

Байесовская деконволюция для эксперимента СФЕРА-2

Игорь Вайман

5 мая 2021 г.

Глава 1

Описание эксперимента СФЕРА-2

Эксперимент СФЕРА-2 основан на сборе и регистрации черенковского света ШАЛ, отражённого от «экрана» – ровной заснеженной поверхности льда озера Байкал. Установка представляет собой сферическое зеркало и мозаику ФЭУ, установленную вблизи фокальной поверхности. Черенковский свет ШАЛ, рассеиваясь на снегу, проходит через диафрагму установки, отражается от зеркала, и попадает на фотокатод. При попадании фотона с длиной волны в области чувствительности фотокатода с него выбиваются 1 – 10 (уточнить) вторичных электронов, которые проходят систему динодов с разностью потенциалов между каждой парой, создавая в результате избыточный на аноде. Этот заряд стекает с анода через цепь усиления, создавая в результате напряжение на входе АЦП. Это напряжение считывается электроникой, в результате и составляя (вместе с некоторой телеметрией) экспериментальные данные.

Этот набор физических процессов мы описываем в следующих положениях:

1. В системе нет электронных шумов – единственным источником шума являются фоновые фотоны (от звёздного и зодиакального света, так как эксперимент проводится в ясные безлунные ночи). Фотоны шума попадают в установку аналогично «сигнальным» фотонам – после рассеяния на поверхности. Их поток можно приблизительно оценить по каталогам светимости звёзд и из моделей зодиакального света. (Ссылка на работу Энтиной)

2. Работу ФЭУ можно эффективно описать одной «рандомизированной» импульсной характеристикой (РИХ). Понятие и применение РИХ будет раскрыто далее, в общем виде это случайная функция времени, описывающая отклик системы на один фотон, падающий на фотокатод. Главный источник случайности в установке СФЕРА-2 – неопределённость числа вторичных фотонов, вылетевших с фотокатода, приводящая к неопределённости амплитуды импульса тока. Стоит отметить, что случайный характер ИХ проявляется именно на границе между режимом счёта фотонов и режимом измерения потока, поскольку поток фотонов уже слишком велик, чтобы разрешать отдельные импульсы, но недостаточно – чтобы случайные характеристики эффективно усреднялись. Мы также предполагаем, что случайность импульсной характеристики проявлялась независимо между отдельными фотонами и между разными ФЭУ.
3. Используя экспериментальные данные (данные, записанные АЦП, постоянную компоненту анодного тока и абсолютную калибровку ФЭУ) мы можем восстановить показания анодного тока (ссылка на работу по электронике). Неопределённость, вносимая анодной цепью, усилителем и (главным образом) дискретизацией сигнала на входе АЦП, хорошо известна и учитывается во вторую очередь.

Глава 2

Байесовская деконволюция

2.1 Общие допущения

Рассмотрим N последовательных равновеликих временных бинов. Для простоты будем считать единицей времени длительность одного бина. Тогда мы можем задать временные бины интервалами $[i-1, i]$, $i = 1 \dots N$. Обратим внимание, что при такой нумерации бинов эффект от фотонов, попавших в i -тый временной бин проявляется впервые в момент времени i . Обозначим число фотонов в каждом бине как n_i , $i = 1, \dots, N$.

Время прихода отдельного фотона относительно начала бина $t_{inbin} = \{t\}$ будем описывать случайной величиной, подразумевая, что мы не интересуемся такими детальными характеристиками сигнала. Эта величина может быть, вообще говоря, распределена произвольным образом в интервале $[0, 1)$, однако мы в простейшем случае будем считать $t_{inbin} \sim U(0, 1)$. Это оправдано для независимых друг от друга фоновых фотонов, и может служить приближением для фотонов ШАЛ в случае, если дисперсия времён прихода фотонов внутри «пакета» сильно превышает длительность временного бина. По данным модельных ливней это не всегда так, поэтому влияние неравномерности распределения времён прихода фотонов будет исследовано отдельно.

Предположим, что импульсная характеристика системы – случайная функция $\tilde{h}(t)$ в том смысле, что для входного сигнала, состоящего из конечного числа дельта-функций на единицу времени, каждая из этих дельта функций сворачивается с отдельной независимой реализацией $h(t) \sim \tilde{h}(t)$. Будем также считать, что любая реализация удовлетворяет

условию каузальности, то есть $\forall t < 0 \quad \forall h \sim \tilde{h} \quad h(t) = 0$, и конечности во времени, то есть $\exists \tilde{L}$ такое, что $\forall t > \tilde{L} \quad \tilde{h}(t) = 0$.

Заметим, что эффект от фотонов в i -том бине проявляется в отсчётах с i по $i + \lfloor \tilde{L} \rfloor$, где $\lfloor \tilde{L} \rfloor$ – наибольшее целое число, не превышающее \tilde{L} (округление вниз). Обозначим $L \equiv \lfloor \tilde{L} \rfloor$. Тогда полный сигнал от фотонов гарантированно содержится в отсчётах с 1 по $N + L$.

Таким образом, фотоны, приходящие на вход системы в течение времени $[0, N]$ будут приводить к регистрации отсчётов s_j в моменты времени $j = 1, \dots, N + L$.

Для иллюстрации в течение всей главы 2 будем использовать следующую рандомизированную импульсную характеристику:

$$\begin{aligned} \tilde{h}(t) &= \tilde{A} \exp(-t/2) \\ \tilde{A} &\sim U(0.75, 1) \\ L &= 5 \end{aligned} \tag{2.1}$$

В качестве входных данных сгенерируем сигнал, качественно соответствующий фотонному шуму в эксперименте СФЕРА-2:

$$\begin{aligned} N &= 30 \\ \vec{n} &= \{n_i \sim \text{Poisson}(\lambda) \quad \forall i \in [1, N]\} \\ \lambda &= 15 \end{aligned} \tag{2.2}$$

Иллюстрация \vec{n} и \vec{s} для этой демонстрационной задачи приведена на рис. 2.1.

2.1.1 Эффект окна

В реальном эксперименте входные фотоны не ограничены интервалом $[0; N]$, но приходят постоянно. Модифицируем постановку задачи так, чтобы устранить эффект окна, то есть влияние – плавный рост сигнала в начале и затухание в конце.

