ассоциативных троек (при усреднении по всем изотопам) примерно равно n^2.

Утверждение 8 ([22, утверждение 2.3]). *Для $n \geq 2$ выполнено следующее неравенство:*

$$\frac{1}{n!} \sum_{\beta} a(Q_{\alpha\beta}) \geq n^2,$$

и равенство достигается тогда и только тогда, когда α^{-1} — ортоморфизм квазигруппы Q.

6. Минимальное число ассоциативных троек в квазигруппах малого порядка

В ряде работ [13, 14, 22] путем перебора были получены точные значения минимального числа ассоциативных троек $a(n)$ для квазигрупп порядка $n \leq 7$ (см. Табл. 1). Для квазигрупп порядка $n = 8, 9$ число $a(n)$ уже не может быть получено путем полного перебора, поэтому в работах [17, 24] был предложен способ сократить перебор. С помощью ограниченного перебора были получены точные значения чисел $a(8)$, $a(9)$ и получена оценка снизу для $a(10)$ (а именно, было показано, что не существует квазигрупп порядка 10 с индексом ассоциативности 10). Полученные результаты отображены в Табл. 1. Заметим, что полученные значения меньше существующих теоретических оценок, приведенных в разделе 3.

Таблица 1. Минимальное число ассоциативных троек длквазигрупп порядка $n \leq 10$

n	$a(n)$	Работа
1	1	[13]
2	8	[13]
3	9	[13]
4	16	[13]
5	15	[13]
6	16	[13]
7	17	[22]
8	16	[17]
9	9	[17]
10	> 10	[24]

7. Индексы ассоциативности квазигрупп, заданных правильными семействами функций

В цикле работ [25, 26, 27] было предложено задавать таблицу умножения квазигруппы с помощью правильных семейств функций. В настоящем разделе мы рассмотрим один способ задания квазигрупп с помощью правильных семейств и приведем результаты численных экспериментов по вычислению индексов ассоциативности полученных квазигрупп.

7.1. Правильные семейства функций

Определение 11. Пусть $Q_1, \ldots, Q_n$ — набор непустых конечных множеств. Под семейством функций F_n на $Q_1 \times \ldots \times Q_n$ будем понимать отображение $F_n \colon Q_1 \times \ldots \times Q_n \to Q_1 \times \ldots \times Q_n$ вида

$$F_n \colon \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \to \begin{bmatrix} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_n(x_1, \ldots, x_n) \end{bmatrix}, \; f_i(x_1, \ldots, x_n) \colon Q_1 \times \ldots \times Q_n \to Q_i.$$

Число n будем называть размером семейства. Иногда мы будем опускать размер семейства n из обозначения F_n, если он понятен из контекста.

Замечание 4. *Если $Q_1 = \ldots = Q_n = \mathbb{E}_2$, где $\mathbb{E}_2 = \{0, 1\}$, то F_n будем называть семейством булевых функций.*

Определение 12. Семейство функций F_n на $Q_1 \times \ldots \times Q_n$ называется правильным, если для любых двух неравных наборов

$$\alpha = (\alpha_1, \ldots, \alpha_n), \quad \beta = (\beta_1, \ldots, \beta_n), \quad \alpha \neq \beta,$$

выполняется следующее условие:

$$\exists i\colon \alpha_i \neq \beta_i,\ f_i(\alpha) = f_i(\beta).$$

7.2. Критерий правильности в терминах регулярности

Для семейств булевых функций выполняется следующий критерий правильности.

Утверждение 9 ([27, теорема 2])**.** *Семейство булевых функций $F_n(x)$ является правильным тогда и только тогда, когда для любого набора отображений $\Psi = (\psi_1, \ldots, \psi_n)$, $\psi_i\colon \mathbb{E}_2 \to \mathbb{E}_2$ отображение*

$$x \to x \oplus \Psi(F_n(x)) = \begin{bmatrix} x_1 \oplus \psi_1(f_1(x_1, \ldots, x_n)) \\ \vdots \\ x_n \oplus \psi_n(f_n(x_1, \ldots, x_n)) \end{bmatrix}$$

является биекцией $\mathbb{Z}_2^n \to \mathbb{Z}_2^n$.

Указанное утверждение допускает следующее обобщение.

Теорема 1. *Семейство F_n на $Q_1 \times \ldots \times Q_n$, где $(Q_i, \circ_i)$ — квазигруппы, является правильным тогда и только тогда, когда для любого набора отображений $\psi_i\colon Q_i \to Q_i$ следующее отображение биективно:*

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \to x \circ \Psi(F_n(x)) = \begin{bmatrix} x_1 \circ_1 \psi_1(f_1(x_1, \ldots, x_n)) \\ \vdots \\ x_n \circ_n \psi_n(f_n(x_1, \ldots, x_n)) \end{bmatrix},\ x_i \in Q_i.$$

Док-во. Пусть F_n — правильное семейство на $Q_1 \times \ldots \times Q_n$. Покажем, что отображение $x \to x \circ \Psi(F_n(x))$ инъективно. Пусть $x \neq y$, $x, y \in Q_1 \times \ldots \times Q_n$, тогда по условию правильности найдется такой индекс i, что $x_i \neq y_i$, но $f_i(x) = f_i(y)$, а значит,

$$x_i \circ_i \psi_i(f_i(x)) \neq y_i \circ_i \psi_i(f(y)).$$

Из конечности $Q_1 \times \ldots \times Q_n$ и инъективности отображения следует его биективность.

Пусть F не является правильным. Построим отображение Ψ таким образом, чтобы $x \to x \circ \Psi(F_n(x))$ не было биекцией. Поскольку F не является правильным, то найдутся две точки $x \neq y$, для которых для всех индексов i со свойством $x_i \neq y_i$ следует $f_i(x) \neq f_i(y)$. Рассмотрим все индексы, в которых наборы x и y различаются. Для каждого «плохого» индекса зададим ψ_i таким образом, чтобы $x_i \circ_i \psi_i(f_i(x)) = y_i \circ_i \psi_i(f_i(y))$; это можно сделать, зафиксировав $\psi_i(f_i(x))$ как угодно и доопределить

$\psi_i(f_i(y))$ из уравнения (из условия на «плохие» индексы мы имеем $f_i(x) \neq f_i(y)$, а значит, определение ψ_i корректно). В тех индексах, где $x_i = y_i$, зададим ψ_i как правый нейтральный элемент для x_i для любого значения аргумента.

Если мы зададим ψ_i обозначенным выше образом, то получим

$$x \neq y, \quad x \circ \Psi(F_n(x)) = y \circ \Psi(F_n(y)),$$

а значит, отображение не может быть биективным. □

7.3. Один способ построения квазигрупп с помощью правильных семейств

Заметим, что с помощью теоремы 1 можно предложить следующий способ задания квазигруппы. Пусть F, G — два правильных семейства функций размера n над группой $(H^n, +)$ (группа H не обязана быть абелевой). Для $x, y \in H^n$ зададим операцию $\circ$ следующим образом:

$$x \circ y = x + F(x) + y + G(y).$$

Поскольку отображение $x \to \pi_F(x) = x + F(x)$, где F — правильное, является биекцией, то операция $\circ$ задает главный изотоп группы H^n (а значит, задает квазигрупповую операцию).

Замечание 5. *Указанный способ задания квазигруппы отличается от «стандартного» построения на основе одного правильного семейства (см., например, [25, 26]).*

Потребуем дополнительно, чтобы группа H^n была коммутативной, и рассмотрим условие на ассоциативность тройки (x, y, z) в квазигруппе Q, построенной по паре правильных семейств (F, G):

$$(x \circ y) \circ z = (x + F(x)) + (y + G(y)) + (z + G(z)) + F\left(x + F(x) + y + G(y)\right),$$
$$x \circ (y \circ z) = (x + F(x)) + (y + F(y)) + (z + G(z)) + G\left(y + F(y) + z + G(z)\right),$$

и из условия $(x \circ y) \circ z = x \circ (y \circ z)$ получаем, что:

$$F(y) - G(y) = F\left(x + F(x) + y + G(y)\right) - G\left(y + F(y) + z + G(z)\right). \quad (1)$$

Из подобного эквивалентного представления относительно легко следуют два наблюдения, которые могут быть доказаны прямой проверкой.

Утверждение 10. *Тройка (x, y, z) является ассоциативной в квазигруппе Q, построенной по паре семейств (F, G), тогда и только тогда, когда тройка (z, y, x) является ассоциативной в квазигруппе Q', построенной по паре семейств (G, F).*

В частности, индексы ассоциативности квазигрупп, построенных по парам семейств (F, G) и по парам семейств (G, F), совпадают.

Утверждение 11. *Пусть $\mathcal{A}$ — такое обратимое линейное отображение (т.е. $\mathcal{A}(x+y) = \mathcal{A}(x) + \mathcal{A}(y)$), что семейства*

$$F'(x) = \mathcal{A}^{-1}\left(F\left(\mathcal{A}(x)\right)\right), \quad G'(y) = \mathcal{A}^{-1}\left(G\left(\mathcal{A}(y)\right)\right)$$

также являются правильными (так, в качестве $\mathcal{A}$ можно рассмотреть преобразование обратимой линейной перекодировки, см. [28]). В таком случае (x, y, z) является ассоциативной тройкой для квазигруппы, построенной по паре правильных семейств (F, G), тогда и только тогда, когда тройка $\left(\mathcal{A}^{-1}(x), \mathcal{A}^{-1}(y), \mathcal{A}^{-1}(z)\right)$ является ассоциативной для квазигруппы, построенной по паре правильных семейств (F', G').

В частности, индексы ассоциативности квазигрупп, построенных по парам семейств (F, G) и (F', G'), совпадают.

В случае $A^n = \mathbb{Z}_2^n$ выполняется несколько дополнительных свойств.

Утверждение 12. *Тройка (x, y, z) является ассоциативной для квазигруппы, построенной по паре правильных семейств (F, G), тогда и только тогда, когда она является ассоциативной для квазигруппы, построенной по паре правильных семейств $(F \oplus \alpha, G \oplus \alpha)$, где $\alpha \in \mathbb{Z}_2^n$.*

Утверждение 13. *Количество ассоциативных троек в квазигруппе, построенной по паре правильных булевых семейств (F, G), четно.*

*Док-во.*Зафиксируем значения x, y и найдем все значения z, которые удовлетворяют требованию ассоциативности (1):

$$F(y) \oplus G(y) = F\left(x \oplus F(x) \oplus y \oplus G(y)\right) \oplus G\left(y \oplus F(y) \oplus z \oplus G(z)\right).$$

После фиксации x, y, мы получим уравнение на z вида

$$G\left(z \oplus G(z) \oplus \alpha\right) = \beta, \quad \alpha, \beta \in \mathbb{Z}_2^n. \tag{2}$$

Как было показано ранее [29, теорема 7], уравнение вида $G(t) = \beta$ всегда имеет четное число решений для булевых правильных семейств. Поскольку отображение $z \to z \oplus G(z) \oplus \alpha$ является биекцией, для каждой фиксации переменных x, y уравнение (2) будет иметь четное число решений z. Тем самым мы получим четное число ассоциативных троек. □

Указанные свойства могут быть использованы при исследовании индексов ассоциативности квазигрупп, построенных по различным парам правильных семейств.

7.4. Индексы ассоциативности для квазигрупп, построенных по правильным булевым семействам малых размеров

Приведем результаты численных экспериментов. Для $n = 2$ имеется 12 правильных булевых семейств, с помощью которых можно задать $12^2 = 144$ квазигруппы (используя конструкцию, описанную в разделе 7.3). Для $n = 3$ имеется 744 правильных булевых семейства, с помощью которых можно задать $744^2 = 553536$ квазигрупп. Все порождаемые квазигруппы будут попарно различны: если $F \neq G$, то для некоторого x имеем $\pi_F(x) = x \oplus F(x) \neq x \oplus G(x) = \pi_G(x)$. Результаты численных экспериментов для $n = 2$ приведены в Табл. 2, для $n = 3$ — приведены в Табл. 3 и на рис. 1.

Таблица 2. Число квазигрупп с заданным $a(Q)$ для квазигрупп, построенных по правильным булевым семействам размера $n = 2$

$a(Q)$	Кол-во Q
16	32
32	96
64	16

Для $n = 4$ был проведен статистический эксперимент. Случайно равновероятно (среди всех возможных пар) выбирались $N = 10^5$ пар правильных семейств, по каждой паре строилась квазигруппа, подсчитывался индекс ассоциативности полученной квазигруппы. Была построена ядерная оценка плотности полученной случайной величины, результат приведен на рис. 2.

Заметим, что при $n = 2$ достигается минимально возможное значение индекса ассоциативности для квазигрупп порядка 4 (а именно 16). При $n \geq 3$ все полученные индексы ассоциативности существенно превышают теоретически возможные для квазигрупп заданного порядка. Отметим также, что во всех исследованных случаях $n = 2, 3, 4$ минимально достижимый индекс ассоциативности у построенных квазигрупп оказался равным квадрату порядка квазигруппы, в связи с чем можно выдвинуть гипотезу, что у квазигрупп, построенным по парам правильных булевых семейств размера n число ассоциативных троек не может быть меньше, чем 2^{2n}.

