

Программные системы и комплексы

УДК 519.233.5

DOI:10.25729/ESI.2026.41.1.013

О программной реализации метода Монте-Карло для оценивания доверительных полос линейной регрессии

Степанов Александр Владимирович

Всероссийский научно-исследовательский институт метрологии им. Д.И. Менделеева (ВНИИМ), Россия, Санкт-Петербург, stepanov17@yandex.ru

Аннотация. Рассматривается задача расчета полос неопределенности линейной регрессии с коррелированными исходными данными. Для оценивания доверительных полос применяется обобщенный метод наименьших квадратов, при этом для оценивания их границ вводится коэффициент охвата, который, будучи умноженным на стандартную неопределенность регрессии в рассматриваемых точках, дает указанные границы. Актуальность исследования определяется тем, что стандартные методы построения доверительных интервалов, основанные на предположении о независимости ошибок, при наличии автокорреляции приводят к систематическому занижению ширины полос неопределенности. Это, в свою очередь, создает ложное впечатление о точности прогнозов и может служить причиной ошибочных статистических выводов. Для корректного построения доверительных полос необходимо учитывать структуру временной зависимости ошибок. В работе рассматриваются следующие модели коррелированных шумов: авторегрессионные процессы с экспоненциальным затуханием корреляции, а также цветные шумы, характеризующиеся степенным затуханием и долговременной памятью.

В отличие от классического случая независимых ошибок, где коэффициент охвата соответствует квантили нормального распределения, при наличии корреляции его аналитическое выражение отсутствует. Значение коэффициента напрямую зависит от структуры ковариационной матрицы ошибок, объема обучающей выборки и горизонта прогнозирования. Для его определения в работе применяется численный метод Монте-Карло в сочетании с итерационной процедурой бисекции, позволяющий находить коэффициент охвата с заданной точностью.

Разработано специализированное программное обеспечение на языке Python с использованием библиотек NumPy и SciPy. Программная реализация решает задачу оценивания полосы линейной регрессии гиперболической формы, когда корреляционная структура ошибок описывается вышеупомянутыми моделями. Приведены соответствующие примеры оценивания упомянутого коэффициента охвата и полос регрессии, дана ссылка на программную реализацию. Модульная архитектура разработанной программы допускает расширение на другие типы корреляционных структур. Применимость работы обусловлена необходимостью обеспечения корректной оценки неопределенности при статистической обработке экспериментальных данных, получаемых при решении измерительных задач.

Ключевые слова: линейная регрессия, доверительные полосы, обобщенный метод наименьших квадратов, коррелированный шум, цветной шум, метод Монте-Карло, численные методы

Цитирование: Степанов А.В. О программной реализации метода Монте-Карло для оценивания доверительных полос линейной регрессии / А.В. Степанов // Информационные и математические технологии в науке и управлении, 2026. – № 1(41). – С. 180-188 – DOI:10.25729/ESI.2026.41.1.013.

Введение. Оценивание доверительных полос линейной регрессии представляет собой фундаментальную задачу статистического анализа, особенно актуальную в условиях наличия коррелированных ошибок (под ошибками в данном контексте будем понимать разность между наблюдаемыми значениями и значениями, предсказанными рассматриваемой моделью). Пренебрежение учетом корреляционной структуры ошибок приводит к серьезному искажению статистических выводов и прогнозов, а именно, к систематической недооценке ширины доверительных полос, что создает ложное впечатление о точности оценок (прогнозов) [1, 2]. Стандартные методы, предполагающие независимость ошибок, могут существенно занижать реальную неопределенность даже при умеренной автокорреляции, что может нести серьезные негативные последствия при решении конкретных технических и научных задач (некорректной калибровке оборудования, ложным выводам о соответствии заявленным требованиям, и

т. д.). Разработка методов построения доверительных полос регрессии в случае коррелированных данных остается актуальной задачей, требующей сочетания теоретических подходов и вычислительных алгоритмов. Необходимость применения численных методов (в данном случае, метода Монте-Карло) обусловлена, в частности, отсутствием универсального аналитического решения в общем случае и ограниченностью объема выборок (что делает нецелесообразным применение асимптотических методов).

