Математические методы исследования
Mathematical methods of investigation

DOI: https://doi.org/10.26896/1028-6861-2023-89-5-71-80

РЕГРЕССИОННЫЙ АНАЛИЗ ДАННЫХ НА ОСНОВЕ МЕТОДА НАИМЕНЬШИХ МОДУЛЕЙ В ДИНАМИЧЕСКИХ ЗАДАЧАХ ОЦЕНИВАНИЯ

© **Олег Александрович Голованов[1,2], Александр Николаевич Тырсин[1,3*]**

[1] Уральский федеральный университет имени первого Президента России Б. Н. Ельцина, Россия, 620002, г. Екатеринбург, ул. Мира, д. 19; *e-mail: at2001@yandex.ru
[2] Институт экономики Уральского отделения РАН, Россия, 620014, г. Екатеринбург, ул. Московская, д. 29.
[3] Научно-инженерный центр «Надежность и ресурс больших систем и машин» Уральского отделения РАН, Россия, 620049, г. Екатеринбург, ул. Студенческая, д. 54а.

Статья поступила 10 октября 2022 г. Поступила после доработки 22 ноября 2022 г. Принята к публикации 28 декабря 2022 г.

Применение регрессионного анализа в динамических задачах оценивания систем требует от алгоритма высокого быстродействия определения параметров модели. Также исходные данные могут иметь стохастическую неоднородность. Поэтому наряду с быстродействием необходимо, чтобы оценки параметров модели были устойчивыми к различным аномалиям в данных. Однако устойчивые методы оценивания, включая метод наименьших модулей, значительно уступают параметрическим методам. Цель работы — описание вычислительно эффективного алгоритма реализации метода наименьших модулей для динамического оценивания регрессионных моделей и исследование его возможностей для решения практических задач. Этот алгоритм основан на спуске по узловым прямым. При этом вместо значений целевой функции рассматривают ее производную по направлению спуска. Вычислительная трудоемкость алгоритма снижена также за счет использования в качестве начальной точки решения задачи на предыдущем шаге и эффективного обновления наблюдений в текущей выборке данных. Проведен сравнительный анализ фактического быстродействия предложенного динамического варианта алгоритма градиентного спуска по узловым прямым со статическим вариантом, а также с методом наименьших квадратов. Показано, что динамический вариант алгоритма градиентного спуска по узловым прямым позволил для распространенных практических ситуаций приблизиться по быстродействию к методу наименьших квадратов. Это позволяет использовать предложенный вариант алгоритма градиентного спуска по узловым прямым на практике в динамических задачах оценивания широкого класса систем.

Ключевые слова: метод наименьших модулей; линейная регрессия; динамика; алгоритм; узловая прямая; вычислительная эффективность.

REGRESSION ANALYSIS OF DATA BASED ON THE METHOD OF LEAST ABSOLUTE DEVIATIONS IN DYNAMIC ESTIMATION PROBLEMS

© **Oleg A. Golovanov[1,2], Aleksandr N. Tyrsin[1,3*]**

[1] The first President of Russia B. N. Yeltsin Ural Federal University, 19, ul. Mira, Yekaterinburg, 620002, Russia; *e-mail: at2001@yandex.ru
[2] Institute of Economics, Ural Branch of RAS, 29, Moskovskaya ul., Yekaterinburg, 620014, Russia.
[3] Science and Engineering Center “Reliability and Resource of Large Systems and Machines”, Ural Branch of RAS, 54a, Studencheskaya ul., Yekaterinburg, 620049, Russia.

Received October 10, 2022. Revised November 22, 2022. Accepted December 28, 2022.