Для $n = 3$ также был проведен следующий эксперимент. Все 744 правильных семейства были разбиты на 10 классов эквивалентности относительно изометрий пространства Хэмминга (см. [28]). Затем для каждой пары классов эквивалентности $(\mathcal{F}, \mathcal{G})$ перебирались все пары представителей $F \in \mathcal{F}$, $G \in \mathcal{G}$ и вычислялся индекс ассоциативности

Таблица3. Число квазигрупп с заданным $a(Q)$ для квазигрупп, построенных по правильным булевым семействам размера $n = 3$

$a(Q)$	Кол-во Q
64	27648
80	103424
88	18432
96	82944
104	33792
112	21504
120	21504
128	116352

$a(Q)$	Кол-во Q
144	3072
160	84480
176	6144
192	18432
208	3072
256	10368
320	2304
512	64

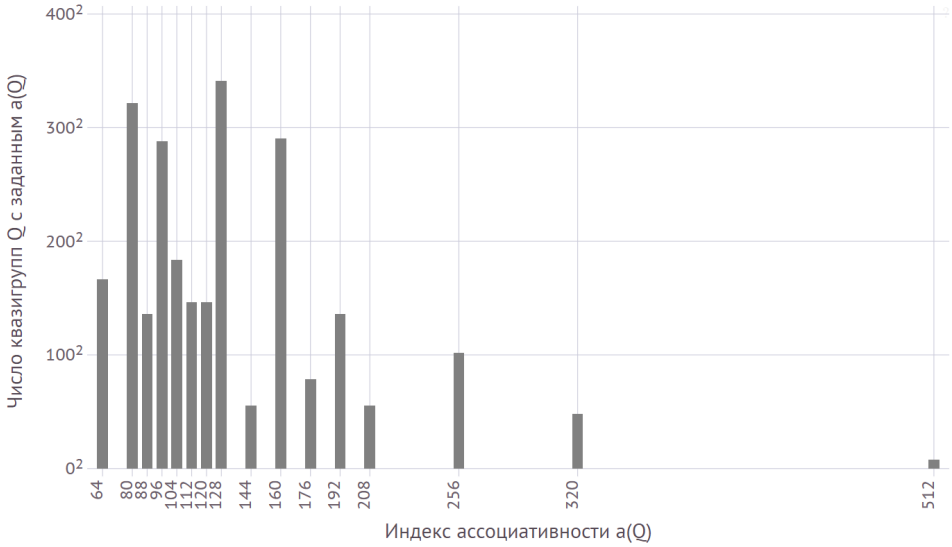


Рисунок 1. Распределение числа квазигрупп с заданным $a(Q)$ для $n = 3$

квазигруппы, порождаемой парой правильных булевых семейств (F, G), после чего вычислялся «средний индекс ассоциативности» для пары классов эквивалентности $(\mathcal{F}, \mathcal{G})$. Результаты эксперимента отображены на рис. 3. Из приведенной тепловой карты видно, что наиболее неассоциативные квазигруппы порождаются при использовании 6-го класса эквивалентности, представителем которого является, например, семейство

$$(x_2x_3,\ x_1 \oplus x_1x_3,\ x_1 \oplus x_2 \oplus x_1x_2)\,. \tag{3}$$

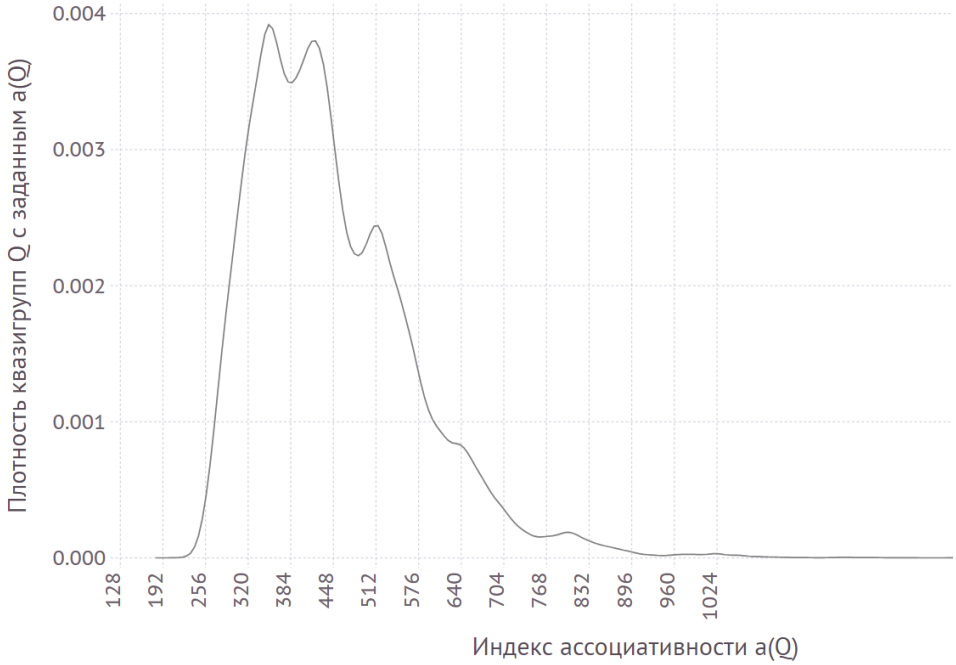


Рисунок 2. Оценка плотности распределения квазигрупп, построенных по парам правильных булевых семейств, с заданным $a(Q)$ для $n = 4$

Заметим, что класс правильных семейств, к которому принадлежит указанный представитель, отдельно изучался ранее (см. [29, раздел 4]); в частности, было отмечено, что «канонические» представители рассматриваемого семейства сильно квадратичны при нечетных n [30, теорема 1] и имеют полный граф существенной зависимости.

8. Заключение

В настоящей работе были рассмотрены основные результаты, касающиеся оценок числа ассоциативных троек в квазигруппах. Также был рассмотрен один способ построения квазигрупп на основе правильных семейств функций и получен ряд утверждений об индексах ассоциативности получаемых квазигрупп, приведены результаты вычислительных экспериментов для размеров семейств $n = 2, 3, 4$.

В качестве дальнейших направлений исследований можно выделить следующие:

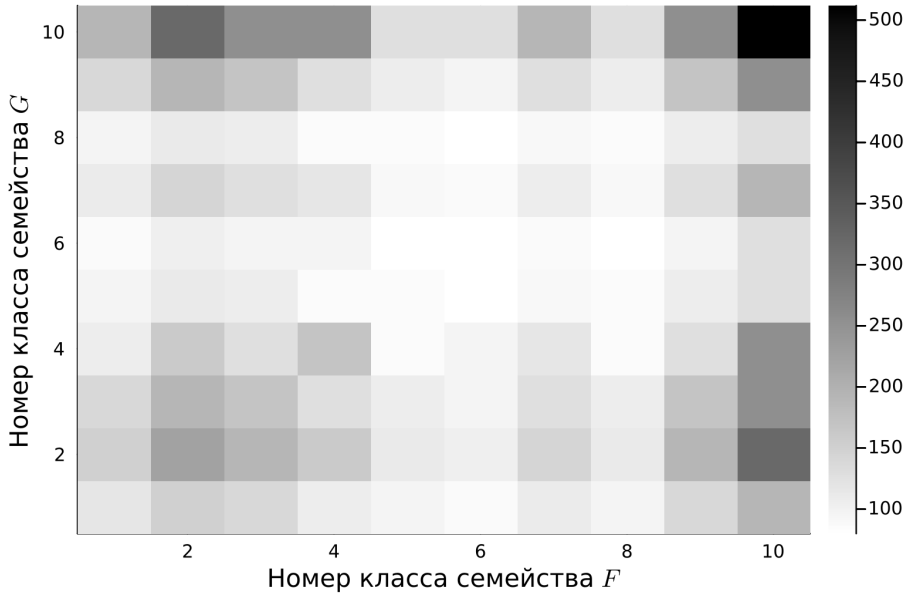


Рисунок 3. Тепловая карта для среднего индекса ассоциативности, усреднение берется по представителям классов эквивалентности, $n = 3$

- исследование алгебраических свойств квазигрупп, порождаемых семействами вида (3);
- исследование связи индекса ассоциативности и графа существенной зависимости семейства (чем «ближе» граф к полному на n вершинах, тем меньше «тривиальных» соотношений в уравнении ассоциативности, а значит, меньше индекс ассоциативности);
- дальнейшее исследование связи индексов ассоциативности квазигрупп, построенных по эквивалентным (в некотором смысле) парам правильных семейств.

Автор выражает признательность научному руководителю А. Е. Панкратьеву и А. В. Галатенко за оказанную помощь при написании настоящей статьи.

Список литературы

[1] J. Denes, A. Keedwell, *Latin squares and their applications (2nd Edition)*, North Holland, 2015, 428 pp.

[2] М. М. Глухов, "О применениях квазигрупп в криптографии", *Прикладная дискретная математика*, 2008, № 2, 28–32.

[3] D. Chauhan, I. Gupta, R. Verma, "Quasigroups and their applications in cryptography", *Cryptologia*, **45**:3 (2021), 227–265.

[4] V. A. Shcherbacov, *Elements of Quasigroup Theory and Applications (1st Edition)*, Chapman and Hall/CRC, 2017, 598 pp.

[5] М. Э. Тужилин, "Латинские квадраты и их применение в криптографии", *Прикладная дискретная математика*, 2012, № 3(17), 47–52.

[6] V. Valent, *Quasigroups with few associative triples*, Bachelor thesis, Univerzita Karlova, Matematicko-fyzikální fakulta, 2016.

[7] T. Kepka, "A note on associative triples of elements in cancellation groupoids", *Commentationes Mathematicae Universitatis Carolinae*, **21**:3 (1980), 479–487.

[8] T. Kepka, "Notes on associative triples of elements in commutative groupoids", *Acta Universitatis Carolinae. Mathematica et Physica*, **22**:2 (1981), 39–47.

[9] A. Drбpal, T. Kepka, "A note on the number of associative triples in quasigroups isotopic to groups", *Commentationes Mathematicae Universitatis Carolinae*, **22**:4 (1981), 735–743.

[10] A. Kotzig, C. Reischer, "Associativity index of finite quasigroups", *Glasnik Matematicki Series III*, **18**:38 (1983), 243–253.

[11] A. Drбpal, "On quasigroups rich in associative triples", *Discrete Mathematics*, **44**:3 (1983), 251–265.

[12] В.Д. Белоусов, *Основы теории квазигрупп и луп*, Наука, 1967, 224 с.

[13] J. Jeəek, T. Kepka, "Notes on the number of associative triples", *Acta Universitatis Carolinae. Mathematica et Physica*, **31**:1 (1990), 15–19.

[14] O. Groўek, P. Horбk, "On quasigroups with few associative triples", *Designs, Codes and Cryptography*, **64**:1-2 (2012), 221–227.

[15] D. Gligoroski, S. Markovski, L. Kocarev, "Edon-R, An Infinite Family of Cryptographic Hash Functions", *International Journal of Security and Networks*, **8**:3 (2009), 293–300.

[16] В. А. Артамонов, "Квазигруппы и их приложения", *Чебышевский сборник*, **19**:2 (2018), 111–122.

[17] A. Drбpal, V. Valent, “High nonassociativity in order 8 and an associative index estimate”, *Journal of Combinatorial Designs*, **27**:4 (2019), 205–228.

[18] A. Drбpal, P. Lisonжk, “Maximal nonassociativity via nearfields”, *Finite Fields and Their Applications*, **62** (2020), 101610.

[19] A. Drбpal, I. Wanless, “Maximally nonassociative quasigroups via quadratic orthomorphisms”, *Algebraic Combinatorics*, **4**:3 (2021), 501–515.

[20] P. Lisonжk, “Maximal nonassociativity via fields”, *Designs, Codes and Cryptography*, **88**:12 (2020), 2521–2530.

[21] Ю. Ионин, “Конечные проективные плоскости”, *Математическое просвещение*, **13** (2009), 50–79.

[22] A. Drбpal, V. Valent, “Few associative triples, isotopisms and groups”, *Designs, Codes and Cryptography*, **86**:3 (2018), 555–568.

[23] V. Artamonov, S. Chakrabarti, S. K. Pal, “Characterizations of highly non-associative quasigroups and associative triples”, *Quasigroups and Related Systems*, **25**:1 (2017), 1–19.

[24] A. Drбpal, V. Valent, “Extreme nonassociativity in order nine and beyond”, *Journal of Combinatorial Designs*, **28**:1 (2020), 33–48.

[25] В. А. Носов, “Построение классов латинских квадратов в булевой базе данных”, *Интеллектуальные системы*, **4**:3–4 (1999), 307–320.

[26] В. А. Носов, “Построение параметрического семейства латинских квадратов в векторной базе данных”, *Интеллектуальные системы*, **8**:1–4 (2006), 517–529.

[27] В. А. Носов, А. Е. Панкратьев, “Латинские квадраты над абелевыми группами”, *Фундаментальная и прикладная математика*, **12**:3 (2006), 65–71.

[28] А. В. Галатенко, А. Е. Панкратьев, К. Д. Царегородцев, “Об одном критерии правильности семейства функций”, *Фундаментальная и прикладная математика*, **24**:4 (2023), 61–73.

[29] К. Д. Царегородцев, “О свойствах правильных семейств булевых функций”, *Дискретная математика*, **33**:1 (2021), 91–102.

[30] А. В. Галатенко, В. А. Носов, А. Е. Панкратьев, К. Д. Царегородцев, “О порождении n-квазигрупп с помощью правильных семейств функций”, *Дискретная математика*, **35**:1 (2023), 35–53.

On the associativity index of finite quasigroups
Tsaregorodtsev K. D.

In this paper we review the results on the number of associative triples in generic quasigroups and quasigroups from restricted classes. Lower and upper bounds on the number of triples are considered, and experimental results on the associativity index of quasigroups generated by proper families of boolean functions are provided.

Keywords: associative triple, quasigroup, proper family of boolean functions.

References

[1] J. Denes, A. Keedwell, *Latin squares and their applications (2nd Edition)*, North Holland, 2015, 428 pp.