Для этого достаточно исключить из всего дальнейшего рассмотрения эти участки сигнала. Так, в реальной ситуации фотоны продолжают приходить после $t = N$, и вносят соответствующий вклад в отсчёты начиная с $N - 1$. Фотоны, пришедшие до $t = 0$, вносят вклад в отсчёты до L .

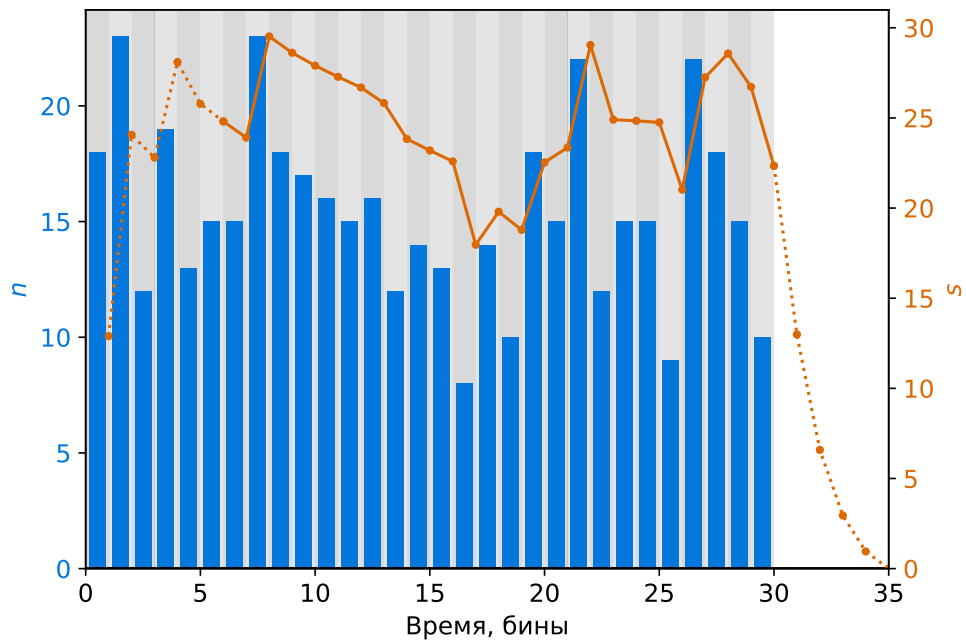


Рис. 2.1: Пример входных и выходных данных для задачи байесовской деконволюции. РИХ определена выражениями (2.1); пунктиром показаны участки, исключаемые из рассмотрения для устранения эффекта окна.

Поэтому для восстановления значений n_i , $i = 1, \dots, N$ мы будем использовать только отсчёты S_j при $j = L + 1, \dots, N$. Этот участок изображён на рис. 2.1 сплошной линией, участки по краям, исключаемые из рассмотрения – пунктиром.

2.2 Постановка задачи

Поставим задачу статистической деконволюции следующим образом, используя байесовскую терминологию (поэтому будем также называть эту процедуру байесовской деконволюцией):

Пусть дана рандомизированная импульсная характеристика системы $\tilde{h}(t)$ и значения s_j , $j = 1, \dots, N + L$. Найти апостериорные функции плотности вероятности для значений n_i , $i = 1, \dots, N$.

Заметим, что, в отличие от обычной деконволюции, мы не ставим задачу оценить исходный сигнал сам по себе, представляющий собой сумму δ -функций, но только его несколько обобщённую характеристику.

2.3 Решение

2.3.1 Выходной сигнал как реализация случайного процесса

Ясно, что в силу случайного характера отклика системы, а также агрегирования фотонов в бины, значения отсчётов $\{s_j\}$ являются реализациями некоторых случайных величин. Обозначим сами эти случайные величины как $\{S_j\}$.

Запишем S_j как сумму вкладов от фотонов разных бинов

$$S_j = \sum_{l=0}^L C(n_{j-l}, l) \quad (2.3)$$

Здесь $C(n, l)$ – случайная величина, описывающая вклад в сигнал на j -том временном отсчёте от n фотонов в бине $j - l$, иначе говоря, вклад с *задержкой* l бинов. Из выбранной схемы индексации бинов и

условий каузальности и ограниченности во времени РИХ легко видеть, что $l \in [0, L]$, поскольку вклад от фотонов более ранних бинов равен нулю.

Охарактеризуем распределение $C(n, l)$. Проще всего сделать это через Монте-Карло-сэмплирование распределения этой величины. Получим сначала с произвольной точностью эмпирическую функцию плотности распределения для $C(1, l)$. Для этого сгенерируем значения $t_k \sim t_{inbin}$; и функции $h_k(t) \sim \tilde{h}(t)$ для $k = 1 \dots N_{sample}$. Выборка для $C(1, l)$ тогда будет состоять из значений $h_k(l+1-t_k)$. Выборка для $C(n, l)$ легко получить, проделав описанную процедуру n раз и сложив все n реализаций N_{sample} -мерных векторов выборок.

2.3.2 Грубая оценка

Перед тем, как решать задачу деконволюции в статистическом смысле, сделаем грубую оценку \vec{n} , основанную только на соотношениях между математическими ожиданиями случайных величин. Для этого применим операцию вычисления математического ожидания к обеим частям равенства (2.3).

Фотоны независимы друг от друга в пределах одного бина, из чего следует $\mathbb{E} C(n, l) = n \mathbb{E} C(1, l)$. Нетрудно заранее вычислить для данной РИХ выборки значений $C(1, l)$ для $l = 0 \dots L$. Тогда получим, обозначая $c_l \equiv \mathbb{E} C(1, l)$,

$$\bar{S}_j = \mathbb{E} S_j = \sum_{l=0}^L \mathbb{E} C(n_{j-l}, l) = \sum_{l=0}^L n_{j-l} \mathbb{E} C(1, l) = \sum_{l=0}^L n_{j-l} c_l \quad (2.4)$$

Суммирование можно записать в матричном виде для $j = 1, \dots, N + L$:

$$\begin{pmatrix} c_0 & 0 & 0 & \cdots & 0 \\ c_1 & c_0 & 0 & \cdots & 0 \\ c_2 & c_1 & c_0 & & \vdots \\ c_3 & c_2 & c_1 & \ddots & 0 \\ \vdots & c_3 & c_2 & \ddots & c_0 \\ c_L & \vdots & c_3 & \ddots & c_1 \\ 0 & c_L & \vdots & \ddots & c_2 \\ \vdots & & c_L & & c_3 \\ 0 & \cdots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & c_L \end{pmatrix} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ \vdots \\ n_N \end{pmatrix} = \begin{pmatrix} \bar{S}_1 \\ \bar{S}_2 \\ \vdots \\ \bar{S}_N \\ \bar{S}_{N+1} \\ \vdots \\ \bar{S}_{N+L} \end{pmatrix} \quad (2.5)$$

Для исключения граничных эффектов, и тем самым перехода к случаю непрерывного потока фотонов достаточно, как указано в разделе 2.1.1, ограничиться рассмотрением строк $j = L+1, \dots, N$, то есть убрать первые и последние L уравнений системы.