The use of regression analysis in dynamic problems of system estimation requires a high-speed algorithm of model parameter determination. Moreover, the original data may have stochastic heterogeneity which entails the necessity of the estimates of model parameters be resistant to various data anomalies. However, stable estimation methods, including the least absolute deviations method, are significantly inferior to the parametric ones. The goal of the study is to describe a computationally efficient algorithm for imple-

menting the method of least absolute deviations for dynamic estimation of regression models and to study its capabilities for solving practical problems. This algorithm is based on descending along nodal lines. In this case, instead of the values of the objective function, its derivative in the direction of descent is considered. The computational complexity of the algorithm is also reduced due to the use of the solution of the problem at the previous step as a starting point and efficient updating of observations in the current data sample. The external performance of the proposed dynamic version of the algorithm of gradient descent along nodal lines has been compared with the static version and with the least squares method. It is shown that the dynamic version of the algorithm of gradient descent along the nodal lines make it possible to bring the speed close to that of the least squares method for common practical situations and to use the proposed version in dynamic estimation problems for a wide class of systems.

Keywords: least absolute deviations method; linear regression; dynamics; algorithm; nodal straight line; computational efficiency.

Введение

Наиболее часто при оценивании параметров $\alpha_1, ..., \alpha_m$ регрессионных моделей

$$Y = \alpha_1 + \alpha_2 X_2 + \alpha_3 X_3 + ... + \alpha_m X_m + \varepsilon \qquad (1)$$

используют один из базовых математических методов, основанных на минимизации суммы квадратов отклонений [1 – 3]; здесь Y — зависимая переменная; X_k — независимые переменные; $\boldsymbol{\alpha}$ — вектор неизвестных параметров; ε — случайная компонента.

Метод наименьших квадратов (МНК) отличается простотой реализации и быстродействием, однако его работа требует выполнения ряда предпосылок [4], нарушение которых приводит к значительным отклонениям оценок параметров модели от истинных значений. В этом случае часто применяют устойчивые методы, основанные на минимизации суммы модулей отклонений [5 – 7]. Задача оценивания параметров линейной регрессионной модели методом наименьших модулей (МНМ) имеет следующий вид:

$$Q(\mathbf{a}) = \sum_{i=1}^{n} \left| y_i - \sum_{j=1}^{m} a_j x_{ij} \right| \to \min_{a \in R^m}, \qquad (2)$$

где

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}, \mathbf{X} = \{x_{ij}\}_{n \times m} = \begin{pmatrix} 1 & x_{12} & \dots & x_{1m} \\ 1 & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ 1 & x_{n2} & \dots & x_{nm} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \dots \\ \mathbf{x}_n \end{pmatrix},$$

$\mathbf{a}$ — вектор оценок неизвестных параметров.

В настоящее время предложен ряд алгоритмов для решения задачи (2) [8 – 13]. Однако они существенно уступают в быстродействии МНК, что ограничивает их применение в динамических задачах оценивания регрессионных зависимостей. Можно отметить две статьи, в которых рассматривалась данная задача.

В [14] предложен подход к решению задачи динамического оценивания коэффициентов регрессии — метод весовой и временной рекурсий, сочетающий в себе идеи вариационно-взвешенных квадратических приближений и сглаживающего фильтра Калмана. Однако он носит приближенный характер и для приемлемой точности является вычислительно трудоемким.

Другой подход [15] — использование в динамике градиентного спуска по узловым прямым. На первом шаге рассматривается начальная выборка данных $V^{(1)} = (\mathbf{x}_i, y_i)$, $i = 1, ..., n$. Для этой выборки решается задача (2) и находится решение $\mathbf{a}^{(1)}$.

На втором шаге вместо первого наблюдения $(\mathbf{x}_1, y_1)$ в выборку включаем новое наблюдение $(\mathbf{x}_{n+1}, y_{n+1})$ и для выборки $\mathbf{V}^{(2)} = (\mathbf{x}_i, y_i)$, $i = 2, ..., n + 1$ снова определяем МНМ-параметры $\mathbf{a}^{(2)}$ уравнения регрессии (1) и т.д. В результате получаем последовательность решений $\mathbf{a}^{(1)}$, $\mathbf{a}^{(2)}$, ..., мониторинг которых позволяет контролировать состояние исследуемой системы.

Однако в данном алгоритме не были исследованы возможности алгоритма для решения практических задач динамического регрессионного оценивания, а также вопросы эффективного обновления наблюдений в текущей выборке данных. Кроме того, как оказалось, его вычислительное быстродействие может быть увеличено.