[2] M. M. Glukhov, "On the applications of quasi-groups in cryptography", *Applied Discrete Mathematics*, 2008, №2, 28–32 (In Russian).

[3] D. Chauhan, I. Gupta, R. Verma, "Quasigroups and their applications in cryptography", *Cryptologia*, **45**:3 (2021), 227–265.

[4] V. A. Shcherbacov, *Elements of Quasigroup Theory and Applications (1st Edition)*, Chapman and Hall/CRC, 2017, 598 pp.

[5] M. E. Tuzhilin, "Latin squares and their application in Cryptography", *Applied Discrete Mathematics*, 2012, №3(17), 47–52 (In Russian).

[6] V. Valent, *Quasigroups with few associative triples*, Bachelor thesis, Univerzita Karlova, Matematicko-fyzikální fakulta, 2016.

[7] T. Kepka, "A note on associative triples of elements in cancellation groupoids", *Commentationes Mathematicae Universitatis Carolinae*, **21**:3 (1980), 479–487.

[8] T. Kepka, "Notes on associative triples of elements in commutative groupoids", *Acta Universitatis Carolinae. Mathematica et Physica*, **22**:2 (1981), 39–47.

[9] A. Drápal, T. Kepka, "A note on the number of associative triples in quasigroups isotopic to groups", *Commentationes Mathematicae Universitatis Carolinae*, **22**:4 (1981), 735–743.

[10] A. Kotzig, C. Reischer, "Associativity index of finite quasigroups", *Glasnik Matematicki Series III*, **18**:38 (1983), 243–253.

[11] A. Drápal, "On quasigroups rich in associative triples", *Discrete Mathematics*, **44**:3 (1983), 251–265.

[12] V. D. Belousov, *Foundations of the theory of quasigroups and loops*, Nauka, 1967 (In Russian), 224 pp.

[13] J. Ježek, T. Kepka, "Notes on the number of associative triples", *Acta Universitatis Carolinae. Mathematica et Physica*, **31**:1 (1990), 15–19.

[14] O. Grošek, P. Horák, "On quasigroups with few associative triples", *Designs, Codes and Cryptography*, **64**:1-2 (2012), 221–227.

[15] D. Gligoroski, S. Markovski, L. Kocarev, "Edon-R, An Infinite Family of Cryptographic Hash Functions", *International Journal of Security and Networks*, **8**:3 (2009), 293–300.

[16] V. A. Artamonov, "Quasigroups and their applications", *Chebyshevskii Sbornik*, **19**:2 (2018), 111–122 (In Russian).

[17] A. Drápal, V. Valent, "High nonassociativity in order 8 and an associative index estimate", *Journal of Combinatorial Designs*, **27**:4 (2019), 205–228.

[18] A. Drápal, P. Lisoněk, "Maximal nonassociativity via nearfields", *Finite Fields and Their Applications*, **62** (2020), 101610.

[19] A. Drápal, I. Wanless, "Maximally nonassociative quasigroups via quadratic orthomorphisms", *Algebraic Combinatorics*, **4**:3 (2021), 501–515.

[20] P. Lisoněk, "Maximal nonassociativity via fields", *Designs, Codes and Cryptography*, **88**:12 (2020), 2521–2530.

[21] Y. Ionin, "Finite projective planes", *Matematicheskoye prosveshcheniye*, **13** (2009), 50–79 (In Russian).

[22] A. Drápal, V. Valent, "Few associative triples, isotopisms and groups", *Designs, Codes and Cryptography*, **86**:3 (2018), 555–568.

[23] V. Artamonov, S. Chakrabarti, S. K. Pal, "Characterizations of highly non-associative quasigroups and associative triples", *Quasigroups and Related Systems*, **25**:1 (2017), 1–19.

[24] A. Drápal, V. Valent, "Extreme nonassociativity in order nine and beyond", *Journal of Combinatorial Designs*, **28**:1 (2020), 33–48.

[25] V. A Nosov, "Construction of classes of Latin squares in a Boolean database", *Intelligent Systems (Intellektualnye Sistemy)*, **4**:3–4 (1999), 307–320 (In Russian).

[26] V. A Nosov, "Construction of a parametric family of Latin squares in a vector database", *Intelligent Systems (Intellektualnye Sistemy)*, **8**:1–4 (2006), 517–529 (In Russian).

[27] V. A. Nosov, A. E Pankratiev, "Latin squares over Abelian groups", *Journal of Mathematical Sciences*, **163**:5 (2009), 53–542.

[28] A. V. Galatenko, A. E Pankratiev, K. D. Tsaregorodtsev, "A Criterion of Properness for a Family of Functions", *Journal of Mathematical Sciences*, 2024.

[29] K. D. Tsaregorodtsev, "Properties of proper families of Boolean functions", *Discrete Mathematics and Applications*, **32**:5 (2022), 369–378.

[30] A. V. Galatenko, V. A. Nosov, A. E Pankratiev, K. D. Tsaregorodtsev, "On the generation of n-quasigroups using proper families of functions", *Diskretnaya Matematika*, **35**:1 (2023), 35–53 (In Russian).

Часть 3
Математические модели

Быстрые алгоритмы умножения и деления натуральных чисел с помощью клеточных автоматов с локаторами

Э. Э. Гасанов[1], Б. Ф. Хайбуллин[2]

Для умножения и деления n-значных натуральных чисел известны алгоритмы со сложностью порядка $n^{\log_2 3}$ и даже порядка $n^{\log n}$. В данной работе предложен алгоритм умножения n-значных натуральных чисел за $2n+2$ такта. Здесь под значностью числа a понимается число $]\log_2 a[$. Для деления натуральных чисел с остатком предложен алгоритм с временем работы $3n+8$ тактов, где n — значность частного. Предложенные алгоритмы в качестве вычислителей используются двумерные клеточные автоматы с локаторами.

Ключевые слова: умножение натуральных чисел, деление натуральных чисел, клеточные автоматы с локаторами.

Введение

Пусть a и b два натуральных числа, двоичная запись которых содержит по порядку n разрядов. Наиболее известный и быстрый алгоритм умножения таких чисел был предложен А.А.Карацубой [1], и он имеет сложность $O(n^{\log_2 3})$. Более быстрым по порядку алгоритмом умножения, является алгоритм Шёнхаге-Штрассена [2]. Его сложность $O(n \cdot \log n \cdot \log\log n)$. Но на практике алгоритм Шёнхаге-Штрассена быстрее алгоритма Карацубы, только если значность числа более 10 тысяч. Еще более быстрым по порядку является алгоритм Фюрера [3], но его преимущество может проявится при значности чисел более 10^{13}. Относительно недавно появился алгоритм Харвея-Хоевена [4] со сложностью $O(n \log n)$.

Для деления натуральных чисел с остатком известен алгоритм Бурникеля-Циглера [5]. Он использует внутри себя алгоритм умножения. Если в качестве алгоритма умножения взять алгоритм Карацубы, то вычислительная сложность алгоритма Бурникеля-Циглера будет $O(n^{\log_2 3})$, а если использовать алгоритм Шёнхаге-Штрассена, то сложность алгоритма Бурникеля-Циглера будет $O(n \cdot \log^2 n \cdot \log\log n)$.

[1] *Гасанов Эльяр Эльдарович* — зав. каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: el_gasanov@mail.ru.

Gasanov Elyar Eldarovich — Head of Chair Mathematical Theory of Intellegent Systems, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics.

[2] *Хайбуллин Бакир Фаридович* — ведущий программист в ООО "Elius", г. Ташкент, Узбекистан, e-mail: bakir_k@mail.ru.

Khaybullin Bakir Faridovich — lead programmer at Elius LLC, Tashkent, Uzbekistan.

В данной работе предлагаются алгоритмы решения задач умножения и деления с остатком n-значных натуральных чисел с помощью клеточных автоматов с локаторами.

Приведем неформальное описание двумерного клеточного автомата с локаторами.

Расположим в каждой клетке плоской решетки $\mathbb{Z}^2$ один и тот же автомат с локаторами. Понятие локатора определим чуть позже, сейчас важно, что каждый локатор в каждый момент принимает некоторое значение. Автомат имеет функцию перехода, которая по состоянию соседей автомата и по значениям локаторов в текущей момент определяет состояние автомата в следующий момент. Кроме того, у автомата есть функция вещания, которая по состояниям соседей автомата и по значениям локаторов вычисляет сигнал вещания, который передается в эфир. Сигналы вещания образуют конечную аддитивную коммутативную полугруппу, а эфир представляет собой потенциально бесконечный сумматор сигналов элементарных автоматов, где в качестве оператора суммы выступает определяющая операция данной полугруппы. Каждый локатор представляет собой некоторый телесный угол с вершиной в позиции автомата, а значением локатора в текущий момент является сумма сигналов вещания всех автоматов, попадающих в этот телесный угол. Отметим, что в область суммирования локатора не входит вершина телесного угла. т.е. мы сигнал вещания, посылаемый данным автоматом, не включаем в сумму.

В наших алгоритмах будут использоваться один полный локатор, который представляет собой двумерную плоскость с выколотым началом координат, и 8 локаторов, представляющих собой лучи, направленные на север, северо-восток, восток, юго-восток, юг, юго-запад, запад и северо-запад.

В работе показано, что с помощью таких двумерных клеточных автоматов с локаторами можно решить задачу умножения и деления n-разрядных чисел за время порядка n.

Ранее похожий алгоритм умножения был доложен на конференции [6].

1. Постановка задачи и формулировка результатов

Понятие клеточного автомата с локаторами введено в работе Э. Э. Гасанова [7]. В работе Г. В. Калачева [8] были выявлены некоторые неточности, приведенного в [7] определения. Точное формальное определение клеточного автомата с локаторами можно найти в работах Д. Э. Ибрагимовой [9] и Э. Э. Гасанова [10]. Здесь мы не будем приводить это определение, а дадим определение двумерного клеточного автомата с

9 локаторами, с помощью которого задачу умножения и деления чисел можно решить за линейное время.

В общем случае локатор — это телесный угол, границы которого являются частями гиперплоскостей, задаваемых линейными уравнениями с целыми коэффициентами. В нашем случае мы будем рассматривать множество из 9 телесных углов

$$L = \{\Omega, \mathcal{N}, \mathcal{NE}, \mathcal{E}, \mathcal{SE}, \mathcal{S}, \mathcal{SW}, \mathcal{W}, \mathcal{NW}\}, \tag{1}$$

где $\Omega = \mathbb{Z}^2 \setminus \{(0,0)\}$ — называется полным, $\mathbb{Z}^2$ — множество двумерных векторов с целыми координатами, $\mathcal{N} = \{(x,y) : x = 0, y > 0\}$ — назовем "север", $\mathcal{NE} = \{(x,y) : y = x, x > 0\}$ — назовем "северо-восток", $\mathcal{E} = \{(x,y) : y = 0, x > 0\}$ — назовем "восток", $\mathcal{SE} = \{(x,y) : y = -x, x > 0\}$ — назовем "юго-восток", $\mathcal{S} = \{(x,y) : x = 0, y < 0\}$ — назовем "юг", $\mathcal{SW} = \{(x,y) : y = x, x < 0\}$ — назовем "юго-запад", $\mathcal{W} = \{(x,y) : y = 0, x < 0\}$ — назовем "запад", $\mathcal{NW} = \{(x,y) : y = -x, x < 0\}$ — назовем "северо-запад".

Двумерным клеточным автоматом с 9 локаторами называется восьмерка $\sigma = (\mathbb{Z}^2, Q, V, G, +, L, \varphi, \psi)$, где Q — некоторое конечное множество, называемое *множеством состояний*; в множестве Q выделено одно состояние q_0, называемое *состоянием покоя*; $V = (\alpha_1, \ldots, \alpha_{h-1})$ — упорядоченный набор попарно различных векторов из $\mathbb{Z}^2$; G — некоторое конечное множество, "+" — операция на G такая, что $(G, +)$ —коммутативная полугруппа с нейтральным элементом $e \in G$; L — упорядоченный набор телесных углов, задаваемых выражением (1); φ — функция, зависящая от переменных $x_0, x_1, \ldots, x_{h-1}, z_0, z_1, \ldots, z_8$; $\varphi : Q^h \times G^9 \to Q$, $\varphi(q_0, e) = q_0$; $q_0 = (q_0, \ldots, q_0) \in Q^h$, $e = (e, \ldots, e) \in G^9$; ψ — функция зависящая от переменных $x_0, x_1, \ldots, x_{h-1}, z_0, z_1, \ldots, z_8$; $\psi : Q^h \times G^9 \to G$; $\psi(q_0, e) = e$. Элементы множества $\mathbb{Z}^2$ называются *ячейками* клеточного автомата σ; элементы множества Q называются *состояниями ячейки* клеточного автомата σ; набор V называется *шаблоном соседства* клеточного автомата σ; элементы множества G называются *сигналами вещания*; набор L называется *шаблоном локаторов* клеточного автомата σ; функция φ называется *локальной функцией переходов* автомата σ; функция ψ называется *функцией вещания* автомата σ; переменные $x_0, x_1, \ldots, x_{h-1}$ принимают значения из Q, переменные $z_0, z_1, \ldots, z_8$ принимают значения из G. Состояние q_0 интерпретируется как *состояние покоя*, а условие $\varphi(q_0, e) = q_0$ — как *условие сохранения состояния покоя*. Ячейки, находящиеся в состоянии отличном от q_0, будем называть *активными*. Условие $\psi(q_0, e) = e$ означает, что ячейка в состоянии покоя, не имеющая активных соседей и не получающая сигналов из эфира посылает в эфир нейтральный элемент, что можно интерпретировать как то, что она не посылает сигналы в эфир.