Эта возможно несовместная система линейных уравнений, допускает решение в смысле наименьших квадратов с помощью псевдообратной матрицы Мура-Пенроуза [5]. Псевдообратная матрица C^+ для C определяется следующими условиями: (1) $CC^+C = C$, (2) $C^+CC^+ = C^+$, (3) CC^+ и C^+C – эрмитовы матрицы. Псевдообратная матрица всегда существует, и для системы $C\vec{n} = \vec{S}$ вектор $C^+\vec{S}$ даёт искомое МНК-решение системы. Приближённый численный расчёт такой матрицы можно провести, например, с помощью функции `pinv` модуля `numpy.linalg` в Python [4].

Однако для решения системы необходимо знать $\mathbb{E} S_j$, в то время как в эксперименте мы имеем всего лишь единственную реализацию этой случайной величины s_j . Для грубой оценки остаётся положить $\mathbb{E} S_j \cong s_j$. Результат описанной процедуры приведён на рис. 2.2 фиолетовым, пунктиром отмечена область, в которой оценка искажена эффектом окна.

Строгое исследование свойств такой «one-shot» оценки \vec{n} находится за рамками данной работы. Однако для простой численной проверки можно провести описанную процедуру для большего числа входных бинов и сравнить распределения истинных и оцененных значений \vec{n} . На рис. 2.3 приведено такое сравнение для 1000 бинов. Видно, что распределения практически совпадают, а значит отсутствует по крайней мере система-

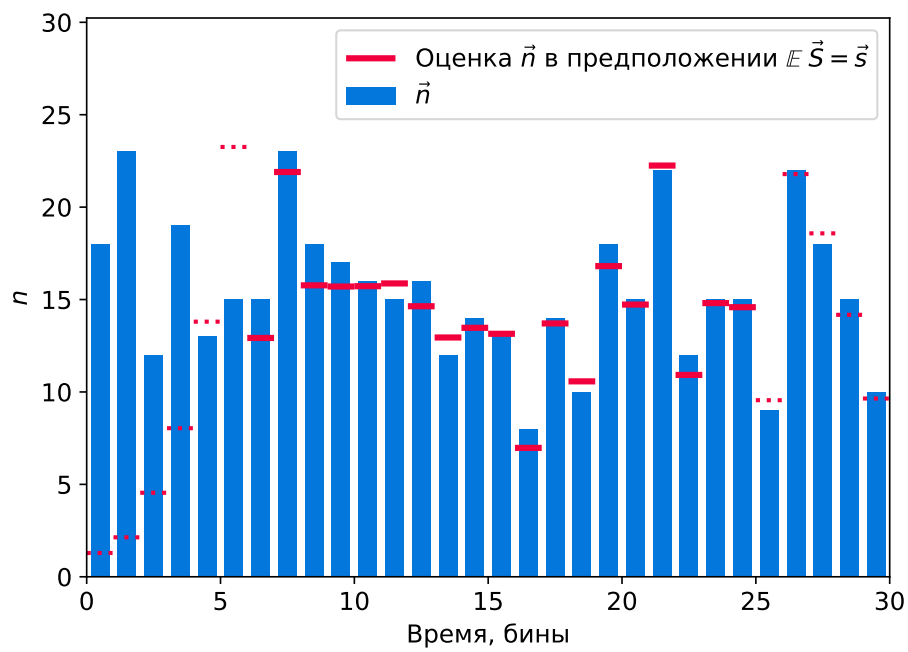


Рис. 2.2: Оценка \vec{n} в предположении, что выходной сигнал \vec{s} равен своему математическому ожиданию. Пунктиром показана область, где оценка искажена эффектом ограниченности выборки во времени.

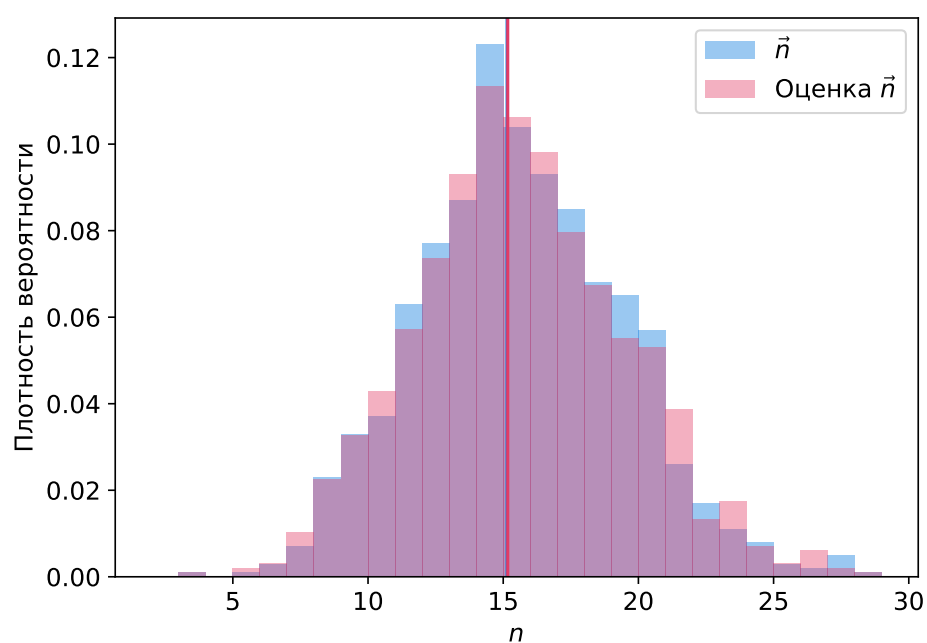


Рис. 2.3: Распределения истинных и грубо восстановленных значений \vec{n} для 1000 значений. Истинные значения выбраны из пуассоновского распределения с $\lambda = 15$, аналогично рис. 2.1 и 2.2.