Цель статьи — описание более быстрого алгоритма реализации МНМ на основе спуска по узловым прямым для динамического оценивания регрессионных моделей и исследование его возможностей для решения практических задач.

Методы реализации

Для решения задачи (2) введем множество гиперплоскостей $\boldsymbol{\Omega}$: $\{\Omega_1, \Omega_2, ..., \Omega_n\}$, где каждая гиперплоскость будет являться наблюдением, представленным уравнением вида

$$y_i - \langle \mathbf{a}, \mathbf{x}_i \rangle = 0, \ (i = 1, 2, ..., n). \qquad (3)$$

Пересечение $(m - 1)$ независимых гиперплоскостей (3) назовем узловой прямой

$$l_{(k_1,...,k_{m-1})} : \bigcap \Omega_i, \ i \in \{k_1, ..., k_{m-1}\},$$
$$k_l \in \{1, 2, ..., n\}. \qquad (4)$$

В таком случае узловой точкой будем считать пересечение m гиперплоскостей (3) или гиперплоскости и узловой прямой (4):

$$\mathbf{u} = \bigcap_{s \in M} \Omega_s,\ M = \{k_1, ..., k_m\},$$
$$k_1 < k_2 < ... < k_m,\ k_l \in \{1, 2, ..., n\}. \qquad (5)$$

Обозначим U как множество всех узловых точек (5).

Алгоритм поиска минимума основан на выпуклости целевой функции (2). Эффективный метод решения задачи — спуск по узловым прямым [13]. Вследствие конечности выборки, а значит, конечности числа гиперплоскостей, решение задачи (2) будет достигнуто за конечное число итераций [16]. В качестве первого приближения берем некоторую случайную узловую точку $\mathbf{u}^{(0)}$, в последующем выступающую в качестве временного «минимума» функции. Она представляет собой пересечение m гиперплоскостей, коэффициенты которых находятся путем решения системы линейных алгебраических уравнений (СЛАУ): $y_i - \langle \mathbf{u}^{(0)}, \mathbf{x}_i \rangle = 0$, $(i = 1, 2, ..., m)$. Далее, исключив одно наблюдение, получаем узловую прямую $l_{(k_1,...,k_{m-1})}$, на которой будем искать точку с меньшим значением целевой функции. Расширенная матрица СЛАУ, соответствующая узловой прямой, будет выглядеть следующим образом:

$$\mathbf{A}_{l_{(k_1,...,k_{m-1})}} = \begin{pmatrix} 1 & x_{k_1,2} & x_{k_1,3} & \dots & x_{k_1,m-1} & x_{k_1,m} & y_{k_1} \\ 1 & x_{k_2,2} & x_{k_2,3} & \dots & x_{k_2,m-1} & x_{k_2,m} & y_{k_2} \\ 1 & x_{k_3,2} & x_{k_3,3} & \dots & x_{k_3,m-1} & x_{k_3,m} & y_{k_3} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_{k_{m-2},2} & x_{k_{m-2},3} & \dots & x_{k_{m-2},m-1} & x_{k_{m-2},m} & y_{k_{m-2}} \\ 1 & x_{k_{m-1},2} & x_{k_{m-1},3} & \dots & x_{k_{m-1},m-1} & x_{k_{m-1},m} & y_{k_{m-1}} \end{pmatrix}.$$

Чтобы сократить число вычислений, необходимых для нахождения узловых точек на прямой, используем метод Жордана – Гаусса [17]. Преобразуем матрицу к виду

$$\mathbf{A}_{l_{(k_1,...,k_{m-1})}} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & x'_{k_1,m} & y'_{k_1} \\ 0 & 1 & 0 & \dots & 0 & x'_{k_2,m} & y'_{k_2} \\ 0 & 0 & 1 & \dots & 0 & x'_{k_3,m} & y'_{k_3} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & x'_{k_{m-2},m} & y'_{k_{m-2}} \\ 0 & 0 & 0 & \dots & 1 & x'_{k_{m-1},m} & y'_{k_{m-1}} \end{pmatrix}.$$