Здесь нам нужно было вводить упорядочение шаблона соседства V и шаблона локаторов L для того, чтобы установить взаимно однозначное соответствие между векторами из V и телесными углами из L и переменными локальной функции переходов φ и функции вещания ψ соответственно $x_0, x_1, \ldots, x_{h-1}$ и $z_0, z_1, \ldots, z_8$. Это соответствие можно сделать более явным, если индексировать переменные функций φ и ψ самими векторами и телесными углами, т.е. считать, что локальная функция переходов φ и функции вещания ψ зависят от переменных $x_0, x_{\alpha_1}, \ldots, x_{\alpha_{h-1}}, z_\Omega, z_\mathcal{N} \ldots, z_{\mathcal{NW}}$, здесь индекс первой переменной есть нулевой вектор $0 = (0,0) \in \mathbb{Z}^2$. Если договориться так индексировать переменные локальной функции переходов и функции вещания, то их можно записывать в любом порядке, и тогда можно воспринимать шаблон соседства и шаблон локаторов как просто множества, а не упорядоченный набор. В дальнейшем мы будем индексировать переменные локальной функции переходов и функции вещания векторами из шаблона соседства и телесными углами из шаблона локаторов.

При этом мы часто будем опускать в индексах внешние круглые скобки у векторов. Например, если $h = 2$, $q = 2$ и $V = \{(-1,0),(1,0)\}$, то пример локальной функции переходов может выглядеть так: $\varphi = x_{-1,0} \& z_\Omega \vee x_{1,0} \& z_\mathcal{N}$.

Если $\alpha \in \mathbb{Z}^2$ и ν — телесный угол из L, то через $\nu(\alpha)$ обозначим телесный угол, полученный параллельным переносом телесного угла ν на вектор α, т.е. вершиной телесного угла $\nu(\alpha)$ является точка α.

Если $\alpha \in \mathbb{Z}^2$ — ячейка клеточного автомата σ, то множество $V(\alpha) = \{\alpha, \alpha+\alpha_1, \ldots, \alpha+\alpha_{h-1}\}$ называется *окрестностью ячейки* α, а множество $L(\alpha) = \{\Omega(\alpha), \mathcal{N}, \ldots, \mathcal{NW}(\alpha_m)\}$ называется *локаторами ячейки* α.

Состоянием клеточного автомата с локаторами σ назовем пару (g, f), где g — произвольная функция, определенная на множестве Z^2, принимающая значения из G, называемая *состоянием эфира*, f — произвольная функция, определенная на множестве Z^2, принимающая значения из Q и называемая *распределением состояний клеточного автомата с локаторами* σ. Такую пару функций можно интерпретировать как некую мозаику, получающуюся в двумерном пространстве приписыванием каждой точке с целочисленными координатами некоторого сигнала из G и некоторого состояния из Q. Множество всевозможных состояний клеточного автомата с локаторами обозначим Σ.

Если $\alpha \in \mathbb{Z}^2$, (g, f) — состояние клеточного автомата с локаторами σ, то значение $g(\alpha)$ назовем *сигналом ячейки* α, определяемым состоянием (g, f), а значение $f(\alpha)$ — *состоянием ячейки* α, определяемым состоянием (g, f).

Для каждого $\nu \in L$

$$s_\nu(\alpha) = \sum_{\beta \in \nu(\alpha) \cap \mathbb{Z}^2} g(\beta) \tag{2}$$

назовем *значением локатора* ν, определяемым состоянием (g, f). Здесь суммирование сигналов осуществляется с помощью определяющей операции $+$ полугруппы G. Отметим, что в формулах (2) используются формально бесконечные суммы, и, чтобы они были определены, мы либо будем считать, что только конечное число слагаемых в суммах отлично от нейтрального элемента, либо предположим, что полугруппа $(G, +)$ является идемпотентным моноидом, т.е. для любого $h \in G$ выполнено $h + h = h$.

На множестве Σ определим *глобальную функцию переходов* Φ_σ клеточного автомата с локаторами σ, полагая $\Phi_\sigma(g, f) = (g', f')$, где $(g, f), (g', f') \in \Sigma$ и для любой ячейки $\alpha \in \mathbb{Z}^k$ выполняются тождества

$$f'(\alpha) = \varphi(f(\alpha), f(\alpha + \alpha_1), \ldots, f(\alpha + \alpha_{h-1}), s_\Omega(\alpha), s_\mathcal{N} \ldots, s_{\mathcal{N}\mathcal{W}}(\alpha)), \tag{3}$$

$$g'(\alpha) = \psi(f(\alpha), f(\alpha + \alpha_1), \ldots, f(\alpha + \alpha_{h-1}), s_\Omega(\alpha), s_\mathcal{N} \ldots, s_{\mathcal{N}\mathcal{W}}(\alpha)). \tag{4}$$

Содержательная интерпретация отображения Φ_σ такова, что сигнал каждой ячейки и состояние каждой ячейки "после перехода" определяется по состоянию упорядоченной окрестности ячейки и по значениям локаторов "до перехода" с помощью законов φ и ψ одинаково для всех ячеек.

Поведениями клеточного автомата с локаторами σ назовем такие последовательности $(g_0, f_0), (g_1, f_1), (g_2, f_2), \ldots$ его состояний, для которых выполняется $(g_{i+1}, f_{i+1}) = \Phi_\sigma(g_i, f_i)$ для всех $i = 0, 1, 2, \ldots$, причем (g_i, f_i) называется *состоянием клеточного автомата с локаторами* σ *в момент* i, а (g_0, f_0) называется *начальным состоянием клеточного автомата с локаторами* σ.

Состояние клеточного автомата, у которого лишь конечное число ячеек находится в отличном от состояния покоя g_0, и сигналы лишь конечного числа ячеек не равны нейтральному элементу e, назовем *конфигурацией*. Множество конфигураций будем обозначать через Σ'.

Определим задачу умножения чисел a и b для клеточного автомата с локаторами. В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами $(0, 0)$ находится в состоянии, которое можно назвать “начало координат”, ячейка с координатами $(a, 0)$ находится в состоянии, которое можно назвать “первый сомножитель”, а ячейка с координатами $(0, b)$ находится в состоянии, которое можно назвать “второй сомножитель”. Клеточный автомат решает задачу умножения чисел, если в финальной

конфигурации ячейка с координатами $(a \cdot b, 0)$ перейдет в состояние "результат умножения", а все остальные ячейки, кроме $(0, 0)$, перейдут в состояние покоя.

Справедлива следующая теорема, доказанная Э. Э. Гасановым.

Теорема 1. *Существует двумерный клеточный автомат с 9 локаторами, который решает задачу умножения чисел a и b за время* $2]\log_2 a[+2$.

Здесь если x — вещественное число, то $]x[$ — это наименьшее целое неменьшее чем x.

Определим задачу деления чисел a и b с остатком для клеточного автомата с локаторами. В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами $(0, 0)$ находится в состоянии, которое можно назвать "начало координат", ячейка с координатами $(a, 0)$ находится в состоянии, которое можно назвать "делимое", а ячейка с координатами $(0, b)$ находится в состоянии, которое можно назвать "делитель". Пусть $c = \lfloor a/b \rfloor$ — целая часть от деления a на b, $d = a \mod b$ — отстаток от деления a на b. Клеточный автомат решает задачу деления чисел, если в финальной конфигурации ячейка с координатами $(c, 0)$ перейдет в состояние "частное", ячейка с координатами $(0, d)$ перейдет в состояние "остаток", а все остальные ячейки, кроме $(0, 0)$, перейдут в состояние покоя.

Справедлива следующая теорема, доказанная Б. Ф. Хайбуллиным.

Теорема 2. *Существует двумерный клеточный автомат с 9 локаторами, который решает задачу деления чисел a и b с остатком за время* $3]\log_2(a/b)[+8$.

2. Вспомогательные задачи

2.1. Удвоение числа

Задача удвоения числа a и b состоит в следующем. В начальной конфигурации только 2 ячейки находятся не в состоянии покоя, а именно ячейка с координатами $(0, 0)$ находится в состоянии "начало координат", а ячейка с координатами $(a, 0)$ находится в состоянии "аргумент". Надо, чтобы в финальной конфигурации ячейка с координатами $(2a, 0)$ перешла в состояние "результат", а все остальные ячейки, кроме $(0, 0)$, оказались в состоянии покоя.

Решить эту задачу можно следующим образом. В начальный момент "начало координат" и "аргумент" подают в эфир сигнал "такт 1". Ячейка, которая услышит этот сигнал в локаторы юг и юго-восток (а это ячейка

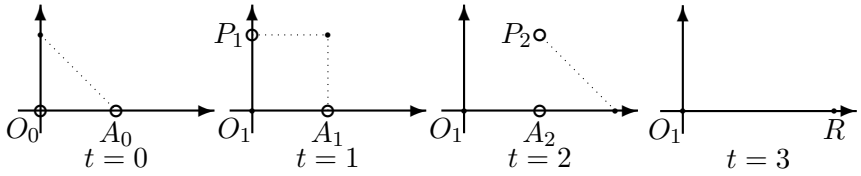


Рисунок 1. Первый алгоритм удвоения чисел

с координатами $(0, a)$) возбуждается и переходит в состояние “проекция 1”. Во второй момент “проекция 1” и “аргумент” подают в эфир сигнал “такт 2”. Ячейка, которая услышит этот сигнал в локаторы запад и юг (а это ячейка с координатами (a, a)), возбуждается и переходит в состояние “проекция 2”. В третий момент ячейки “проекция 2” и “аргумент” подают в эфир сигнал “такт 3”. Ячейка, которая услышит этот сигнал в локаторы запад и северо-запад (а это ячейка с координатами $(2a, 0)$) возбуждается и переходит в состояние “результат”. Т.е. задача решается за 3 такта.

Схематически этот алгоритм отражен на рисунке 1. Здесь ячейки, подающие сигнал в эфир, изображены полыми кружками, “начало координат” обозначается символами O с индексами, “аргумент” — символами A с индексами, “проекции” — символами P с индексами, “результат” — символом R.

Когда на каждой итерации надо удваивать число, то задачу удвоения числа удобнее сформулировать следующим образом. В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами $(0, 0)$ находится в состоянии “начало координат”, а две ячейки с координатами $(a, 0)$ и $(0, a)$ находятся в состоянии “аргумент”. Надо, чтобы в финальной конфигурации ячейки с координатами $(2a, 0)$ и $(0, 2a)$ перешли в состояние “результат”, а все остальные ячейки, кроме $(0, 0)$, оказались в состоянии покоя.

Решить эту задачу можно следующим образом. В начальный момент ячейки “аргумент”подают в эфир сигнал “такт 1”. Ячейка, которая услышит этот сигнал в локаторы запад и юг (а это ячейка с координатами (a, a)), возбуждается и переходит в состояние “проекция”. Во второй момент ячейки “проекция” и “аргумент” подают в эфир сигнал “такт 2”. Ячейка, которая услышит этот сигнал в локаторы запад и северо-запад (а это ячейка с координатами $(2, 0)$) и ячейка, которая услышит этот сигнал в локаторы юг и юго-восток (а это ячейка с координатами $(0, 2a)$) возбуждаются и переходят в состояние “результат”. Тем самым в такой постановке задача решается за 2 такта.

Схематически этот алгоритм отражен на рисунке 2.

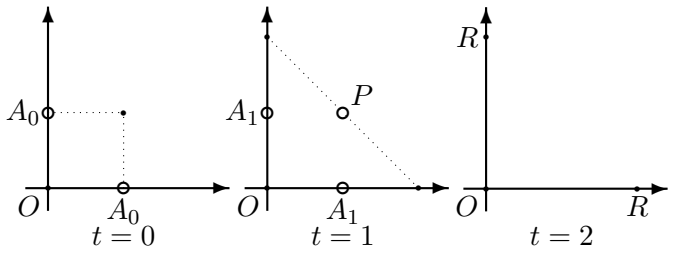


Рисунок 2. Второй алгоритм удвоения чисел

2.2. Сложение чисел

Задача сложения состоит в следующем. Дано два целых числа a и b, причем $a \geq 0$, а $b > 0$. В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами $(0, 0)$ находится в состоянии "начало координат", а две ячейки с координатами $(a, 0)$ и $(0, b)$ находятся в состоянии "слагаемое 1" и "слагаемое 2". В случае, когда $a = 0$, ячейка $(0, 0)$ будет одновременно находиться в состояниях "начало координат" и "слагаемое 1". Чтобы это сделать можно ввести еще одно состояние, или можно считать, что "начало координат" отмечается в отдельной компоненте состояния. Надо, чтобы в финальнойв состояниях конфигурации ячейка с координатами $(a + b, 0)$ перешла в состояние "результат", ячейка $(0, 0)$ осталась в состоянии "начало координат", а все остальные ячейки оказались в состоянии покоя.

Решить эту задачу можно следующим образом. В начальный момент ячейки "слагаемое 1" и "слагаемое 2" подают в эфир сигнал "такт 1". Ячейка, которая услышит этот сигнал в локаторы запад и юг (а это ячейка с координатами (a, b)), возбуждается и переходит в состояние "проекция". Также в состояние "проекция" переходит ячейка "слагаемое 2", если услышит сигнал "такт 1" в локатор "юг" (это нужно для случая, когда $a = 0$). Ячейка "слагаемое 2", которая не слышит сигнал "такт 1" в локатор "юг", переходит в состяние покоя. Ячейка "слагаемое 1", если слышит сигнал "такт 1" в локаторы "северо-запад" или "север", переходит в состояние "слагаемое 3", а иначе остается в прежнем состоянии. Во второй момент ячейки "проекция" и "слагаемое 3" подают в эфир сигнал "такт 2". Ячейка, которая услышит этот сигнал в локаторы запад и северо-запад (а это ячейка с координатами $(a + b, 0)$) возбуждается и переходит в состояние "результат". При этом "слагаемое 3" переходит в состояние покоя. Ячейка "начало координат" оба такта не меняет своего состояния.