Целью представляемой работы являлась разработка программного обеспечения для оценивания доверительных полос линейной регрессии в рамках подхода, предложенного в работах [3, 4].

Для оценивания доверительных полос применяется обобщенный метод наименьших квадратов, при этом для оценивания их границ вводится коэффициент охвата, который, будучи умноженным на стандартную неопределенность регрессии в рассматриваемых точках, дает указанные границы. Программная реализация, предлагаемая в этой работе, решает задачу оценивания полосы линейной регрессии, когда корреляционная структура ошибок описывается авторегрессионными моделями (AR-моделями) [2, 5, 6] или цветными шумами [7, 8]. Обе эти модели активно используются при описании временных рядов при решении различных научных и технических задач, при наличии взаимосвязи между последующими значениями. Авторегрессионные модели адекватно отражают краткосрочные зависимости, характерные для технических измерений. Цветные шумы, такие, как, например, розовый ($1/f$) и красный ($1/f^2$), могут более точно описывать долгосрочные корреляции (в физических экспериментах, геофизических процессах и т.д.). Следует отметить различие в характере затухания корреляции для рассматриваемых моделей: AR-модели предполагают экспоненциальное затухание, в то время как цветные шумы демонстрируют степенное затухание автокорреляционной функции.

1. Оценивание полосы регрессии. Рассмотрим задачу построения доверительной полосы (коридора) неопределённости при оценивании парной регрессии:

$$Y = \beta_0 + \beta_1 X + \varepsilon(X), \quad (1)$$

при наличии экспериментальных данных $\{(x_i, y_i)\}$, $0 \leq i < N$. Величина X , в частности, может представлять собой время (т.е. x_i – моменты измерений), тогда уравнение (1) описывает линейный во времени дрейф. Далее для простоты будем считать, что значения x_i являются равноотстоящими, $x_i = ih$, $h > 0$. Значения x_i считаются измеренными точно, шум ε – в общем случае, коррелированным: $cov(\varepsilon_i, \varepsilon_j) \neq 0$, $i \neq j$, $Var(\varepsilon_i) = \sigma^2$. Ниже рассмотрим вопрос о влиянии вида шума на оценку доверительной полосы регрессии, а также предложим способ ее оценивания.

Следуя [3, 4], будем считать, что доверительная полоса (для уровня вероятности P_0) в каждой точке представляет собой произведение коэффициента K и соответствующей стандартной неопределённости регрессии в заданной точке:

$$Band = \{(x, y): 0 \leq x < (N - 1)h, |y - \hat{y}(x)| \leq Ku(\hat{y}(x))\}, \quad (2)$$

где

$$\hat{y}(x) = \beta_0 + \beta_1 x$$

– оценка регрессии, полученная по данным на интервале $x \in [0, (N_0 - 1)h]$, а коэффициент K подбирается из соображения, чтобы полоса регрессии, экстраполированная на интервал $[0, (N - 1)h]$, покрывала «истинный» тренд $y = \beta_0 + \beta_1 x$ с заданной вероятностью P_0 . Следуя сложившейся в метрологической практике традиции, коэффициент K будем называть коэффициентом охвата. Данный термин отражает его функцию: K представляет собой безразмерный коэффициент, который при умножении на стандартную неопределенность определяет интервал, охватывающий истинное значение измеряемой (прогнозируемой) величины с заданной

вероятностью, масштабируя оценку неопределенности до требуемого уровня доверия. Коэффициент K будет оцениваться ниже методом Монте-Карло, при этом, очевидно, помимо P_0 , он будет зависеть как от модели шума ε , так и от значений N_0, N .

Оценки $\hat{\beta}_0, \hat{\beta}_1$ получим с помощью обобщенного метода наименьших квадратов (ОМНК):

$$\hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} = \Theta A' V^{-1} \begin{pmatrix} y_1 \\ \dots \\ y_{N_0} \end{pmatrix}, \quad \Theta = (A' V^{-1} A)^{-1}, \quad A' = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 0 & h & \dots & (N_0 - 1)h \end{pmatrix},$$

где V – ковариационная матрица шума (штрихом обозначена операция транспонирования матрицы). Стандартная неопределенность оценки $\hat{y}(x)$ имеет вид:

$$u(x) = u(\hat{y}(x)) = \sigma \sqrt{\theta_{1,1} + 2\theta_{1,2}x + \theta_{2,2}x^2}. \quad (3)$$