тическая ошибка. В дальнейшем эта оценка будет играть роль первого приближения, или стартовой точки, на основе которой уже полными статистическим методом можно найти полное решение.

2.3.3 Решение задачи байесовской деконволюции

Теперь перейдём к решению полноценной статистической задачи, используя метод, описанный в предыдущем разделе, как первое приближение.

Запишем сначала теорему Байеса в общем виде, учитывая, что наблюдаемыми значениями является сигнал \vec{s} , а неизвестными параметрами, которые задают распределение наблюдаемых – \vec{n} :

$$P(\vec{n}|\vec{s}) = \frac{P(\vec{s}|\vec{n}) P(\vec{n})}{P(\vec{s})} \quad (2.6)$$

Поясним вероятности, входящие в выражение:

1. $P(\vec{n}|\vec{s})$ – искомое *апостериорное* распределение, описывающее (в байесовском определении вероятности) наши знания о \vec{n} после проведения измерений
2. $P(\vec{s}|\vec{n}) \equiv \mathcal{L}(\vec{s}, \vec{n})$ – функция правдоподобия, описывающая, насколько вероятна регистрация определённых значений \vec{s} при заданных параметрах \vec{n}
3. $P(\vec{n}) \equiv \pi(\vec{n})$ – *априорное* распределение \vec{n} , не требующее знаний о конкретном выходном сигнале
4. $P(\vec{s})$ – полная вероятность регистрации данного сигнала при всех возможных значениях \vec{n} . Она также называемая маргинальной вероятностью или нормировочным множителем, по определению $P(\vec{s}) = \int_{\infty} P(\vec{s}|\vec{n}) P(\vec{n}) d\vec{n}$. Эту величину можно использовать для сравнения моделей, например, если бы мы имели в распоряжении альтернативную модель генерации значений \vec{s} и хотели бы понять, какая из двух лучше описывает данные. Однако для поставленной задачи нет нужды ни вычислять, ни даже учитывать этот множитель.

Учитывая введённые обозначения, запишем

$$P(\vec{n}|\vec{s}) \propto \mathcal{L}(\vec{s}, \vec{n}) \pi(\vec{n}) \quad (2.7)$$

2.3.3.1 Выбор априорного распределения

В качестве априорного распределения будем использовать неограниченное равномерное, или *неинформативное* распределение. Методы байесовской статистики работают даже в ситуации, когда у нас нет вообще никакой информации о распределении \vec{n} до начала измерений. Формально для этого нужно положить $\pi(\vec{n}) = \text{Const} \forall \vec{n}$. Такое распределение нельзя использовать напрямую, поскольку его невозможно отнормировать на 1. Однако тогда в выражении 2.7 можно просто исключить $\pi(\vec{n})$ из правой части.

2.3.3.2 Полное вычисление функции правдоподобия методом Монте-Карло

Функция правдоподобия $\mathcal{L}(\vec{s}, \vec{n})$ определяется как вероятность того, что данный сигнал \vec{s} получился в результате преобразования системой входного сигнала \vec{n} .

Опишем сначала Монте-Карло метод оценки правдоподобия, и затем введём упрощения, которые позволят эффективнее вычислять эту функцию, а также обоснуем корректность этих упрощений. Для этого при фиксированных \vec{n} необходимо смоделировать выборку из большого числа реализаций случайной величины \vec{S} , а затем оценить плотность вероятности в точке \vec{s} по этой выборке.

Существует несколько методов для такой оценки, самый простой из которых – многомерная гистограмма. Для этого пространство реализаций \vec{S} нужно поделить на ячейки, подсчитать число элементов выборки, попавших в каждую ячейку, и поделить на общее число элементов выборки и на объём ячейки. Нетрудно видеть, что полученное число даст оценку плотности вероятности в произвольной точке – а значит, искомую оценку функции правдоподобия.

Однако оказывается, что для поставленной задачи метод прямого вычисления функции правдоподобия плохо подходит из-за «проклятия размерности». Размерность пространства, в котором нужно оценить эмпирическую функцию плотности вероятности, равна ширине окна, в котором рассматривается входной сигнал, в бинах. Соответственно, общее число N -мерных бинов составляет g^N , где g – мощность бинирования каждого отсчёта. В этой ситуации для надёжной оценки придётся генерировать выборку сравнимого объёма.

Поэтому вместо прямого использования описанного метода аппроксимируем распределение \vec{S} более вычислительно эффективной функцией.

2.3.3.3 Аппроксимация функции правдоподобия многомерным нормальным распределением

Проанализируем результат полной Монте-Карло симуляции, чтобы исследовать распределение $p(\vec{S}|\vec{n})$ и показать, что его в самом деле можно эффективно и с высокой точностью аппроксимировать многомерным нормальным распределением.

Частные распределения Из самых общих соображения ясно, что распределение $S_j \forall j$ будет более или менее близко к нормальному – просто в силу того, что каждая из S_j является суммой независимых (хотя и не всех одинаково распределённых) вкладов от фотонов в предыдущих бинах. Также сразу можно сказать, что *чем меньше среднее число фотонов в бине, тем хуже будет аппроксимация нормальным распределением*. На рис. 2.4 приведены несколько распределений «лучшего», «медианного» и «худшего» распределений S_j для нескольких значений среднего числа фотонов в бине $\lambda_n \equiv \mathbb{E} n_i$. Из предварительного анализа и независимых соображений известно, что менее 5 фотонов на бин – весьма нечастая ситуация в экспериментальных данных – но даже в этом случае видно, что самое существенное среднеквадратичное отклонение плотности распределения от гауссианы не превышет 10%. Таким образом, *частное распределение* каждой компоненты вектора \vec{S} можно приблизить нормальным распределением с эффективной погрешностью в несколько процентов для практически значимых случаев.

Полное $(N - L)$ -мерное распределение Провести аналогичную процедуру для полного $(N - L)$ -мерного распределения – непростая задача, поэтому ограничимся здесь визуальным исследованием двухмерных распределений для пар $(S_j; S_k)$ для $k - j < L$ (из постановки задачи ясно, что на больших расстояниях элементы \vec{S} независимы). На рис. 2.5 представлены такие распределения. Видно, что двухмерные распределения представляют собой овалы симметричной формы, как и ожидается от многомерного нормального распределения.