Тем самым задача сложения чисел решается за 2 такта.

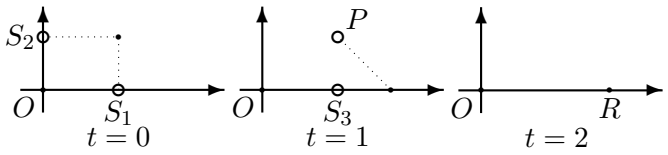


Рисунок 3. Сложение чисел

Схематически этот алгоритм для случая, когда $a > 0$, отражен на рисунке 3. На этом рисунке состояние "начало координат" обозначено символом O, "слагаемое 1" — символом S_1, "слагаемое 2" — S_2, "слагаемое 3" — S_3, "проекция" — P.

Отметим, что если в начальной конфигурации есть ячейка в состоянии "слагаемое 1", но нет ячейки в состоянии "слагаемое 2", то ячейка "слагаемое 1" не будет менять своего состояния.

2.3. Перевод числа из унарного представления в двоичное

Пусть a — натуральное число, $(a_n, a_{n-1}, \ldots, a_1)$ — двоичное представление числа a. Задача перевода числа из унарного представления в двоичное формулируется следующим образом. В начальной конфигурации в активном состоянии находится $n + 1$ ячейка: ячейка $(0, 0)$ — в состоянии "начало координат" и ячейки $(1, 0), (2, 0), \ldots, (n, 0)$ — в состоянии "один". Задача состоит в том, чтобы в такты с первого по n-ый выдавать в эфир сигнал "ноль", если $a_i = 0$, и сигнал "единица", если $a_i = 1$, $i = 1, 2, \ldots, n$. При этом в финальной конфигурации активной остается только "начало координат". Здесь первым тактом считается такт следующий после начального.

Точное решение это задачи приведено в работах [10, 11]. Здесь, как и ранее, мы опишем алгоритм решения на идейном уровне.

Алфавит вещания будет иметь вид $G = \{0, 1\} \times \{0, 1\} \times \{0, 1, 2\}$. Полугрупповой операцией по первой компоненте будет сложение по модулю 2, а по второй и третьей — максимум. Первая компонента будет использоваться для вычисления чисел a_i, $i = 1, 2, \ldots, n$. Вторая компонента — для выявления момента окончания вычислений, а третья компонента — для передачи ответа. Состояние покоя будем обозначать как состояние "ноль". Алгоритм решения задачи будет следующий.

1) В каждый такт все ячейки, которые находятся в состоянии "один" передают в эфир сигнал $(1, 1, 0)$.

2) Ячейка в состоянии “один”, которая в локатор “восток” получает сигнал $(0, *, *)$, переходит в состояние “ноль”. Здесь $*$ означает любой символ. Во всех остальных случаях ячейка не меняет состояние.

3) Если ячейка “начало координат” в локатор “восток” получает сигнал $(0, 1, 0)$, то она посылает в эфир сигнал $(0, 0, 0)$, что соответствует сигналу “ноль”.

4) Если ячейка “начало координат” в локатор “восток” получает сигнал $(1, 1, 0)$, то она посылает в эфир сигнал $(0, 0, 1)$, что соответствует сигналу “единица”.

5) Если ячейка “начало координат” в локатор “восток” получает сигнал $(0, 0, 0)$, то она посылает в эфир сигнал $(0, 0, 2)$, что соответствует окончанию передачи двоичного представления числа.

Поскольку согласно пункту 1 каждая ячейка в состоянии “один” передает в эфир по первой компоненте значение 1, то “начало координат” получит в локатор “восток” по первой компоненте сумму по модулю 2 количества ячеек в состоянии “один”, а это в первый момент равно a_1. Второй пункт гарантирует, что каждый такт число ячеей в состоянии “один” будет сокращаться вдвое, поэтому во второй такт “начало координат” получит в локатор “восток” по первой компоненте получит значение a_2 и т.д. Если “начало координат” получит в локатор “восток” по второй компоненте получит значение 0, то это означает, что ячеек в состоянии “один” больше не осталось, и можно завершать работу.

Отметим, что суммарное время работы алгоритма равно $n + 2$.

В таблице 1 приведено поведение описанного выше клеточного автомата с локаторами для случая, когда число $a = 5$. Здесь символом Q обозначается строка состояний ячеек, причем O соответствует состоянию “начало координат”, 0 — состоянию “ноль”, 1 — состоянию “один”; символом S обозначается строка посылаемых в эфир сигналов; символом E — строка значений локатора “восток”.

Координаты ячеек		$(0,0)$	$(1,0)$	$(2,0)$	$(3,0)$	$(4,0)$	$(5,0)$
$t=0$	Q	O	1	1	1	1	1
	S	$(0,0,2)$	$(1,1,0)$	$(1,1,0)$	$(1,1,0)$	$(1,1,0)$	$(1,1,0)$
	E	$(1,1,0)$	$(0,1,0)$	$(1,1,0)$	$(0,1,0)$	$(1,1,0)$	$(0,0,0)$
$t=1$	Q	O	0	1	0	1	0
	S	$(0,0,1)$	$(0,0,0)$	$(1,1,0)$	$(0,0,0)$	$(1,1,0)$	$(0,0,0)$
	E	$(0,1,0)$	$(0,1,0)$	$(1,1,0)$	$(1,1,0)$	$(0,0,0)$	$(0,0,0)$
$t=2$	Q	O	0	1	0	0	0
	S	$(0,0,0)$	$(0,0,0)$	$(1,1,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
	E	$(1,1,0)$	$(1,1,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
$t=3$	Q	O	0	0	0	0	0
	S	$(0,0,1)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
	E	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
$t=4$	Q	O	0	0	0	0	0
	S	$(0,0,2)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
	E	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$

Таблица 1. Перевод числа из унарного представления в двоичное

В такты 1, 2, 3 в третьей компоненте сигнала вещания мы можем наблюдать двоичное представление числа 5 — $(1,0,1)$.

Легко видеть, что приведенный клеточный автомат будет работать и в случае, когда ячейки в состоянии "один" будут стоять не подряд, а в любых положительных позициях оси абсцисс, и тогда автомат выдаст в эфир двоичное представление количества ячеек в состоянии "один", находящихся правее "начала координат".

Можем также заметить, что легко модифицировать этот автомат, чтобы он выдавал компоненты двоичного представления не каждый такт, а, например, через такт.

И наконец заметим, что аналогичным образом мы можем подсчитать число ячеек в состоянии "один", находящихся на любом из лучей из множества L.

2.4. Перевод числа из двоичного представления в унарное

Задача перевода числа из двоичного представления в унарное обратна к предыдущей задаче.

Сформулирована она может быть следующим образом. Пусть a — натуральное число, $(a_n, a_{n-1}, \ldots, a_1)$ — двоичное представление числа a. В начальной конфигурации в активном состоянии находится только ячейка $(0,0)$ в состоянии "начало координат 0". В такты с первого по n-ый ячейка $(0,0)$ выдает в эфир сигнал "ноль", если $a_i = 0$, и сигнал "единица", если $a_i = 1$, $i = 1, 2, \ldots, n$. В финальной конфигурации активными

должны остаться только ячейки $(0,0)$ (“начало координат 2”) и $(a,0)$ (“результат”).

Точное решение это задачи можно найти в работах [9, 10]. Приведем идею этого решения.

Алфавит вещания будет иметь вид $G = \{0,1\} \times \{0,1,2,3,4\}$. Полугрупповой операцией по первой компоненте будет сложение по модулю 2, а по второй — максимум. Первая компонента будет использоваться для вычисления числа a, вторая компонента — для передачи команд. Состояние покоя будем обозначать как состояние “ноль”. Алгоритм решения задачи будет следующий.

1) В начальный (нулевой) такт ячейка $(0,0)$ (“начало координат 0”) подает в эфир сигнал $(1,2)$, который можно интерпретировать как “начинаем”, и переходит в состояние “начало координат 1”. По команде “начинаем” все ячейки положительной полуоси оси абцисс (т.е. те, кто услышат этот сигнал в локатор “запад”) переходят в состояние “один”.

2) В следующие n тактов ячейка $(0,0)$ подает в эфир по второй компоненте сигнал a_i, $i = 1,2,\ldots,n$ ($a_i = 0$ интерпретируется как сигнал “ноль”, а $a_i = 1$ — как “единица”). При этом на следующий такт после того, как в первый раз a_i окажется равным 1, ячейка $(0,0)$ перейдет в состояние “начало координат 2”, а до этого будет оставаться в состоянии “начало координат 1”. Поскольку число $a > 0$, к финальному состоянию ячейка $(0,0)$ обязательно окажется в состоянии “начало координат 2”. При этом “начало координат 1” по первой компоненте передает в эфир сигнал 1, а “начало координат 2” — сигнал 0.

3) Ячейки в состоянии “один”, если слышат в локатор “запад” по второй компоненте 0 или 1, то передают в эфир сигнал $(1,0)$, а ячейки в состоянии “ноль” — сигнал $(0,0)$.

4) Ячейка в состоянии “один” переходит в состояние “ноль”, если она получает в локатор “запад” сигнал, значение второй компоненты которого равно либо 0, либо 1, и оно не совпадает со значением первой компоненты сигнала.

5) В $(n+1)$-ый такт ячейка $(0,0)$ передает в эфир сигнал $(0,3)$, что означает окончание двоичной записи. В результате все ячейки в состоянии “один” переходят в состояние “два”.

6) В $(n+2)$-ый такт ячейка $(0,0)$ передает в эфир сигнал $(0,3)$, а все ячейки в состоянии “два” подают в эфир сигнал $(0,4)$. При этом

все ячейки в состоянии “два”, которые в локатор “запад” услышат сигнал $(0,4)$ перейдут на следующий такт в состояние “ноль”, а самая левая ячейка в состоянии “два” услышит в локатор “запад” сигнал $(0,3)$ и перейдет в состояние “результат”.

В таблице 2 приведено поведение описанного выше клеточного автомата с локаторами для случая, когда число $a = 5$. Здесь символом Q обозначается строка состояний ячеек, причем O_0, O_1, O_2 соответствует состоянию “начало координат 0”, “начало координат 1” и “начало координат 2”, 0 — состоянию “ноль”, 1 — состоянию “один”, 2 — состоянию “два”, R — состоянию “результат”; символом S обозначается строка посылаемых в эфир сигналов; символом W —строка значений локатора “запад”.

t		$(0,0)$	$(1,0)$	$(2,0)$	$(3,0)$	$(4,0)$	$(5,0)$	$(6,0)$	$(7,0)$	$(8,0)$	$(9,0)$	$(10,0)$	$(11,0)$	$(12,0)$	$(13,0)$
0	Q	O_0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S	$(0,2)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
	W	$(0,0)$	$(0,2)$	$(0,2)$	$(0,2)$	$(0,2)$	$(0,2)$	$(0,2)$	$(0,2)$	$(0,2)$	$(0,2)$	$(0,2)$	$(0,2)$	$(0,2)$	$(0,2)$
1	Q	O_1	1	1	1	1	1	1	1	1	1	1	1	1	1
	S	$(1,1)$	$(1,0)$	$(1,0)$	$(1,0)$	$(1,0)$	$(1,0)$	$(1,0)$	$(1,0)$	$(1,0)$	$(1,0)$	$(1,0)$	$(1,0)$	$(1,0)$	$(1,0)$
	W	$(0,0)$	$(1,1)$	$(0,1)$	$(1,1)$	$(0,1)$	$(1,1)$	$(0,1)$	$(1,1)$	$(0,1)$	$(1,1)$	$(0,1)$	$(1,1)$	$(0,1)$	$(1,1)$
2	Q	O_2	1	0	1	0	1	0	1	0	1	0	1	0	1
	S	$(0,0)$	$(1,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(1,0)$
	W	$(0,0)$	$(0,0)$	$(1,0)$	$(1,0)$	$(0,0)$	$(0,0)$	$(1,0)$	$(1,0)$	$(0,0)$	$(0,0)$	$(1,0)$	$(1,0)$	$(0,0)$	$(0,0)$
3	Q	O_2	1	0	0	0	1	0	0	0	1	0	0	0	1
	S	$(0,1)$	$(1,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(1,0)$
	W	$(0,0)$	$(0,1)$	$(1,1)$	$(1,1)$	$(1,1)$	$(1,1)$	$(0,1)$	$(0,1)$	$(0,1)$	$(0,1)$	$(1,1)$	$(1,1)$	$(1,1)$	$(1,1)$
4	Q	O_2	0	0	0	0	1	0	0	0	0	0	0	0	1
	S	$(0,3)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(1,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(1,0)$
	W	$(0,0)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(1,3)$	$(1,3)$	$(1,3)$	$(1,3)$	$(1,3)$	$(1,3)$	$(1,3)$	$(1,3)$
5	Q	O_2	0	0	0	0	2	0	0	0	0	0	0	0	2
	S	$(0,3)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,4)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,4)$
	W	$(0,0)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,4)$	$(0,4)$	$(0,4)$	$(0,4)$	$(0,4)$	$(0,4)$	$(0,4)$	$(0,4)$
6	Q	O_2	0	0	0	0	R	0	0	0	0	0	0	0	0
	S	$(0,3)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
	W	$(0,0)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$	$(0,3)$

Таблица 2. Перевод числа из двоичного представления в унарное

Будем условно считать, что состояние “начало координат 1” соответствует состоянию “один” для ячейки $(0,0)$, а состояние “начало координат 2” соответствует состоянию “ноль” для ячейки $(0,0)$. Тогда докажем по индукции, что если ячейка $(0,0)$ с первого по n-й такт по второй компоненте будет посылать в эфир последовательность $a_1, a_2, \ldots, a_n$, то к $(n+1)$-му такту в состоянии “один” будут ячейки с координатами $(\sum_{i=1}^{n} 2^i a_i + 2^n k, 0)$, $k = 0, 1, 2, \ldots$.