При получении коэффициента охвата методом Монте-Карло в качестве модельного рассмотрим нулевой тренд ($\beta_0 = \beta_1 = 0$). В этом случае для того, чтобы доверительная полоса покрывала весь дрейф на отрезке экстраполяции $[0, (N - 1)h]$, достаточно, чтобы выполнялось неравенство $|\hat{y}(x)| \leq Ku(x)$. Таким образом, для нахождения коэффициента K методом Монте-Карло достаточно на каждой его итерации сгенерировать шум, в соответствии с выбранной моделью, и вычислить отношение

$$M = \max_x \frac{|\hat{y}(x)|}{u(x)}.$$

Далее, рассматривая совокупность значений M , как эмпирическое распределение, коэффициент K следует выбрать, как квантиль уровня P_0 этого распределения, т. е. такое значение, что доля симуляций с $M \leq K$ равна P_0 .

Имея вычисленный коэффициент K , доверительную полосу можно оценить с помощью формул (2), (3). Подчеркнем еще раз, что отсутствие замкнутой аналитической формулы для коэффициента охвата (аналитического выражения для квантилей распределения максимума M) при произвольной корреляционной структуре шума и ограниченности объема выборки делает метод Монте-Карло практически единственным универсальным обоснованным подходом к решению данной задачи.

2. Реализованные модели коррелированных шумов. Опишем используемые модели коррелированного шума ε .

2.1. Авторегрессионные модели шумов. Рассмотрим автокорреляционные модели шумов AR(1):

$$\varepsilon_k = \varphi \varepsilon_{k-1} + w_k, \quad (4)$$

и AR(2):

$$\varepsilon_k = \varphi_1 \varepsilon_{k-1} + \varphi_2 \varepsilon_{k-2} + w_k, \quad (5)$$

где w_k – белый шум (может быть как Гауссовым, так и нет), с дисперсией σ_w^2 . Ковариационные матрицы имеют следующий вид. Для (4):

$$V = \frac{\sigma_w^2}{1-\varphi^2} \begin{pmatrix} 1 & \varphi & \dots & \varphi^{N_0-2} & \varphi^{N_0-1} \\ \varphi & 1 & \dots & \varphi^{N_0-3} & \varphi^{N_0-2} \\ \dots & \dots & \ddots & \dots & \dots \\ \varphi^{N_0-2} & \varphi^{N_0-1} & \dots & \varphi & 1 \end{pmatrix};$$

для (5):

$$V = \sigma^2 \begin{pmatrix} \rho_0 & \rho_1 & \rho_2 & \dots & \rho_{N_0-1} \\ \rho_1 & \rho_0 & \rho_1 & \dots & \rho_{N_0-2} \\ \dots & \dots & \dots & \ddots & \dots \\ \rho_{N_0-1} & \rho_{N_0-2} & \rho_{N_0-3} & \dots & \rho_0 \end{pmatrix},$$

где значения ρ_i автокорреляционной функции вычисляются по формулам: $\rho_0 \equiv 1$, $\rho_1 = \frac{\varphi_1}{1-\varphi_2}$, $\rho_i = \varphi_1\rho_{i-1} + \varphi_2\rho_{i-2}$, $i \geq 2$. Дисперсию σ^2 процесса AR(2) получаем решением уравнений Юла-Уокера (в случае модели AR(2) она имеет порядок, равный трем):

$$\begin{aligned}\gamma_0 &= \varphi_1\gamma_1 + \varphi_2\gamma_2 + \sigma_w^2, \\ \gamma_1 &= \varphi_1\gamma_0 + \varphi_2\gamma_1, \\ \gamma_2 &= \varphi_1\gamma_1 + \varphi_2\gamma_0,\end{aligned}$$

где $\gamma_i = cov(\varepsilon_k, \varepsilon_{k-i})$ представляет собой автоковариацию с лагом i (соответственно, дисперсию σ^2 для $i = 0$: $\sigma^2 = \gamma_0$). Для получения i -го уравнения данной системы следует умножить исходную модель на ε_{k-i} и взять математическое ожидание.