При помощи подобного преобразования можно значительно сократить вычислительные затраты на нахождение узловых точек на прямой за счет отсутствия необходимости повторять однообразные вычисления. Таким образом, для нахождения каждой последующей точки останется добавить уравнение, соответствующее m-й гиперплоскости, и произвести остаточные преобразования:

$$\mathbf{A}_{u_{(k_1,...,k_{m-1},i)}} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & x'_{k_1,m} & y'_{k_1} \\ 0 & 1 & 0 & \dots & 0 & x'_{k_2,m} & y'_{k_2} \\ 0 & 0 & 1 & \dots & 0 & x'_{k_3,m} & y'_{k_3} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & x'_{k_{m-1},m} & y'_{k_{m-1}} \\ 1 & x_{i,2} & x_{i,3} & \dots & x_{i,m-1} & x_{i,m} & y_i \end{pmatrix},$$

где $k_1 < k_2 < ... < k_{m-1}$, $i \in \{1, 2, ..., n\}$, $i \notin \{k_1, k_2, ..., k_{m-1}\}$.

Далее, варьируя номер i в преобразованной расширенной матрице, находим оставшиеся $(n - m)$ лежащие на ней узловые точки и упорядочиваем их по направлению прямой в соответствии с последним коэффициентом. После этого необходимо найти точку, в которой целевая

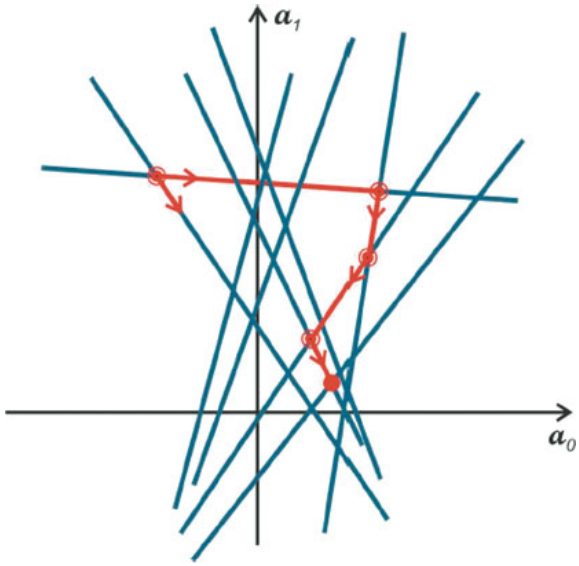

Рис. 1. Принцип поиска решения путем спуска по узловым прямым

Fig. 1. The principle of finding a solution by descending along the nodal straight lines

функция (2) примет наименьшее значение на всей узловой прямой, и аналогичным образом проверить оставшиеся прилежащие к анализируемой точке узловые прямые. Линейность выражений под знаком модуля в (2) приводит к достаточно простому виду производной функции $Q(\mathbf{a})$ по направлению узловой прямой. Вычисляем в узловой точке $\mathbf{u}^{(0)}$ значение производной по направлению узловой прямой $l_{(k_1,\ldots,k_{m-1})}$. В силу выпуклости целевой функции (2) двигаемся в сторону убывания производной. В результате достигнем узловой точки, в которой производная не существует, именно в ней и будет находиться минимум текущей узловой прямой. Поэтому находим производную в очередной узловой точке $\mathbf{u}^{(*)} = (u_1^{(*)}, u_2^{(*)}, \ldots, u_m^{(*)})$ с двух сторон. Каждая из производных равна сумме слагаемых по числу наблюдений:

$$\frac{\partial Q(\mathbf{u}^{(*)})}{\partial \mathbf{l}_{(k_1,\ldots,k_{m-1})}} = \sum_{i=1}^{n} (c_1 + x_{i2}c_2 + \ldots + x_{im}c_m) \times$$

$$\times \operatorname{sign}\left(\sum_{j=1}^{m} u_j^{(*)} x_{ij} - y_i \right),$$

где $\mathbf{l}_{(k_1,\ldots,k_{m-1})} = (c_1, c_2, \ldots, c_m)$ — направляющий вектор узловой прямой $l_{(k_1,\ldots,k_{m-1})}$.