Таким образом, можно положить, что *распределение вектора \vec{S} при заданных \vec{n} можно приблизить многомерным нормальным распреде-*

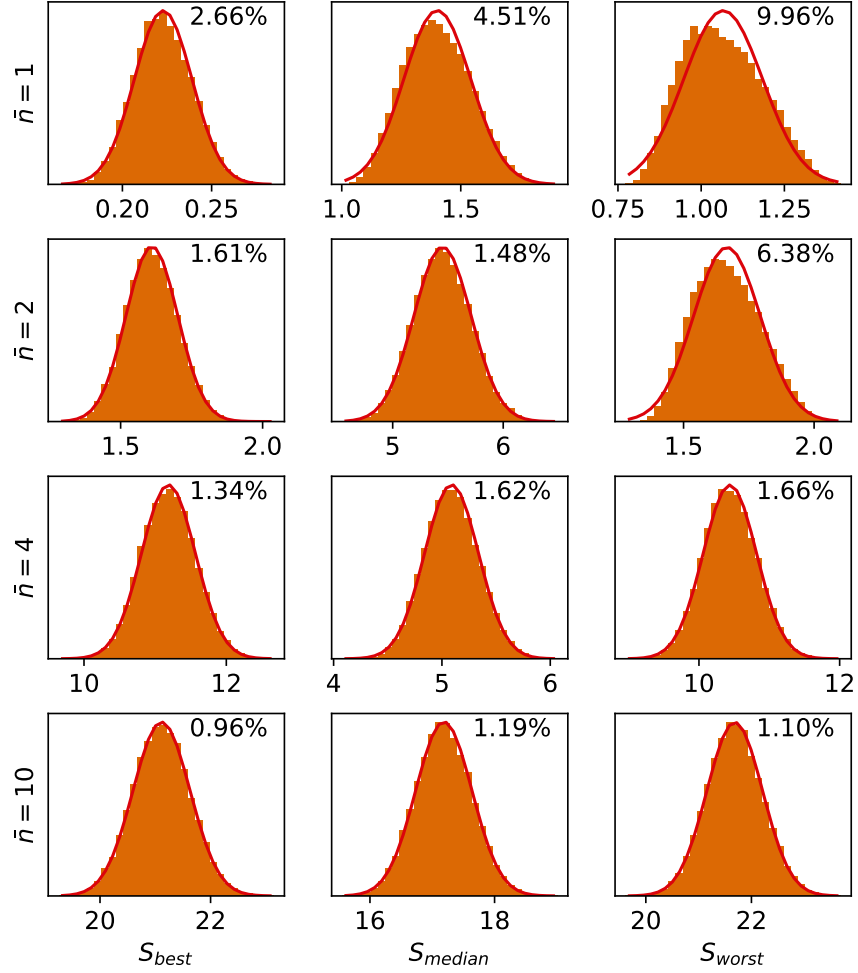


Рис. 2.4: Оценка нормальности частных распределений компонент \vec{S} для разных интенсивностей входного потока: от экстремально низких 1 и 2 до средней ожидаемой 10. «Лучший», «медианный» и «худший» бины определены эвристически по квадрату разницы выборочного среднего и медианы (чем больше эта величина, тем менее симметрично распределение). На каждом графике построена плотность нормального распределения с соответствующими μ и σ , а также приведено среднеквадратичное отклонение гистограммы от гауссианы в процентах относительно максимального значения.