Заметим, что условия стационарности AR-процесса имеют вид: $|\varphi| < 1$ для AR(1) и $\varphi_2 \pm \varphi_1 < 1$, $|\varphi_2| < 1$ для AR(2).

2.2. Цветные шумы. Цветные шумы с заданной спектральной мощностью (PSD) плотности $1/f^a$ генерируются спектральным методом:

- генерируется спектр шума (случайные фазы с амплитудой \sqrt{PSD});
- обратным преобразованием Фурье получается временной ряд;
- выполняется масштабирование до целевого уровня дисперсии.

Помимо параметра α логарифмического наклона спектра и целевой дисперсии шума, при задании модели также задаются частота дискретизации f_s и минимальная частота f_{min} . Автокорреляционная функция и дисперсия оцениваются по сгенерированной выборке цветного шума большой длины. Также предусмотрена генерация смеси белого и цветного шума с заданными весами (включая нормализацию до единичного уровня дисперсии); подобная модель ранее рассматривалась в работе [9].

3. Архитектура программы. Код программы и примеры использования доступны по ссылке [10]. Программа написана на языке программирования Python (версия 3.12), с использованием библиотеки NumPy, и построена по модульному принципу с использованием объектно-ориентированного подхода. Перечислим основные ее компоненты:

1. Интерфейсы: абстрактные классы для генераторов шума и калькуляторов корреляционной матрицы (заметим, что последняя в целях оптимизации используется на всех итерациях метода Монте-Карло).
2. Реализации AR-моделей – классы для генерации AR(1) и AR(2) процессов.
3. Реализации цветных шумов – классы генерации шумов, спектральная плотность мощности которых описывается законом $S(f) \sim 1/f^a$.
4. Реализации смесей шумов – классы для генерации комбинаций белого и цветного шумов.
5. Вычислительный модуль – функции для оценивания регрессии, вычисления соответствующей неопределенности и коэффициента K , а также получения оценки границ доверительной полосы регрессии при заданной выборке y_i по формулам (2), (3).
6. Программа также содержит ряд примеров оценивания доверительных полос регрессии ([10], файл examples.py), реализована утилита для их визуализации.

Абстрактные классы, подлежащие наследованию, имеют вид:

Листинг 1. Абстрактные классы генератора шума и калькулятор корреляционной матрицы / дисперсии шума

```
class NoiseGenerator(ABC):
    @abstractmethod
    def generate(self, n: int) -> np.ndarray:
        """Генерирует шум заданной длины"""

class CovarianceCalculator(ABC):
    @abstractmethod
```

```
def compute_correlation_matrix(self, n: int) -> np.ndarray:
    """Вычисляет корреляционную матрицу"""

    @abstractmethod
    def get_variance(self) -> float:
        """Возвращает дисперсию шума"""
```

Конкретные реализации генераторов шума и калькуляторов корреляционной матрицы реализованы для всех перечисленных моделей (соответствующие имена классов: <модель_шума>NoiseGenerator, <модель_шума>CovarianceCalculator, где <модель_шума> – одна из AR1, AR2, ColoredNoise, MixedNoise). Помимо указанных классов, реализованы три общих функции: для вычисления стандартной неопределенности $u(x)$, коэффициента K (ее сигнатура приведена в Листинге 2 ниже), и оценивания доверительной полосы регрессии (подробности реализации (листинг) функции опущены; в коде может быть найдена по имени estimate_confidence_band)

Листинг 2. Функция вычисления коэффициента K

```
def compute_k(n0: int, # N0
             n: int, # N
             h: float,
             noise_generator: NoiseGenerator,
             cov_calculator: CovarianceCalculator,
             rho: float = 0.95,
             n_sim: int = 1000000,
             k_low: float = 1.5, # нижняя граница K
             k_high: float = 4., # верхняя граница K
             k_tol: float=1e-4) # точность определения K
    """Вычисляет коэффициент K методом Монте-Карло и бисекции"""
    ...
```

Заметим, что функция принимает абстрактные генератор шума и калькулятор корреляционной матрицы в качестве аргументов, что делает библиотеку легко расширяемой относительно новых моделей шумов. В случае необходимости границы k_low и k_high отрезка, на котором производится поиск значения K методом бисекции, подстраиваются автоматически (пользователь получает соответствующее предупреждение). Массив значений неопределенности будет распечатан, в случае установки флага print_u.