Базис индукции. $n = 1$. На первом такте все ячейки неотрицательной полуоси оси абцисс, включая ячейку $(0,0)$, посылают в эфир сигнал 1 по первой компоненте. Поэтому все ячейки с координатами $(2k, 0)$, $k = 0, 1, 2, \ldots$ (четные ячейки) получат в локатор “запад” по первой компоненте значение 0, а все ячейки с координатами $(2k+1, 0)$,

$k = 0, 1, 2, \ldots$ (нечетные ячейки) — значение 1. Следовательно, если $a_1 = 0$, то ко второму такту в состоянии "один" останутся четные ячейки, а если $a_1 = 1$, то — нечетные. Базис индукции доказан.

Индуктивный переход. Пусть ячейка $(0, 0)$ с первого по $(n-1)$-й такт по второй компоненте посылала в эфир последовательность $a_1, a_2, \ldots, a_{n-1}$, к n-му такту в состоянии "один" остались ячейки с координатами $(\sum_{i=1}^{n-1} 2^i a_i + 2^{n-1}k, 0)$, $k = 0, 1, 2, \ldots$. Обозначим $b = \sum_{i=1}^{n-1} 2^i a_i$. Тогда все ячейки с координатами $(k, 0$, $k = 0, 1, \ldots, b$, получат на локатор "запад" по первой компоненте значение 0. Все ячейки с координатами $(k, 0$, $k = b+1, b+2, \ldots, b+2^{n-1}$, получат на локатор "запад" по первой компоненте значение 1. Опять все ячейки с координатами $(k, 0$, $k = b + 2^{n-1} + 1, b + 2^{n-1} + 2, \ldots, b + 2^n$, получат на локатор "запад" по первой компоненте значение 0 и т.д.

Пусть на n-м такте ячейка $(0, 0)$ посылает в эфир по второй компоненте значение a_n. Тогда если $a_n = 0$, то на $(n+1)$-м такте в состоянии "один" останутся ячейки с координатами $(b + 2^n k, 0)$, $k = 0, 1, 2, \ldots$. Если $a_n = 1$, то на $(n+1)$-м такте в состоянии "один" останутся ячейки с координатами $(b + 2^{n-1} + 2^n k, 0)$, $k = 0, 1, 2, \ldots$.

Индуктивный переход доказан.

Таким образом, если $= \sum_{i=1}^{n} 2^i a_i$, то к $(n+1)$-му такту самая левая ячейка в состоянии "один" будет иметь координаты $(a, 0)$, а значит на $(n+3)$-м такте в состояние "результат" перейдет ячейка $(a, 0)$, что мы и хотели получить.

Отметим, что суммарное время работы алгоритма равно $n + 3$.

Отметим, что аналогичным образом можно отложить значение a на ось ординат, т.е. чтобы в финальной конфигурации в состоянии "результат" оказалась ячейка $(0, a)$.

2.5. Задача подсчета числа единиц

Пусть в начальной конфигурации ячейка $(0, 0)$ находится в состоянии "начало координат" и на положительной полуоси оси абцисс разбросано некоторое конечное число ячеек в состоянии "один", а остальные ячейки находятся в состоянии покоя (состоянии "ноль"). Хочется подсчитать сколько ячеек находится в состоянии "один", и если их число равно a, то хочется в финальной конфигурации, чтобы ячейка $(0, a)$ перешла в состояние "результат".

Пусть двоичное представление числа a имеет вид $(a_n, a_{n-1}, \ldots, a_1)$. Мы можем запустить алгоритм из раздела 2.3 и тогда, начиная с первого такта, ячейка $(0, 0)$ будет посылать в эфир последовательность $a_1, a_2, \ldots, a_n$. Теперь, параллельно используя алгоритм из раздела 2.4, мы можем отложить число a на оси ординат.

Отметим, что суммарное время работы алгоритма равно $n + 3$, поскольку оба алгоритма работают такое время, работают параллельно и синхронно.

Решение этой задачи мы в дальнейшем используем при алгоритме деления.

3. Умножение чисел

Напомним задачу умножения чисел a и b для клеточного автомата с локаторами. В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами $(0, 0)$ находится в состоянии, которое можно назвать "начало координат", ячейка с координатами $(a, 0)$ находится в состоянии, которое можно назвать "первый сомножитель", а ячейка с координатами $(0, b)$ находится в состоянии, которое можно назвать "второй сомножитель". Клеточный автомат решает задачу умножения чисел, если в финальной конфигурации ячейка с координатами $(a \cdot b, 0)$ перейдет в состояние "результат умножения", а все остальные ячейки, кроме $(0, 0)$, перейдут в состояние покоя.

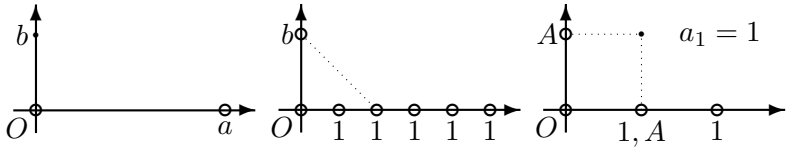
Крупными мазками опишем алгоритма решения задачи умножения чисел a и b. Пусть $(a_n, a_{n-1}, \ldots, a_1)$ — двоичное представление числа a, т.е. $a = \sum_{i=1}^{n} 2^{i-1} a_i$. Тогда $ab = \sum_{i=1}^{n} 2^{i-1} b a_i$. Клеточный автомат, который будет решать задачу умножения чисел a и b, будет одновременно выполнять 3 задачи. Чтобы выполнять эти три задачи параллельно в нашемавтомате будут работать одновременно 3 разных автомата, каждый надсвоим множеством компонент состояний и своим множеством компонентсигналов вещания.

С помощью первого автомата будет решаться задача вычисления чисел $2^{i-1}b$, $i = 1, 2, \ldots, n$. Делать это будем с помощью второго алгоритма удвоения чисел, описанного в разделе 2.1, т.е. сначала сложим два числа b за 2 такта, затем сложим два числа $2b$ за 2 такта и т.д., т.е. каждые 2 такта мы сможем получать числа $2^{i-1}b$, $i = 1, 2, \ldots, n$.

С помощью второго автомата будем находить двоичное представление числа a, т.е. будем получать числа $a_1, a_2, \ldots, a_n$. Делать это будем как описано в разделе 2.3. Но будем притормаживать получение чисел $a_2, a_3, \ldots, a_n$ так, чтобы они появлялись одновременно с числами $2^1 b, 2^2 b, , \ldots, 2^{n-1} b$.

С помощью третьего автомата будем накапливать суммы

$$S_i = \sum_{j=1}^{i} 2^{j-1} b a_j, i = 1, 2, \ldots, n, S_0 = 0.$$



Риснок 4.Умножение чисел,$t = 0$, $t = 1$, $t = 2$.

Т.е. если вычисленное число a_i равно единице, то к числу S_{i-1} добавим вычисленное число $2^{i-1}b$. Это тоже можно сделать за 2 такта, как было описано в разделе 2.2.

Тем самым описанный клеточный автомат с локаторами может приблизительно за $2n$ тактов вычислить число ab.

На самом деле алгоритм несколько сложнее и нам понадобится еще четвертый автомат, который будет управляющим и будет координировать действия описанных выше трех автоматов. Этот автомат будет иметь свое множество компонент состояний и свое множество компонент сигналов вещания.

Чтобы запускать удвоение чисел первым автоматом и суммирование чисел третьим автоматом итеративно каждый второй такт, отождествим состояния “аргумент” и “результат” в первом автомате, и состояния “слагаемое 1” и результат в третьем автомате.

Опишем более детально наш алгоритм и получим точную оценку времени работы этого алгоритма.

1) В начальный (нулевой) такт ячейка $(0, 0)$ (“начало координат”) и ячейка $(a, 0)$ (“первый сомножитель”) посылают в эфир сигнал, который можно назвать “строим унарное представление”. Этот сигнал посылается по компоненте четвертого автомата. Все ячейки, которые услышат этот сигнал в локаторы “восток” и “запад” поймут, что они между “началом координат” и “первым сомножителем” и перейдут в состояние “один” второго автомата, “первый сомножитель” тоже переходит в состояние “один” второго автомата.

2) В результате на первом такте ячейки с координатами $(1, 0), (2, 0), \ldots,$ $(a, 0)$ окажутся в состоянии “один”, и мы на них запустим второй автомат для вычисления чисел $a_1, a_2, \ldots, a_n$. Ячейка $(0, b)$ (“второй сомножитель” проецируется на ось абцисс. Для этого “начало координат” и “второй сомножитель” посылают в эфир по компоненте четвертого автомата сигнал, который можно назвать “проецируем на ось абцисс”. Ячейка, которая услышит этот сигнал в локаторы “запад” и “северо-запад” (а это будет ячейка $(b, 0)$), перейдет в

состояние “аргумент” первого автомата, “второй сомножитель” тоже перейдет в состояние “аргумент” первого автомата.

3) На втором такте в эфире от второго автомата появится число a_1. Если $a_1 = 1$, то ячейка $(b, 0)$ переводится в состояние “слагаемое 1” третьего автомата, а если $a_1 = 0$, то ячейка $(0, 0)$ переводится в состояние “слагаемое 1” третьего автомата. Также на втором такте ячейки $(b, 0)$ и $(0, b)$ окажутся в состоянии ‘аргумент” первого автомата, поэтому запускается первый автомат для получения чисел $2^{i-1}b$, $i = 2, 3, \ldots, n$.

4) На третьем такте в эфире от второго автомата появится число a_2. Если $a_2 = 1$, то первый автомат, у которого на следующий такт появится ячейка на оси ординат в состоянии “результат” (это будет ячейка $(0, 2b)$, переводит добавочно эту ячейку в состояние “слагаемое 2” третьего автомата. Если $a_2 = 0$, то у третьего автомата ячеки в состоянии “слагаемое 2” не появится. Также с этого момента второй автомат переключается в режим выдачи ответа через такт.

5) На тактах $2i$, $i = 2, 3, \ldots$, будут появлятся результаты удвоения, которые одновременно могут превращаться в “слагаемое 2” третьего автомата (если на предыдущем такте $a_i = 1$), и третий автомат начнет складывать числа. Также на этих тактах может появится результат сложения третьего автомата, если за 2 такта до этого процесс сложения был запущен. Результат сложения сразу же превратится в “слагаемое 1”, поскольку мы отождествили состояния “слагаемое 1” и “результат” у третьего автомата.

6) На тактах $2i + 1$, $i = 2, 3 \ldots, n - 1$, будут появлятся числа a_{i+1} от второго автомата. Если $a_{i+1} = 1$, то первый автомат, у которого на следующий такт появится ячейка на оси ординат в состоянии “результат” (а это будет ячейка $(0, 2^i b)$) переводит добавочно эту ячейку в состояние “слагаемое 2” третьего автомата, что приводит к запуску процесса сложения третьим автоматом. Если $a_{i+1} = 0$, у третьего автомата не появляется ячейка в состоянии “слагаемое 2” и процесс сложения чисел не запускается.

7) На такте $2n + 1$ второй автомат поймет, что обработка числа a закончилась и прокричит в эфир сигнал “окончание работы”. В результате все ячейки, кроме “начала координат” и результата сложения третьего автомата, перейдут в состояние покоя, а результат сложения третьего автомата перейдет в состояние “результат умножения”.

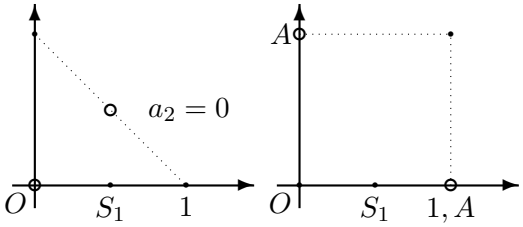


Рисунок 5. Умножение чисел, $t = 3$, $t = 4$.

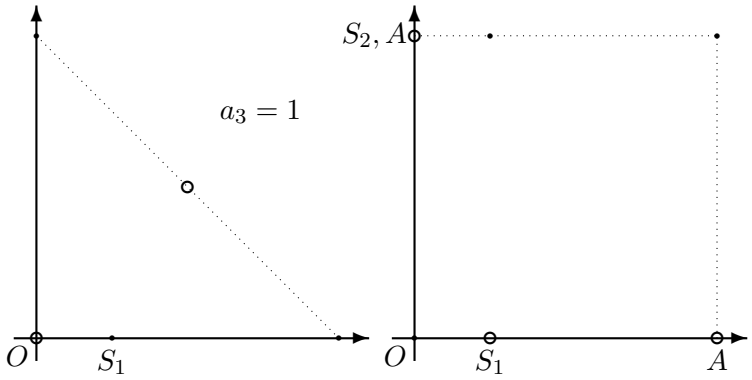


Рисунок 6. Умножение чисел, $t = 5$, $t = 6$.

8) На такте $2n + 2$ автомат завершает работу в требуемой финальной конфигурации.

Таким образом описанный клеточный автомат с локаторами решает задачу умножения чисел a и b за время $2]\log_2 a[+2$. Теорема 1 доказана.