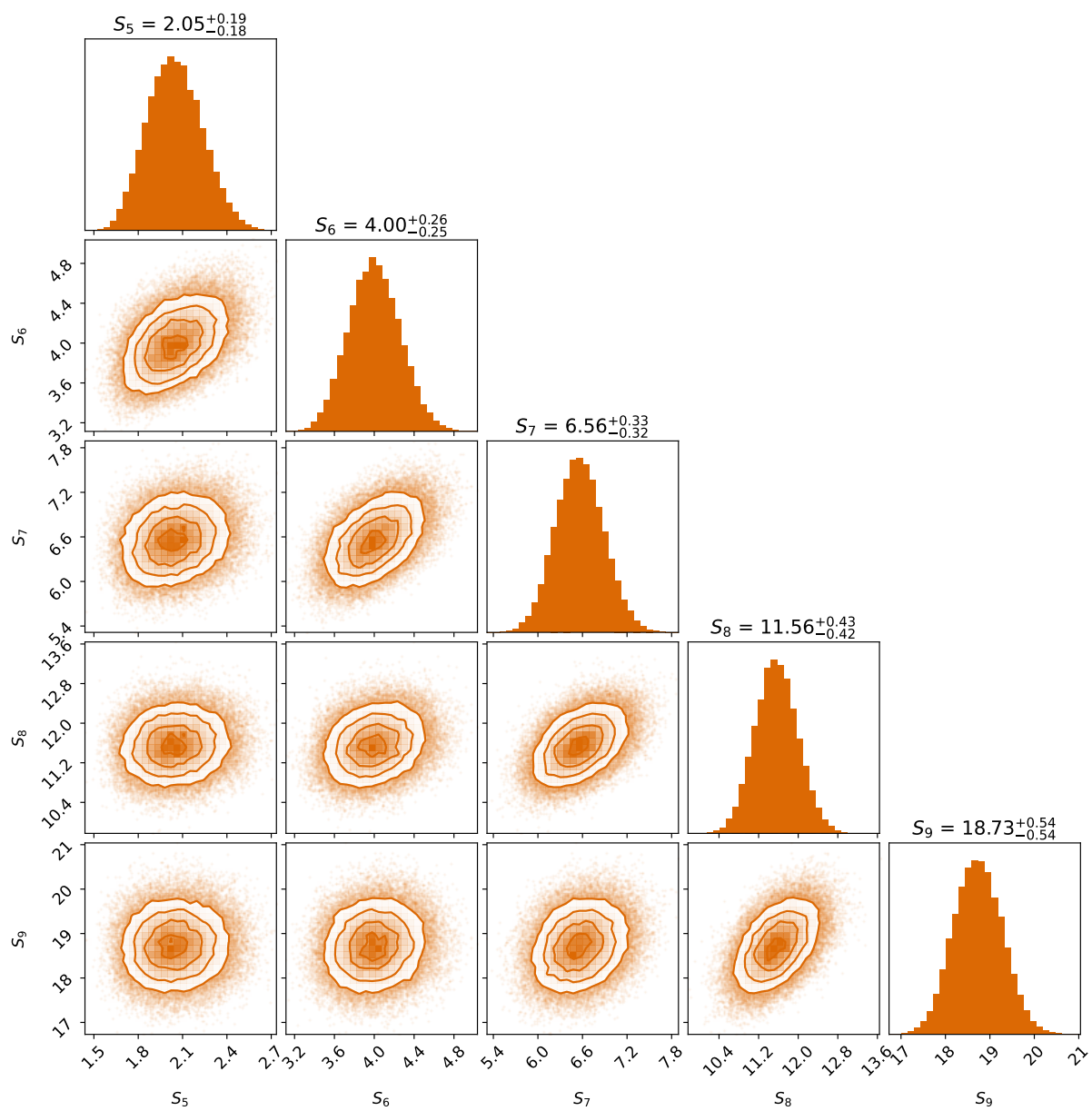


Рис. 2.5: Оценка нормальности совместных двумерных распределений для пар компонент \vec{S} . График построен с помощью библиотеки corner [2].

лением. Это приближение существенно упрощает вычисление функции правдоподобия $\mathcal{L}(\vec{n}) \equiv P(\vec{S}|\vec{n})$, которая и сводится к вычислению плотности вероятности условного распределения \vec{S} в точке, заданной наблюдаемым сигналом. Этот метод оказывается гораздо более устойчивым и вычислительно эффективным, чем наивный метод Монте-Карло, описанный в разделе 2.3.3.2.

Многомерное нормальное распределение в общем виде задаётся вектором математических ожиданий $\vec{\mu}$ и матрицей ковариаций Σ . Функция плотности вероятности $(N + L)$ -мерного случайного вектора \vec{S} тогда вычисляется как

$$p(\vec{S}) = ((2\pi)^{N+L} \det \Sigma)^{-1/2} \exp \left(-\frac{1}{2} (\vec{S} - \vec{\mu})^T \Sigma^{-1} (\vec{S} - \vec{\mu}) \right) \quad (2.8)$$

Вектор $\vec{\mu} \equiv \mathbb{E} \vec{S}$ при данных \vec{n} вычисляется в соответствии с выражением (2.5). Вычислим произвольный элемент матрицы $\Sigma_{ij} \equiv \text{cov}(S_i, S_j)$. В силу симметричности матрицы ковариаций можно положить для определённости $i \leq j$. Запишем выражение (2.3) для двух интересующих нас элементов \vec{S} :

$$S_i = \sum_{l=0}^L C(n_{i-l}, l) \quad (2.9)$$

$$S_j = \sum_{k=0}^L C(n_{j-k}, k) \quad (2.10)$$

Источник возможной ненулевой ковариации этих величин – члены, описывающие вклады фотонов из одного бина, но на разных задержках, все прочие слагаемые независимы. Условие, описывающее такие попарно-скоррелированные члены, получается из равенства индексов n в суммах: $i - l = j - k$. Вводя $\Delta \equiv j - i$, отсюда получаем $k = l + \Delta$. Отсюда сразу следует интуитивный вывод о том, что при $\Delta > L$ ковариация будет заведомо нулевой в силу конечности РИХ во времени.

Используем следующее свойство ковариации: если x, y, ϵ, η – случайные величины, из которых только x и y являются зависимыми, то $\text{cov}(x + \epsilon, y + \eta) = \text{cov}(x, y)$. Иначе говоря, заведомо независимые члены при вычислении ковариации можно просто опустить. Прделаем это в

суммах выше, получив таким образом «скореллированные части» \hat{S}_i и \hat{S}_j такие, что $\text{cov}(S_i, S_j) = \text{cov}(\hat{S}_i, \hat{S}_j)$:

$$\hat{S}_i = \sum_{l=0}^{L-\Delta} C(n_{i-l}, l) \quad (2.11)$$

$$\hat{S}_j = \sum_{k=\Delta}^L C(n_{j-k}, k) = [k = l + \Delta] = \sum_{l=0}^{L-\Delta} C(n_{i-l}, l + \Delta) \quad (2.12)$$

Наконец, используем ещё одно свойство ковариации: если среди случайных величин x_1, x_2, y_1, y_2 независимы все пары, кроме x_1 и y_1, x_2 и y_2 , то $\text{cov}(x_1 + x_2, y_1 + y_2) = \text{cov}(x_1, y_1) + \text{cov}(x_2, y_2)$. Индуктивно обобщая на случай сумм с произвольным числом членов, и принимая во внимание, что фотоны в разных бинах независимы, получаем

$$\text{cov}(S_i, S_j) = \text{cov}(\hat{S}_i, \hat{S}_j) = \sum_{l=0}^{L-\Delta} \text{cov}(C(n_{i-l}, l), C(n_{i-l}, l + \Delta)) \quad (2.13)$$

Наконец, принимая во внимание, что сама по себе величина $C(n, l)$ есть сумма n независимых одинаково распределённых случайных величин, а между членами сумм $C(n, l)$ и $C(n, l + \Delta)$ есть только попарные корреляции в отношении 1 к 1, можно записать окончательную формулу для вычисления элемента матрицы ковариации:

$$\text{cov}(S_i, S_{i+\Delta}) = \sum_{l=0}^{L-\Delta} n_{i-l} \text{cov}(C(1, l), C(1, l + \Delta)) \quad (2.14)$$

Из формулы видно, что автокорреляция сигнала описана полностью в терминах автокорреляции РИХ и чисел фотонов в соответствующих бинах, служащих весами для вклада разных участков РИХ.

Для удобства вычислений можно записать это равенство в матричном виде для значений $\Delta \in [0; L]$. Для этого введём обозначение $\xi(l, \Delta) \equiv \text{cov}(C(1, l), C(1, l + \Delta))$ и с помощью него запишем:

$$\begin{pmatrix} \xi(L, 0) & \xi(L-1, 0) & \xi(L-2, 0) & \cdots & \xi(0, 0) \\ 0 & \xi(L-1, 1) & \xi(L-2, 1) & \cdots & \xi(0, 1) \\ 0 & 0 & \xi(L-2, 2) & \cdots & \xi(0, 2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \xi(0, L) \end{pmatrix} \begin{pmatrix} n_{i-L} \\ n_{i-(L-1)} \\ n_{i-(L-2)} \\ \vdots \\ n_i \end{pmatrix} = \begin{pmatrix} \Sigma_{i,i} \\ \Sigma_{i,i+1} \\ \Sigma_{i,i+2} \\ \vdots \\ \Sigma_{i,i+L} \end{pmatrix} \quad (2.15)$$

Это достаточно громоздкое и неинтуитивное выражение позволяет, однако, единожды вычислить матрицу Ξ и эффективно рассчитывать матрицу ковариаций Σ при каждом заданном \vec{n} .