4. Примеры использования программы. Приведем примеры использования программы для получения значений коэффициента K .

Листинг 3. Примеры вычисления коэффициента K

```
n0, n = 20, 50
h = 1

# Пример 1: AR(1) с нормальным белым шумом
phi = 0.7
sigma_w = 1.0
noise_generator_ar1 = AR1NoiseGenerator(phi, sigma_w)
cov_calculator_ar1 = AR1CovarianceCalculator(phi, sigma_w)
k_ar1 = compute_k(n0, n, h, noise_generator_ar1, cov_calculator_ar1)
print(f"K = {k_ar1:.2f}") # результат: K = 2.44

# Пример 2: AR(1) с белым шумом, описываемым симметричным TSP-распределением
for p_tsp in [0.5, 1, 10]: # параметр формы TSP-распределения
    ar1_tsp_noise = AR1NoiseGenerator(phi, sigma_w,
                                       white_noise_func=lambda size: tsp_white_noise(size, p))
    k_ar1_tsp = compute_k(n0, n, h, noise_generator=ar1_tsp_noise,
```

```

cov_calculator=cov_calculator_ar1)
print(f"K = {k_ar1_tsp:.2f}") # результат: K = 2.40, K = 2.41, K = 2.70

# Пример 3: AR(2) с нормальным белым шумом
phi1, phi2 = 0.5, -0.3
noise_generator_ar2 = AR2NoiseGenerator(phi1, phi2, sigma_w)
cov_calculator_ar2 = AR2CovarianceCalculator(phi1, phi2, sigma_w)
k_ar2 = compute_k(n0, n, h, noise_generator_ar2, cov_calculator_ar2)
print(f"K = {k_ar2:.2f}") # результат: K = 2.44

# Пример 4: Цветной шум (1/f^a)
colored_noise_gen = ColoredNoiseGenerator(
    a=1., scale=1., fs=1., target_std=1., f_min=0.01)
colored_noise_cov = ColoredNoiseCovarianceCalculator(
    a=1., scale=1., fs=1., target_std=1., f_min=0.01, simulation_length=100000)
k_colored = compute_k(n0, n, h, noise_generator=colored_noise_gen,
    cov_calculator=colored_noise_cov)
print(f"K = {k_colored:.2f}") # результат: K = 2.95

# Пример 5: Комбинация белого и цветного шума
white_weight = 0.7
white_sigma = 1.0
colored_a = 1.0 # 1/f

mixed_noise_gen = MixedNoiseGenerator(white_weight, white_sigma, colored_a,
    colored_scale=1., colored_fs=1., colored_target_std=1., colored_f_min=0.01)
mixed_noise_cov = MixedNoiseCovarianceCalculator(white_weight, white_sigma,
    colored_a, colored_scale=1., colored_fs=1., colored_target_std=1.,
    colored_f_min=0.01, simulation_length=100000)
k_mixed = compute_k(n0, n, h,
    noise_generator=mixed_noise_gen, cov_calculator=mixed_noise_cov)
print(f"K = {k_mixed:.2f}") # результат: K = 2.83

```

Значения K , вычисленные с помощью программы, содержатся в комментариях («результат: $K = \dots$ »). Программа также дает возможность при генерации белого шума использовать законы распределений, отличные от нормального: в Примере 2 Листинга 3 белый шум описывается симметричным двусторонним степенным распределением [11] (такой пример для шума, описываемого моделью AR(1), рассматривался ранее в работе [4]).

По ссылке [10] также можно найти примеры вычисления доверительных полос регрессии. Приведем один из них ниже (заметим, что вид полосы в каждом конкретном примере, очевидно, зависит от конкретных значений y_i).

Пример: пусть $n_0 = 20$, $n = 100$, шум описывается моделью AR(1) с $\varphi = 0.5$, белый шум имеет симметричное двустороннее степенное распределение с параметром степени $p = 10$ и единичной дисперсией. Тогда $K \approx 2.47$, и доверительная полоса для $P_0 = 0.95$ и конкретной реализации y_i приведена ниже на рисунке 1.