На рисунках 4 – 7 изображен процесс умножения чисел $a = 5$ и $b = 2$. На этих рисунках полыми кружками изображены ячейки, подающие сигнал в эфир. Символом O обозначено "начало координат". Символом A обозначено состояние "аргумент" первого автомата. Символом 1 обозначено состояние "один" второго автомата. Символами S_1 и S_2 обозначены состояния "слагаемое 1" и "слагаемое 2" третьего автомата. Символом R обозначено состояние "результат умножения". Когда ячейка одновременно находится в нескольких описанных выше состояниях разных автоматов, то они перечислены через запятую.

4. Деление чисел

Идея алгоритма решения состоит в последовательном проецировании отрезков делителя на абсциссу, на которой также отложено делимое.

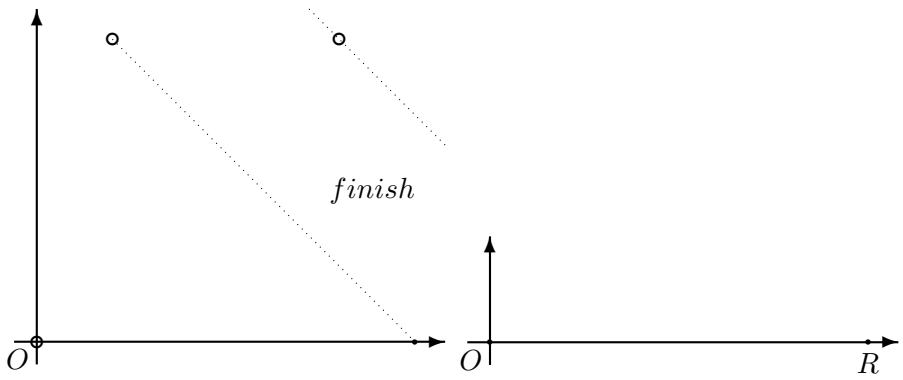


Рисунок 7. Умножение чисел, $t = 7$, $t = 8$.

Как только сумма отрезков делителя становится больше или равна делимому, значит частное и остаток от деления найдены. Далее следует откладывание частного и остатка от деления на абсциссе и ординате соответственно, в унарном формате. Таким образом операцию деления можно разбить на два этапа:

- откладывание отрезков делителя на оси абсцисс;
- откладывание частного и остатка от деления на осях абсцисс и ординат.

4.1. Первый этап

Рассмотрим следующий двумерный клеточный автомат с локаторами $\sigma = (\mathbb{Z}^2, Q, \emptyset, E_8, \max, L, \varphi, \psi)$. Q — это множество состояний ячеек автомата, $Q = \{*, O, O_1, O_2, O_3, O_4, O_5, A, A_1, A_2, A_3, A_4, A_5, A_6, B, B_1, B_2, B_3, B_4, B_5, X, Y_2, Y_3, R\}$, где * — состояние покоя. Шаблон соседства пустой. Шаблон локаторов L задается соотношением (1). $E_8 = \{0, 1, \ldots, 7\}$ — алфавит сигналов вещания, полугрупповая операция — максимум. Ячейки передают сигналы о своем местоположении и состоянии.

Пусть b — делимое, а a — делитель, и $b/a \leq 2^n$. Тогда клеточный автомат с локаторами σ будет решать задачу первого этапа, т.е. откладывания отрезков делителя на оси абцисс за время $T_1 = 2n + 5$.

В качестве примера рассмотрим случай, когда делимое делится на делитель не полностью. Делитель $a = 3$, делимое $b = 17$. Начало координат обозначено буквой O. На рисунках ячейки, которые посылают сигналы в эфир, обозначены белыми кругами, а рядом указано значение сигнала вещания, посылаемого в эфир. Ячейки без кругов посылают в

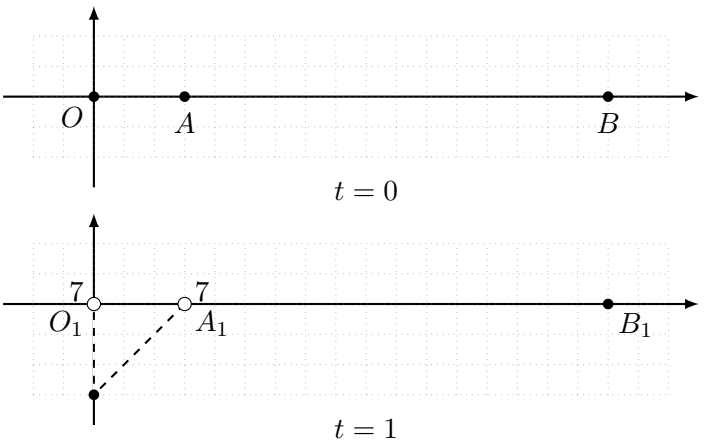


Рисунок 8. Деление чисел, такты $0, 1$

эфир нейтральный сигнал 0. Состояние покоя на рисунках изображено в виде пустой клетки.

Первые два такта клеточный автомат готовится откладывать отрезки делителя на абсциссу.

На нулевом такте ($t = 0$) ячейка $(0, 0)$ в состоянии O (начало координат) и ячейка $(a, 0)$ в состоянии A (делитель) переходят соответственно в состояния O_1 и A_1, принимают решение о посылке в эфир сигнала 7. Ячейка $(b, 0)$ в состоянии B (делимое) переходит в состояние B_1.

В первом такте ($t = 1$) ячейки в состояниях O_1 и A_1 посылают в эфир сигналы 7, переходят соответственно в состояния O_2 и A_2. Ячейка в состоянии покоя, которая получает сигналы 7 на локаторы “север” и “северо-восток”, переходит в состояние Y_2. Вспомогательная ячейка в состоянии Y необходима для создания временных ячеек в состоянии X, используемых для проецирования отрезков делителя на ось абсцисс. Ячейка в состоянии B_1 переходит в состояние B_2.

Далее следует цикл из двух тактов, который работает пока не закончится процедура откладывания отрезков делителя на ось абсцисс. Во время работы цикла ячейки в состоянии A_3 не переходят в другое состояние.

На первом такте цикла (в нашем примере - такты 2,4,6,8) ячейка в состоянии O_2 посылает в эфир сигнал 2, и если она получает сигнал 5 на локатор “восток”, то переходит в состояние O_4, иначе – в состояние O_3. Ячейка в состоянии A_2 посылает в эфир сигнал 2, и если она получает сигнал 1 на локатор “восток”, то переходит в состояние A_5, иначе – в состояние A_3. Ячейка в состоянии B_2, которая получает сигнал 1 на

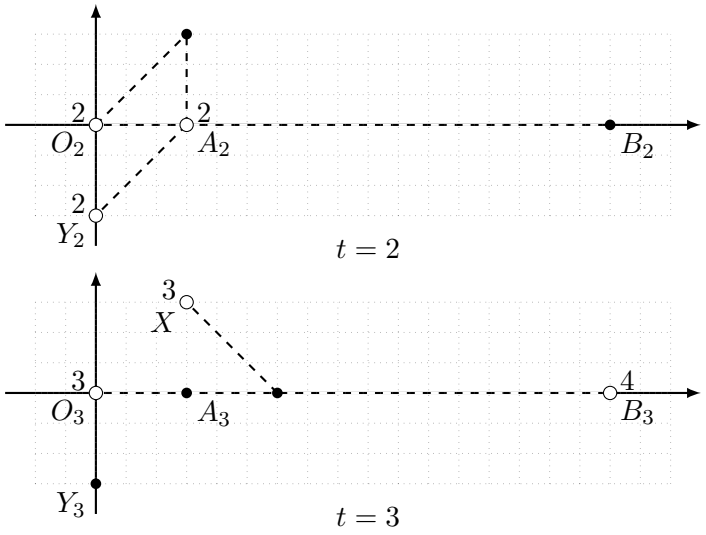


Рисунок 9. Деление чисел, такты 2, 3

локатор “восток”, переходит в состояние B_5, иначе – в состояние B_3. Ячейка в состоянии Y_2 посылает в эфир сигнал 2 и переходит в состояние Y_3. Ячейка в состоянии покоя, которая получает сигнал 2 в локаторы “юго-запад” и “юг”, переходит в состояние X.

Если в первом такте присутствует ячейка в состоянии A_4, то она посылает в эфир сигнал 1 и автоматически переходит в состояние покоя. Если присутствует ячейка в состоянии B_4, то она посылает в эфир сигнал 5 и переходит в состояние A_3.

На втором такте цикла (в нашем примере - такты 3,5,7,9) ячейка в состоянии O_3 посылает в эфир сигнал 3. Ячейка переходит в состояние O_4, если получает сигнал 5 на локатор “восток”, иначе возвращается в состояние O_2. Ячейки в состоянии X посылают в эфир сигнал 3 и переходят в состояние покоя. Ячейка в состоянии B_3 посылает в эфир сигнал 4. При этом если ячейка получает сигнал 3 на локаторы “северо-запад” и “запад”, то переходит в состояние B_4, иначе возвращается в состояние B_2. Ячейка в состоянии Y_3 переходит снова в состояние Y_2. Ячейки в состоянии покоя, которые получают сигналы 3 на локаторы “северо-запад” и “запад”, переходят в состояние A_2. Ячейки в состоянии покоя, которые получают сигнал 3 на локатор “северо-запад” и сигнал 4 на локатор “запад”, переходят в состояние A_4.

Если на втором такте присутствует ячейка в состоянии O_4, то она посылает в эфир сигнал 6 и переходит в состояние O_5. Ячейка в состоянии A_5 посылает в эфир сигнал 5 и переходит в состояние A_3. Ячейка в состоянии B_5 также посылает в эфир сигнал 5 и переходит в состояние

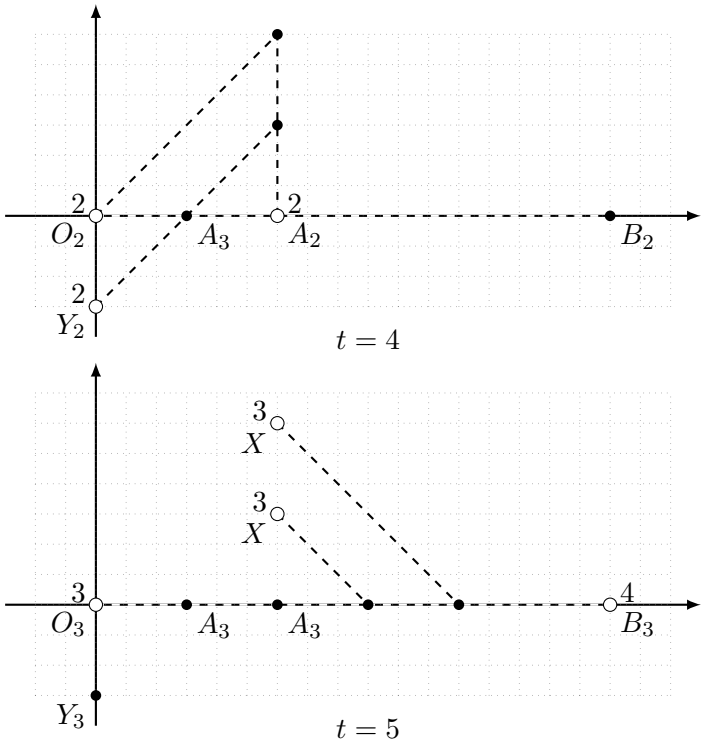


Рисунок 10. Деление чисел, такты $4, 5$

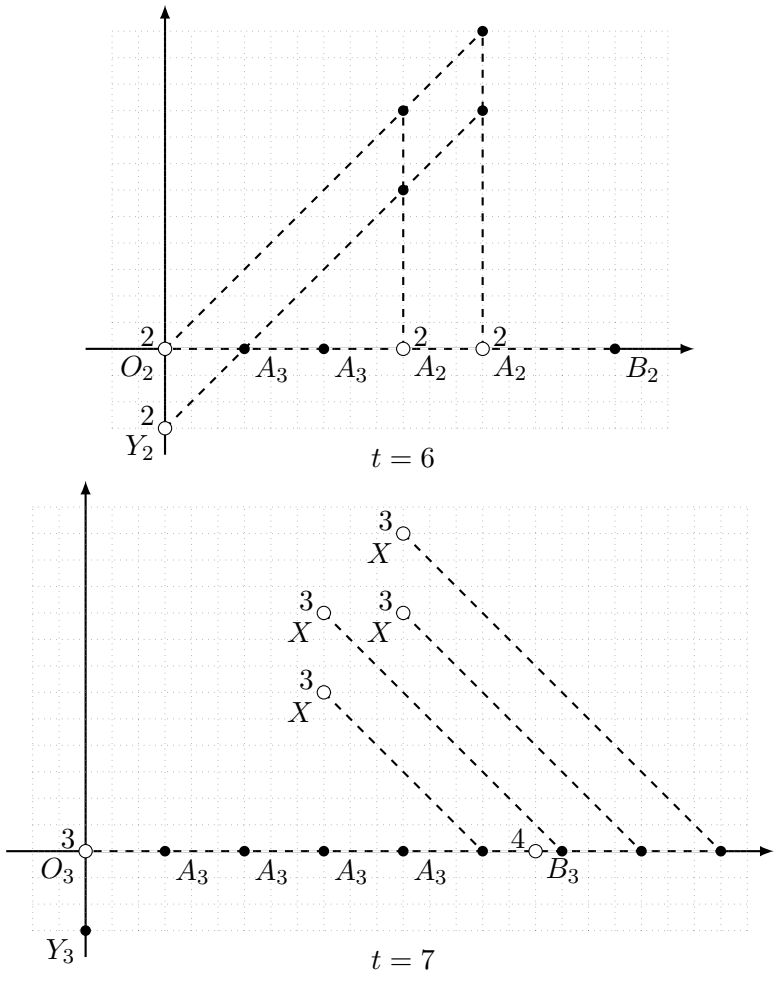


Рисунок 11. Деление чисел, такты $6, 7$

покоя. Ячейка в состоянии покоя, которая получает сигнал 5 на локаторы "юг" и "юго-восток", переходит в состояние R.