После исключения областей влияния эффекта окна – первых и последних L элементов \vec{S} – нас будут интересовать только внутренняя часть матрицы Σ , то есть индексы $i, j \in [L+1; N]$. Это удобно, так как для вычисления интересующей нас части Σ не нужно рассматривать «урезанную» матрицу Ξ , что потребовалось бы, если бы мы не исключали границы.

2.3.3.4 Сэмплирование апостериорного распределения

Имея функцию правдоподобия \mathcal{L} и учитывая выбранные неинформативные априорные распределения, мы получаем функцию плотности апостериорной вероятности $P(\vec{n}|\vec{s}) \propto \mathcal{L}(\vec{n}, \vec{s})$, которую требуется исследовать.

Требуется получить не только наиболее оптимальное значение \vec{n} , но и оценку неопределённости этой величины. Поэтому, например, простой метод максимального правдоподобия, в котором мы нашли бы максимум \mathcal{L} в пространстве параметров, не подходит – хотя существуют методы поиска доверительных интервалов. Мы будем использовать метод Монте-Карло с марковскими цепями (Markov Chain Monte Carlo), который хорошо применим к задачам большой размерности. Суть метода сводится к запуску марковского процесса случайного блуждания в пространстве параметров \vec{n} с такой специально подобранной вероятностью перехода, зависящей от апостериорной вероятности, что в пределе это блуждание даст выборку из исследуемой функции плотности. Общей характеристикой этих методов является то, что они зависят только от отношения вероятностей в разных точках пространства параметров – именно поэтому в выражении 2.7 и во всех последующих вычислениях мы не интересовались ни полной маргинальной вероятностью наблюдаемых данных, ни

нормировкой априорного распределения.

Подробный обзор и теоретическое исследование весьма широкой области МСМС-сэмплирования находятся за рамками данной работы, но могут быть найдены, например, в обзоре [6].

Аффинно-инвариантное МСМС-сэмплирование В данной работе будем использовать популярный алгоритм аффинно-инвариантного сэмплирования, предложенный в 2010 году в работе [3].

В этом алгоритме используется не одна марковская цепь (англ. *walker*), а целый ансамбль из нескольких сотен или даже тысяч цепей, и новый шаг генерируется для каждой цепи в зависимости от состояний остальных. Конкретный алгоритм генерации шага (или просто «шаг», англ. *move*) не фиксирован и может быть выбран в зависимости от конкретной задачи. Каждый из шагов обладает своими особенностями и по-своему влияет на предпочтительное количество цепей и необходимое количество итераций. Самый универсальный – так называемый «шаг-растяжка» (англ. *stretch move*), опишем его вкратце.

Рассмотрим i -тую цепь, находящуюся в состоянии $X_i(t)$ (состояние цепи есть точка в пространстве параметров, иначе говоря, конкретное значение \vec{n}). Выберем случайным образом комплементарную цепь из ансамбля и обозначим её $X_j(t)$, $j \neq i$. Сгенерируем «предложение» (англ. *proposal*) для данной цепи:

$$Y = X_j(t) + Z (X_i(t) - X_j(t)) \quad (2.16)$$

Предложенное новое положение i -той цепи находится на прямой, соединяющей её с другой цепью на данном шаге. Масштабный множитель Z – случайная величина, на которую накладывается важное теоретическое ограничение: для существования стационарного распределения марковской цепи достаточно обеспечить детальное равновесие, то есть равенство вероятностей перехода $X_i \rightarrow Y$ и $Y \rightarrow X_i$. Для этого достаточно потребовать, чтобы функция плотности вероятности величины Z должна удовлетворять свойству симметрии $g(1/z) = z g(z)$. В качестве конкретного вида этой функции авторы предлагают

$$g(z) \propto \begin{cases} \frac{1}{\sqrt{z}} & \text{при } z \in [\frac{1}{a}, a], \\ 0 & \text{в остальных случаях} \end{cases} \quad (2.17)$$

Наконец, после генерации предложения для нового положения i -той цепи (не важно, описанным шагом-растяжкой или любым другим алгоритмом) требуется принять или отвергнуть предложенное значение Y . Вероятность того или иного исхода получается из условия *частичного ресэмплинга*, которое гласит: чтобы переход ансамбля цепей в новое состояние не менял совместного распределения всего ансамбля (которое полагается равным целевому сэмплируемому распределению), достаточно, чтобы переход одной цепи не менял условную плотность вероятности для данной цепи при фиксированных всех остальных. В конце концов, это сложное условие сводится к обычному детальному равновесию, но записанному для всего ансамбля цепей. Вероятность принять предложение Y , обеспечивающая детальное равновесие, задаётся величиной

$$P(X_i(t+1) = Y) = \min \left\{ 1, Z^{N-1} \frac{p(Y)}{p(X_i(t))} \right\} \quad (2.18)$$

Упрощённо можно сказать, что цепи ансамбля будут с большей вероятностью переходить в области большего значения сэмплируемой функции (то есть исследуемой плотности апостериорной распределения), но при этом также чаще оставаться вблизи «соседей» по ансамблю.

Имплементация МСМС-сэмплирования в пакете emcee Эффективная и устойчивая реализация алгоритма МСМС-сэмплирования представляет собой непростую задачу. В приложении **TBD** приведена авторская реализация процедуры в **MATLAB**, однако в дальнейшем будет использоваться популярная имплементация описанного алгоритма в пакете **emcee** на языке **Python** [2].

Пакет предоставляет возможность не только провести сэмплирование, но и оценить корректность результата. Строгое математическое доказательство позволяет установить только *асимптотическую* сходимость ансамбля цепей к целевому распределению. При этом реальная длина цепей конечна, и степень сходимости требует отдельной оценки – иначе говоря, количество шагов должно быть *достаточно большим* для использования результата.

Но даже если ансамбль сошёлся с хорошей точностью, из него ещё требуется получить выборку независимых реализаций случайной величины. Состояние ансамбля на конкретном (например, последнем) шаге даёт ряд значений, однако число цепей в ансамбле чаще всего мень-

ше, чем желаемое количество элементов выборки – а значит, приходится брать значения на нескольких шагах. Шаги каждой цепи принципиально зависимы, но корреляция падает с расстоянием – следовательно, для получения выборки нужно выбирать *достаточно редкие* состояния ансамбля цепей.