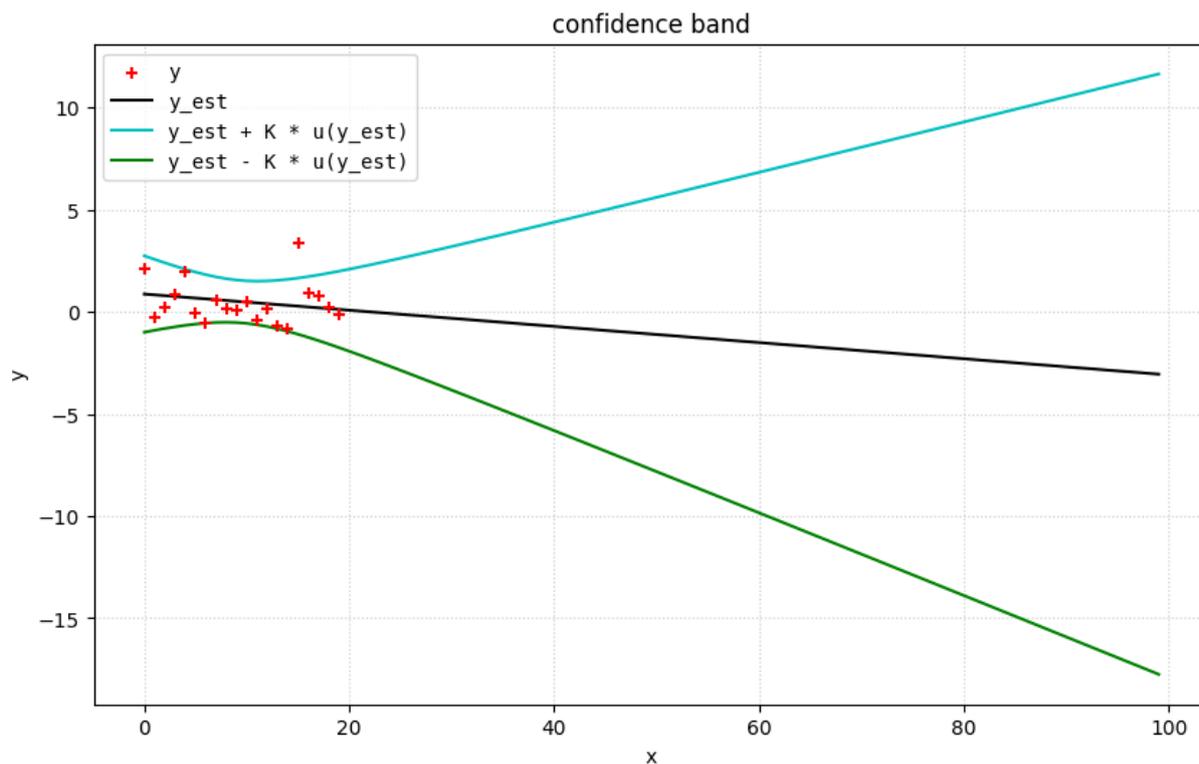


Рис. 1. Пример построенной доверительной полосы регрессии

Заметим, что использование библиотеки NumPy при написании Программы дает следующие преимущества: векторизация операций и использование оптимизированных матричных операций. Современные реализации NumPy, как правило, используют оптимизированные библиотеки линейной алгебры, которые, в свою очередь, автоматически распараллеливают выполнение многих стандартных операций (таких, как умножение матриц, разложения, некоторые поэлементные операции) на уровне потоков без прямого участия пользователя. Это позволяет добиться высокой производительности вычислений, сравнимой с таковой для низкоуровневых языков программирования.

Заключение. В статье рассмотрена задача оценивания доверительных полос линейной регрессии в условиях коррелированных ошибок, описываемых AR-моделями и цветными шумами. Стандартные методы построения доверительных интервалов, основанные на предположении о независимости и нормальности ошибок, приводят к систематическому занижению ширины полос и, как следствие, к недооценке реальной неопределенности прогноза. Установлена принципиальная необходимость применения численных методов, в частности, метода Монте-Карло в сочетании с алгоритмом бисекции, для определения коэффициента охвата K , обеспечивающего заданный уровень доверия, аналитическое вычисление которого невозможно из-за сложной зависимости от структуры ковариационной матрицы, параметров регрессии и объема данных. Предложенный алгоритм позволяет находить значения K с заданной точностью при разумных вычислительных затратах. Заметим, что критически важное значение при этом имеет корректный выбор модели шума, определяемый природой исследуемых данных.