Если знак производной по направлению слева $\left.\frac{\partial Q(\mathbf{u}^{(*)})}{\partial \mathbf{l}_{(k_1,\ldots,k_{m-1})}}\right|_{\mathbf{u}_-^{(*)}}$ и справа $\left.\frac{\partial Q(\mathbf{u}^{(*)})}{\partial \mathbf{l}_{(k_1,\ldots,k_{m-1})}}\right|_{\mathbf{u}_+^{(*)}}$ меняет знак с отрицательного на положительный, то эту точку можно считать точкой разрыва и она является тем местом, где целевая функция примет свое минимальное значение на всей узловой прямой. Фиксируем новый «минимум» функции вместо $\mathbf{u}(0)$ и продолжаем спуск. Спуск производим до тех пор, пока не будет найдена точка, из которой дальнейший спуск невозможен. Эта точка и будет считаться точным решением задачи (2). На рис. 1 показан принцип поиска решения при помощи алгоритма покоординатного спуска.

Рассмотрим динамическую реализацию алгоритма. Пусть мы нашли решение задачи $\mathbf{a}^{(*)}$ по начальной выборке данных $(\mathbf{X}_i, y_i) = (x_{i1}, \ldots, x_{im}, y_i)$. Решение $\mathbf{a}^{(*)}$ является пересечением m гиперплоскостей (обозначим их как множество Ω^*), образованных m наблюдениями (обозначим их как множество Z^*).

Динамический анализ данных предполагает постоянное добавление новых наблюдений, что приводит к бесконтрольному росту исследуемой выборки. Чтобы избежать этого, ограничим выборку числом наблюдений n, рассмотренных в первой статической итерации алгоритма, и заменим одно из «старых» наблюдений на «новое». Обозначим старое и новое наблюдения как $(\mathbf{X}^*, y^*)$ и $(\mathbf{X}_1, y_1)$ соответственно.

В табл. 1 приведены способы замены наблюдения, которые исследованы в процессе работы. Способы замены, соответствующие столбцам, характеризуются возможностью замены наблюдений, формировавших решение на предыдущей итерации алгоритма. Таким образом, при попытке заменить подобное наблюдение выбираем либо следующее случайное наблюдение, либо следующее наблюдение согласно их порядковому номеру. Порядковый номер присваивается каждому наблюдению согласно порядку его попадания в выборку в процессе генерации или считывания из файла.

Предположим, что выбран способ замены из первого столбца, без возможности замены наблюдения из Z^*. Таким образом, решение, полученное при проведении предыдущей итерации алго-

Таблица 1. Способы замены «старого» наблюдения на новое в динамической реализации алгоритма

Table 1. Methods for replacing the “old” observation with a new one in the dynamic implementation of the algorithm

Подвыборки	Замена наблюдения из Z^* невозможна	Возможна замена наблюдения из Z^*
Случайный выбор наблюдения	D_{11}	D_{12}
Выбор согласно порядковому номеру	D_{21}	D_{22}

ритма, не было заменено. Считаем его начальной точкой (присваиваем $\mathbf{a}^{(0)} = \mathbf{a}^{(*)}$) и продолжаем спуск к новому решению. В ином случае при выборе способа с возможностью замены наблюдения из множества Z^* получаем узловую прямую $l_{(k_1^*,\ldots k_{m-1}^*)}$, сформированную пересечением оставшихся $(m-1)$ гиперплоскостей из множества Ω^*. Пересечение полученной узловой прямой и гиперплоскости Ω_1, сформированной новым наблюдением $(\mathbf{X}_1, y_1)$, полагаем начальной точкой $\mathbf{a}^{(0)} = l_{(k_1^*,\ldots,k_{m-1}^*)} \cap \Omega_1$ и проводим поиск решения при помощи алгоритма градиентного спуска.

Рассмотрим конкретный пример (рис. 2) при $m = 2$ и $n = 4$. На первой итерации алгоритма начальную узловую точку выбираем случайным образом, пусть это будет точка, соответствующая пересечению узловых прямых *2* и *3*. При поиске точки для перехода рассматриваем все прилегающие узловые прямые, на которых вычисляем значения целевой функции. После нахождения точки с минимальным для всех прилегающих прямых значением целевой функции проводим переход; в данном случае такой точкой будет пересечение прямых *2* и *4*. По тому же принципу проводим спуск к решению текущей регрессионной задачи — в точку пересечения прямых *1* и *4*.