На практике эти две проблемы выражаются в необходимости выбрать время «разгона» (англ. *burn-in*) и «прореживание» (англ. *thinning*) цепей. Информированный выбор обоих параметров можно провести с помощью оценки *интегрированного времени автокорреляции* τ [7]. Интегрированное время автокорреляции даёт оценку числа шагов, на котором корреляция значений цепи падает в e раз. Соответственно, период разгона можно взять равным нескольким τ , и прореживать значения не реже чем каждые τ шагов. Пакет **emcee** предоставляет готовые средства для оценки величины τ ¹.

Более подробное описание конфигурации сэмплера и контроля качества сэмплирования приведено в разделе **TBD**, так как оно зависит от конкретной используемой РИХ.

Помимо метопараметров (количество цепей, параметр a для «шага-растяжки» или соответствующих параметров для других шагов) важной является инициализация сэмплера. Несмотря на теоретически предсказываемое «забывания» состояния цепей в течении нескольких τ , в большинстве практических задач начальное положение цепей желательно задавать разумным образом. Здесь пригодится результат раздела 2.3.2 – простым и быстрым способом можно получить \vec{n}_{rough} – грубую оценку \vec{n} . Следуя практической рекомендации из статьи [2], инициализируем N_{walker} цепей значениями, близкими к этой оценке, но дополнительно разбросанными в соответствии с нормальным распределением числа фотонов в каждом бине. Стандартное отклонение этого начального разброса должно быть заметно меньше разброса самих данных², например $\hat{\sigma}_{rough}/10$, где $\hat{\sigma}_{rough}$ – выборочное стандартное отклонение, рассчитанное по элементам \vec{n}_{rough} .

Результат таким образом проведённого сэмплирования для демонстрационной задачи приведён на 2.6.

¹*Autocorrelation analysis & convergence*. 2019. URL: <https://emcee.readthedocs.io/en/stable/tutorials/autocorr/>.

²«Another general approach is to start the walkers in a very tight N-dimensional ball in parameter space around one point that is expected to be close to the maximum probability point.» ([2])

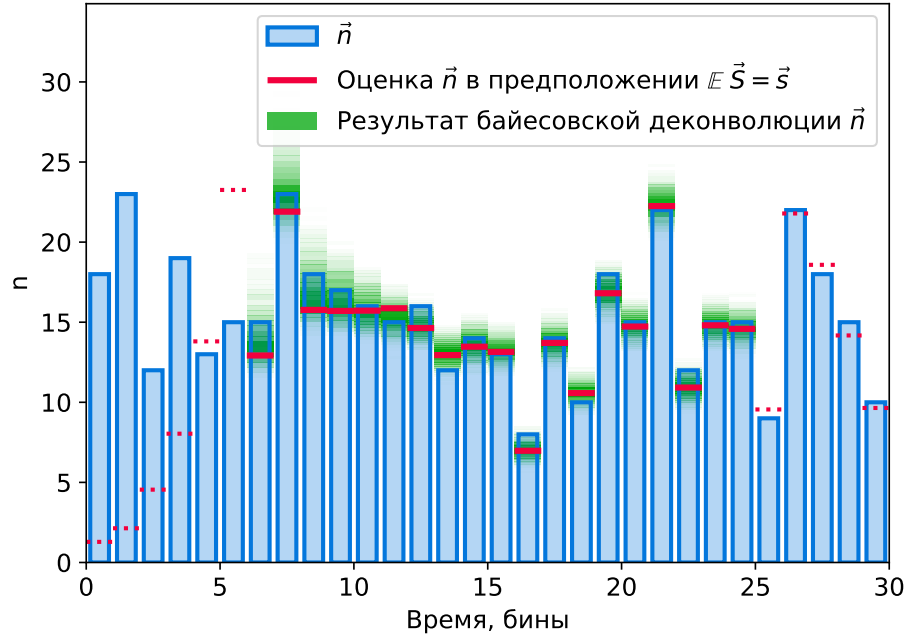


Рис. 2.6: Результат решения задачи байесовской деконволюции. Зелёным показаны *маргинальные апостериорные распределения*, то есть однопараметрические проекции апостериорного распределения как функции n_i для области i , не подверженной влиянию эффекта окна. Следует заметить, что на самом деле многомерное апостериорное распределение не обязательно факторизуемо, поэтому для дальнейшего анализа нельзя рассматривать маргинальные распределения по отдельности – они приведены только для иллюстрации результата.

2.4 Деконволюция экспериментальных данных

Общий метод решения задачи байесовской деконволюции, изложенный в предыдущем разделе, нужно теперь применить к экспериментальным данным.

2.4.1 Экспериментальная рандомизированная импульсная характеристика

Список литературы

- [1] *Autocorrelation analysis & convergence*. 2019. URL: <https://emcee.readthedocs.io/en/stable/tutorials/autocorr/>.
- [2] Daniel Foreman-Mackey. «corner.py: Scatterplot matrices in Python». в: *The Journal of Open Source Software* 1.2 (июнь 2016), с. 24. DOI: 10.21105/joss.00024.
- [3] Jonathan Goodman и Jonathan Weare. «Ensemble samplers with affine invariance». в: *Communications in Applied Mathematics and Computational Science* 5.1 (янв. 2010), с. 65—80. DOI: 10.2140/camcos.2010.5.65.
- [4] Charles R. Harris и др. «Array programming with NumPy». в: *Nature* 585.7825 (сент. 2020), с. 357—362. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [5] R. Penrose. «On best approximate solutions of linear matrix equations». в: *Mathematical Proceedings of the Cambridge Philosophical Society* 52.1 (янв. 1956), с. 17—19. DOI: 10.1017/s0305004100030929.
- [6] Sanjib Sharma. «Markov Chain Monte Carlo Methods for Bayesian Data Analysis in Astronomy». в: *Annual Review of Astronomy and Astrophysics* 55.1 (авг. 2017), с. 213—259. DOI: 10.1146/annurev-astro-082214-122339.
- [7] A. Sokal. «Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms». в: *Functional Integration*. Springer US, 1997, с. 131—192. DOI: 10.1007/978-1-4899-0319-8_6.