Разработанное программное обеспечение допускает дальнейшее расширение и адаптацию к конкретным прикладным задачам. Архитектура позволяет относительно просто добавлять как новые типы корреляционных структур, так и распределения ошибок. В качестве одного из дальнейших направлений для разработки можно упомянуть расширение на нелинейные регрессионные модели.

Список источников

1. Lee J., Lund R. Revisiting Simple Linear Regression with Autocorrelated Errors. *Biometrika*, 2004, 91(1), pp. 240–245, DOI:10.1093/biomet/91.1.240.
2. Safi S., White A. The Efficiency of OLS in the presence of auto-correlated disturbances in regression models. *Journal of Modern Applied Statistical Methods*, 2006, 5(1), pp. 133-143, DOI:10.22237/jmasm/1146456540.
3. Степанов А.В. Оценивание доверительных полос для линейной регрессии при коррелированных входных данных / Степанов А.В., Чуновкина А.Г. // Измерительная техника, 2019. – № 5. – С. 14-21.
4. Chunovkina A.G., Stepanov A.V., Wimmer G. On estimation of linear regression confidence bands: Analytical solution and Monte Carlo simulation. *Advanced Mathematical and Computational Tools in Metrology and Testing XII*, 2022, pp. 188-196.
5. Staudenmayer J. Buonaccorsi J. Measurement Error in Linear Autoregressive Models. *Journal of the American Statistical Association*, 2005, 100(471), pp. 841-852, DOI:10.1198/016214504000001871 .
6. Cox M., Harris P. Statistical error modelling. NPL Report (2004) CMSC 45/04.
7. Allan D. Should the classical variance be used as a basic measure in standards metrology? *IEEE Transactions on Instrumentation and Measurement*, 1987, IM-36(2), pp. 646-654, DOI: 10.1109/TIM.1987.6312761.
8. Vernotte F., Lantz E. Metrology and 1/f noise: Linear regressions and confidence intervals in flicker noise context. *Metrologia*, 2015, 52(2), pp. 222-237, DOI: 10.48550/arXiv.1407.7760
9. Stepanov A., Chunovkina A., Katkov A. Evaluation of Zener Standard Drifts. *Conference on Precision Electromagnetic Measurements (CPEM 2018)*, Paris, France, 2018, pp. 1-2.
10. Код приложения. – URL: <https://github.com/stepanov17/imt-2025> (дата обращения: 10.12.2025).
11. Kotz S, Van Dorp J.R. Beyond beta: other continuous families of distributions with bounded support and applications. *World Scientific Publishing*, Singapore, 2004, DOI:10.1142/9789812701282.

Степанов Александр Владимирович. К.ф.-м.н., ведущий научный сотрудник лаборатории теоретической метрологии, Всероссийский научно-исследовательский институт метрологии им. Д.И.Менделеева (ВНИИМ). AuthorID: 875816, SPIN: 9570-8972, ORCID: 0000-0002-5917-1037, stepanov17@yandex.ru. 190005, Россия, Санкт-Петербург, Московский пр., 19.

UDC 519.233.5

DOI:10.25729/ESI.2026.41.1.013

On the software implementation of the Monte Carlo method for estimating confidence bands of linear regression

Aleksandr V. Stepanov

D.I. Mendeleev Institute for Metrology (VNIIM), Russia, St. Petersburg, stepanov17@yandex.ru

Abstract. The paper addresses the problem of calculating uncertainty bands for linear regression with correlated input data. To estimate confidence bands, the generalized least squares method is applied. To determine their boundaries, a coverage factor is introduced; when multiplied by the standard uncertainty of the regression at specific points, this factor yields the required limits. The relevance of this research stems from the fact that standard methods for constructing confidence intervals, based on the assumption of independent errors, lead to a systematic underestimation of the uncertainty band width in the presence of autocorrelation. This, in turn, creates a false impression of forecast accuracy and may result in erroneous statistical conclusions. To construct confidence bands correctly, the structure of the temporal dependence of errors must be taken into account. This study considers the following models of correlated noise: autoregressive processes with exponential correlation decay, and colored noise characterized by power-law decay and long-term memory.