Появление ячейки в состоянии B_4 или B_5 говорит о завершении откладывания отрезков делителя на ось абсцисс. Далее следуют два такта, в которых определяется остаток от деления и удаляется вспомогательная ячейка в состоянии Y_2 или Y_3.

Ячейка в состоянии O_4 посылает в эфир сигнал 6 и переходит в состояние O_5. Ячейки в состоянии A_3, которые получают сигнал 6 на локатор "запад", переходят в состояние A_6. Ячейка в состоянии Y_2 или Y_3, которая получает сигнал 6 на локатор "север", переходит в состояние покоя.

В нашем примере первый этап завершается на такте 11. В результате определено количество ячеек в состоянии A_6 (частное), и расстояние между самой правой ячейкой в состоянии A_6 и ячейкой в состоянии R (остаток).

Можно заметить, что за каждую двухтактовую итерацию количество ячеек в состоянии A с индексами удваивается. Это происходит из-за того, что ячейки в состоянии X получаются на пересечении вертикальных прямых, проходящих через ячейки в состоянии A_2 и двух наклонных прямых, проходящих через ячейки в состоянии O_2 и Y_2. Поскольку на первой итерации появляется 1 точка в состоянии A с индексами, а последней итерации таких точек будет 2^n, то всего итераций будет $n+1$. Добавляя к этому 2 такта в начале работы алгоритма и 1 такт в конце (для формирования остатка), получим, что время работы первого этапа будет равно $T_1 = 2(n+1)+3 = 2n+5$.

4.2. Второй этап

Второй этап — это представление результатов первого этапа в нужном формате.

После первого этапа частное представлено в виде ячеек в состоянии A_6. Если к описанному выше автомату добавить автомат для подсчета числа единиц, описанный в разделе 2.5, и если отождествить состояние A_6 с состоянием "один" автомата для подсчета числа единиц, то в момент появления ячеек в состоянии A_6 автоматически запустится автомат для подсчета числа единиц. В результате работы этого автомата частное будет отображено на ось абсцисс в унарном формате за время $T_2 = n+3$.

Для отображения остатка на ось ординат достаточно одного такта. Для этого ячейка R и начало координат, должны послать в эфир некий сигнал, а ячейка, которая услышит этот сигнал в локаторы "восток" и "юг" и будет ячейкой, представляющей остаток. Причем это можно сделать в любое время на фоне вычисления частного.

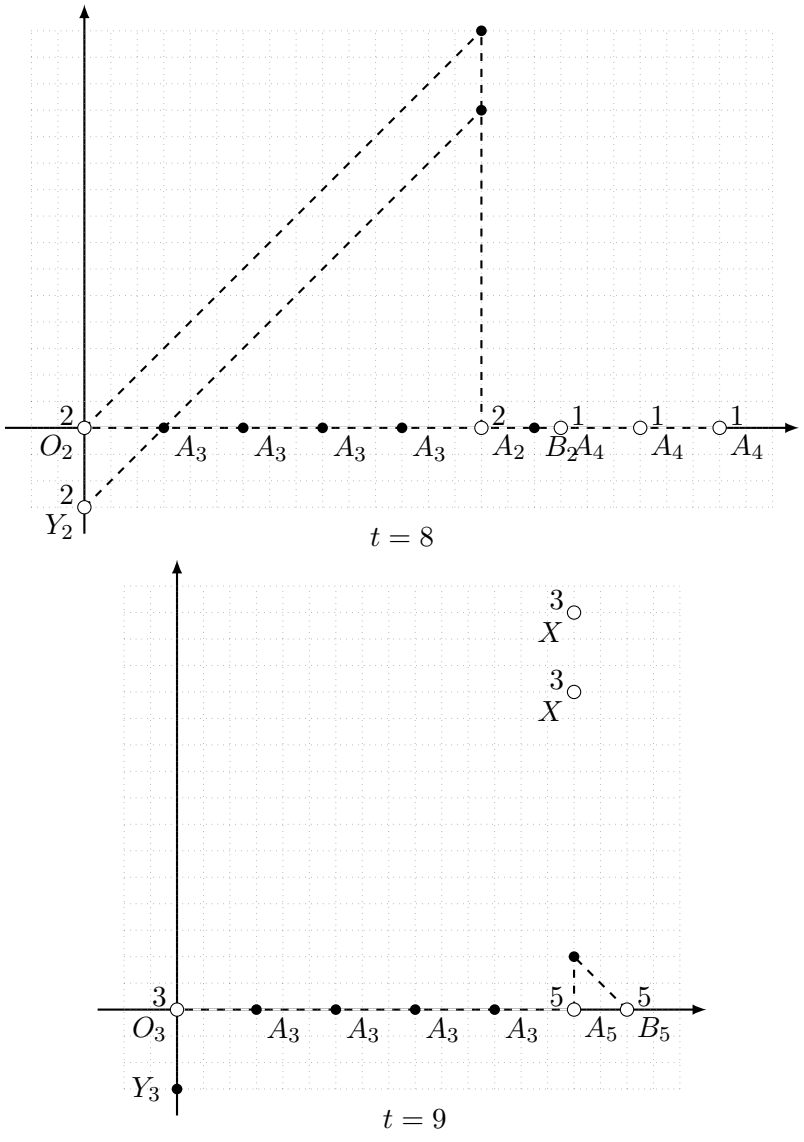


Рисунок 12. Деление чисел, такты $8, 9$

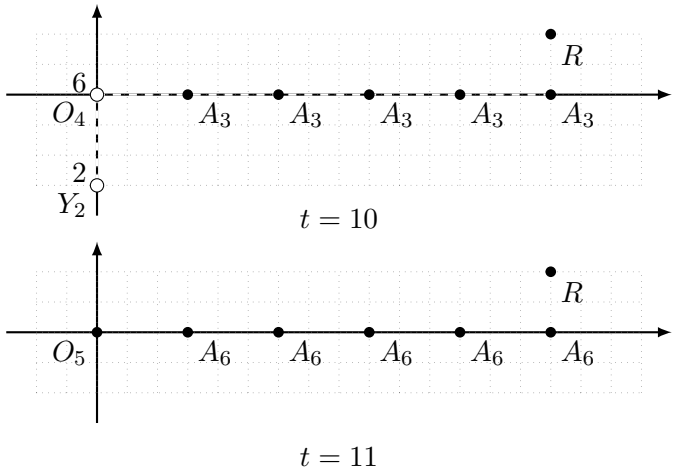


Рисунок 13. Деление чисел, такты $10, 11$

Таким образом задача деления чисел b и a, где $b/a \leq 2^n$, решается нашим двумерным клеточным автоматом с локаторами за время $T_1 + T_2 = 3n + 8$. Теорема 2 доказана.

Список литературы

[1] Карацуба А., Офман Ю., "Умножение многозначных чисел на автоматах", *Доклады АН СССР*, **145**:2 (1962), 293–294.

[2] Schönhage A., Strassen V., "Schnelle Multiplikation großer Zahlen", *Computing*, 1971, № 7, 281–292.

[3] Fürer M., "Faster integer multiplication", *STOC 2007 Proceedings*, 2007, 57–66.

[4] David Harvey, Joris van der Hoeven, "Integer multiplication in time $O(n \log n)$", *Annals of Mathematics*, **193**:2 (2021), 563–617.

[5] Christoph Burnikel C., Ziegler J., "Fast Recursive Division", *Max-Planck-Institut für Informatik*, 1998.

[6] Гасанов Э. Э., "Линейный по порядку алгоритм умножения чисел с помощью двумерного клеточного автомата с локаторами", Международная научная конференция "Математика в созвездии наук" к юбилею академика В.А. Садовничего, Москва, Россия, 1-2 апреля 2024, стр. 316-318.

[7] Гасанов Э. Э., “Клеточные автоматы с локаторами”, *Интеллектуальные системы. Теория и приложения*, **24**:2 (2020), 120–133.

[8] Калачев Г. В., “Замечания к определению клеточного автомата с локаторами”, *Интеллектуальные системы. Теория и приложения*, **24**:4 (2020), 47–56.

[9] Ибрагимова Д. Э., “Сложение векторов на прямой с помощью клеточного автомата с локаторами”, *Интеллектуальные системы. Теория и приложения*, **26**:4 (2022), 134–162.

[10] Гасанов Э. Э., “Клеточные автоматы с локаторами как модель устройств с беспроводной связью”, *Математические вопросы кибернетики*, **21** (2023), 5–51.

[11] Васильев Д. И., “Поиск ближайшего соседа на прямой с помощью клеточного автомата с локаторами”, *Интеллектуальные системы. Теория и приложения*, **24**:3 (2020), 99–119.

Fast algorithms for multiplication and division of natural numbers using cellular automata with locators
Gasanov E.E., Khaybullin B.F.

For multiplication and division of n-digit natural numbers, algorithms with complexity of order $n^{\log_2 3}$ and even order $n^{\log n}$ are known. In this paper, an algorithm for multiplying n-digit natural numbers in $2n + 2$ cycles is proposed. Here, the digit of number a is understood as the number $]\log_2 a[$. For division of natural numbers with remainder, an algorithm with a running time of $3n + 8$ cycles is proposed, where n is the digit of the quotient. The proposed algorithms use two-dimensional cellular automata with locators as calculators.

Keywords: multiplication of natural numbers, division of natural numbers, cellular automata with locators.

References

[1] Karatsuba A., Ofman Yu., “Multiplication of multi-digit numbers on automata”, *Reports of the USSR Academy of Sciences*, **145**:2 (1962), 293–294 (In Russian).

[2] Schönhage A., Strassen V., “Schnelle Multiplikation großer Zahlen”, *Computing*, 1971, № 7, 281–292.

[3] Fürer M., “Faster integer multiplication”, *STOC 2007 Proceedings*, 2007, 57–66.

[4] David Harvey, Joris van der Hoeven, "Integer multiplication in time $O(n \log n)$", *Annals of Mathematics*, **193**:2 (2021), 563–617.

[5] Christoph Burnikel C., Ziegler J., "Fast Recursive Division", *Max-Planck-Institut für Informatik*, 1998.

[6] Gasanov E. E., "Linear in order algorithm for multiplication of numbers using a two-dimensional cellular automaton with locators", *International scientific conference "Mathematics in the constellation of sciences" to the anniversary of academician V.A. Sadovnichy, Moscow, Russia, April 1-2, 2024*, p.316-318 (In Russian).

[7] Gasanov E. E., "Cellular automata with locators", *Intelligent Systems. Theory and Applications*, **24**:2 (2020), 120–133 (In Russian).

[8] Kalachev G. V., "Notes on the Definition of a Cellular Automaton with Locators", *Intelligent Systems. Theory and Applications*, **24**:4 (2020), 47–56 (In Russian).

[9] Ibragimova D. E., "Vector addition on a line using a cellular automaton with locators", *Intelligent Systems. Theory and Applications*, **26**:4 (2022), 134–162 (In Russian).

[10] Gasanov E. E., "Cellular automata with locators as a model for wireless communication devices", *Mathematical issues of cybernetics*, **21** (2023), 5–51 (In Russian).

[11] Vasilev D. I., "Finding the nearest neighbor on a line using a cellular automaton with locators", *Intelligent Systems. Theory and Applications*, **24**:3 (2020), 99–119 (In Russian).

К сведению авторов публикаций в журнале «Интеллектуальные системы. Теория и приложения»

В соответствии с требованиями ВАК РФ к изданиям, входящим в перечень ведущих рецензируемых научных журналов и изданий, в которых могут быть опубликованы основные научные результаты диссертаций на соискание ученой степени доктора и кандидата наук, статьи в журнал «Интеллектуальные системы. Теория и приложения» предоставляются авторами в следующей форме:

1. Статьи, набранные в пакете LaTeX, предоставляются к загрузке через WEB-форму http://intsysmagazine.ru/generator_form .
2. К статье прилагаются файлы, содержащие название статьи на русском и английском языках, аннотацию на русском и английском языках (не более 50 слов), список ключевых слов на русском и английском языках (не более 20 слов), информация об авторах: Ф.И.О. полностью, место работы, должность, ученая степень и/или звание (если имеется), для аспирантов ФИО научного руководителя, контактные телефоны (с кодом города и страны), e-mail, почтовый адрес с индексом города (домашний или служебный).
3. Список литературы оформляется в едином формате, установленном системой Российского индекса научного цитирования. Список на русском языке приводится в конце файла с текстом статьи, в то время как список, переведённый на английский язык, прилагается отдельным файлом.
4. За публикацию статей в журнале «Интеллектуальные системы. Теория и приложения» с авторов (в том числе аспирантов высших учебных заведений) статей, рекомендованных к публикации, плата не взимается. Авторам бесплатно предоставляется номер журнала, в котором вышла статья. Журнал распространяется по подписке, экземпляры журнала рассылаются подписчикам наложенным платежом. Условия подписки публикуются в каталоге НТИ «Роспечать», индекс журнала 64559.
5. Доступ к электронной версии последнего вышедшего номера осуществляется через НЭБ «Российский индекс научного цитирования». Номера, вышедшие ранее, размещаются на сайте

http://intsysmagazine.ru,

и доступ к ним бесплатный. Там же будут размещены полные тексты всех публикуемых статей.

Подписано в печать: 25.09.2024
Дата выхода: 30.09.2024
Тираж: 200 экз.
Цена свободная
Свидетельство о регистрации СМИ: ПИ № ФС77–58444 от 25 июня 2014 г., выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций(Роскомнадзор).