Пусть выбран способ замены D_{i1} (рис. 3, *а*). В этом случае невозможно провести замену наблюдений, соответствующих прямым *1* и *4*, так как они формируют решение на предыдущей итерации. Тогда проведем замену «старого» наблюдения *2*, на новое *5**. Таким образом, сохранив предыдущее решение и обозначив его как начальную точку, спускаемся в новый минимум в точке *1*, *5**.

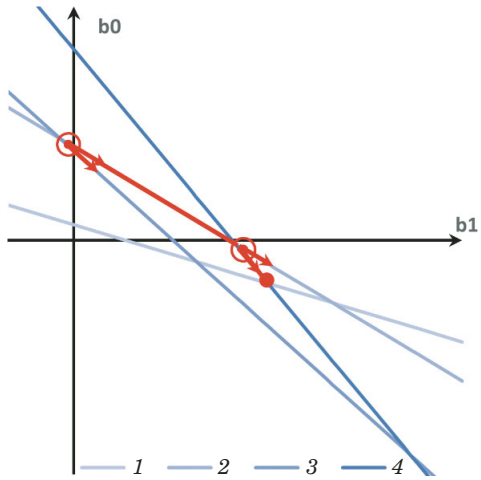

Рис. 2. Первая итерация поиска решения ($m = 2, n = 4$)

Fig. 2. First iteration of finding a solution ($m = 2, n = 4$)

Иначе был выбран способ D_{i2} (рис. 3, *б*), при котором возможна замена любого наблюдения из анализируемой выборки. Проведем замену наблюдения из Z^* под номером *1* на *6**, тогда новой начальной точкой будет место пересечения оставшейся и новой прямых *4*, *6**. Спускаясь из этой точки, достигаем минимума для текущей выборки в точке *2*, *6**.

Необходимо определить, какой из описанных ранее способов замены является более выгодным при сравнении их вычислительной сложности. Характерный показатель в данном случае — число осуществленных переходов по узловым прямым до достижения решения задачи на каждой из итераций. В первую очередь необходимо определить, существует ли какая-либо значимая разница между числом переходов, а если она отсут-

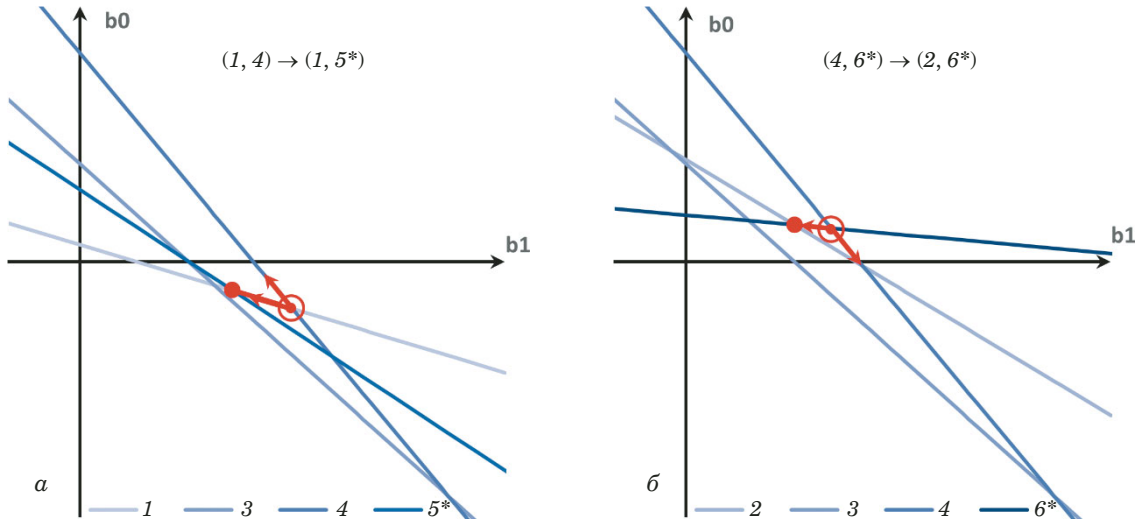

Рис. 3. Способы замены наблюдения: *а* — не участвовавшего в формировании решения; *б* — формировавшего решение на предыдущей итерации