Unlike the classical case of independent errors, where the coverage factor corresponds to a quantile of the normal distribution, no analytical expression exists for this factor in the presence of correlation. The value of the factor directly depends on the structure of the error covariance matrix, the training sample size, and the forecasting horizon. To determine it, the paper employs a numerical Monte Carlo method combined with an iterative bisection procedure, which allows finding the coverage factor with a specified accuracy.

Specialized software has been developed in Python using the NumPy and SciPy libraries. The software implementation solves the problem of estimating hyperbolic-shaped linear regression bands when the error correlation structure is described by the aforementioned models. Corresponding examples for estimating the coverage factor and regression bands are provided, along with a link to the software implementation. The modular architecture of the developed program allows for expansion to other types of correlation structures. The applicability of the work is due to the need for correct uncertainty estimation in the statistical processing of experimental data obtained during the solution of measurement problems.

Keywords: linear regression, confidence bands, generalized least squares, correlated noise, colored noise, Monte Carlo method, numerical methods

References

1. Lee J., Lund R. Revisiting Simple Linear Regression with Autocorrelated Errors. *Biometrika*, 2004, 91(1), pp. 240–245, DOI:10.1093/biomet/91.1.240.
2. Safi S., White A. The Efficiency of OLS in the presence of auto-correlated disturbances in regression models. *Journal of Modern Applied Statistical Methods*, 2006, 5(1), pp. 133-143, DOI:10.22237/jmasm/1146456540.
3. Stepanov A.V., Chunovkina A.G. Ocenivanie doveritel'nyh polos dlya linejnoj regressii pri korrelirovannyh vhodnyh dannyh [Estimation of linear regression bands for correlated input data]. *Izmeritel'naya Tekhnika [Measurement Techniques]*, 2019, № 5, pp. 14-21.
4. Chunovkina A.G., Stepanov A.V., Wimmer G. On estimation of linear regression confidence bands: Analytical solution and Monte Carlo simulation. *Advanced Mathematical and Computational Tools in Metrology and Testing XII*, 2022, pp. 188-196.
5. Staudenmayer J. Buonaccorsi J. Measurement Error in Linear Autoregressive Models. *Journal of the American Statistical Association*, 2005, 100(471), pp. 841-852, DOI:10.1198/016214504000001871 .
6. Cox M., Harris P. Statistical error modelling. NPL Report (2004) CMSC 45/04.
7. Allan D. Should the classical variance be used as a basic measure in standards metrology? *IEEE Transactions on Instrumentation and Measurement*, 1987, IM-36(2), pp. 646-654, DOI: 10.1109/TIM.1987.6312761.
8. Vernotte F., Lantz E. Metrology and 1/f noise: Linear regressions and confidence intervals in flicker noise context. *Metrologia*, 2015, 52(2), pp. 222-237, DOI: 10.48550/arXiv.1407.7760
9. Stepanov A., Chunovkina A., Katkov A. Evaluation of Zener Standard Drifts. *Conference on Precision Electromagnetic Measurements (CPEM 2018)*, Paris, France, 2018, pp. 1-2.
10. The application code. Available at <https://github.com/stepanov17/imt-2025> (accessed: 12/10/2025).
11. Kotz S, Van Dorp J.R. *Beyond Beta: Other Continuous Families of Distributions with Bounded Support and Applications*. World Scientific Publishing, Singapore, 2004, DOI:10.1142/9789812701282.

Stepanov Aleksandr Vladimirovich. Candidate of Physical and Mathematical Sciences, Researcher at the Laboratory of Theoretical Metrology, D.I. Mendeleev Institute for Metrology (VNIIM). AuthorID: 875816, SPIN: 9570-8972, ORCID: 0000-0002-5917-1037, stepanov17@yandex.ru. 190005, Russia, St. Petersburg, Moskovsky pr., 19.

Статья поступила в редакцию 01.11.2025; одобрена после рецензирования 09.12.2025; принята к публикации 10.02.2026.

The article was submitted 11/01/2025; approved after reviewing 12/09/2025; accepted for publication 02/10/2026.