Fig. 3. Ways to replace an observation an observation: *a* — not participating in the formation of the solution; *b* — forming the solution at the previous iteration

ствует, выбор следует делать в зависимости от сложности реализации способа замены.

Для решения поставленной задачи исследуем число переходов на однородность при помощи критерия Стьюдента для двух независимых выборок. Действительно, полученные выборки для четырех способов замены наблюдения можно считать независимыми, так как наборы показателей в группах формируются вне зависимости от этого процесса в других группах, т.е. число переходов между узловыми прямыми для одного способа никаким образом не влияет на число переходов для другого. Воспользуемся методом Монте-Карло [18] и проведем генерацию 10 000 наблюдений, выделив 300 из них для проведения первой статической итерации. Возьмем средние для каждых 100 испытаний оставшихся динамических итераций, тем самым получив 97 нормализованных наблюдений при $m = 2, 3, ..., 8$ для каждой выборки. Поскольку мы исследуем устойчивый к выбросам метод, то необходимо проверить его в условиях стохастической неоднородности данных. Для генерации загрязненной выборки используем модель Тьюки – Хьюбера [19, 20]

$$F_\gamma(x) = (1 - \gamma)F(x) + \gamma f F_H(x), \qquad (6)$$

где $F(x)$ — функция распределения случайных ошибок, обладающая необходимыми «хорошими» признаками; $F_H(x)$ — функция распределения засорений; γ — вероятность засорения.

Проведем генерацию одной «чистой» выборки ($\gamma = 0$) и трех выборок с псевдослучайными загрязнениями, обладающими различными свойствами, взяв в качестве $F(x)$ распределение Гаусса, а в качестве засорений используем распределения со следующими плотностями вероятности:

$$f_H(x) = \frac{1}{\sqrt{2\pi}\sigma_H}\exp\left[-\frac{(x - a_H)^2}{2\sigma_H^2}\right],$$

$$a_H = 2,\ \sigma_H = 3,\ \gamma = 0{,}1,$$

$$f_H(x) = \frac{1}{\pi}\frac{\gamma_H}{(x - a_H)^2 + \gamma_H^2},\ a_H = 0,\ \gamma_H = 1,\ \gamma = 0{,}1,$$

$$f_H(x) = \begin{cases} 0, x < a_H, \\ \dfrac{2}{\pi}\dfrac{\gamma_H}{(x - a_H)^2 + \sigma_H^2}, x \geq a_H, \end{cases}$$

$$a_H = 0,\ \gamma_H = 1,\ \gamma = 0{,}1.$$

Таким образом, рассмотрев несколько вариантов генерации данных, можно определить наиболее выгодный способ замены наблюдений для проведения анализа в динамике вне зависимости от вида распределения выборки.

Обсуждение результатов

В табл. 2 приведены средние значения t-статистики Стьюдента при $m = 2, 3, ..., 7$, для четырех способов замены наблюдений D и четырех вариантов генерации данных. Видно, что наиболее отличающимися способами замены являются D_{11} и D_{12}. Это можно объяснить случайным характером выбора наблюдения из выборки, вследствие которого возможна замена «хорошего» наблюдения, гиперплоскость которого находится вблизи от решения в предыдущей итерации, на «плохое», а также обратная ситуация во втором сравниваемом способе. Подобное отклонение усугубляется в условиях загрязненной выборки. Однако все расчетные значения t-статистики

Таблица 2. Средние значения t-статистики для независимых выборок при $m = 2, 3, ..., 7$
Table 2. Average values of t-statistics for independent samples at $m = 2, 3, ..., 7$

D_{ij}	$m = 2$				$m = 3$				$m = 4$			
	D_{11}	D_{12}	D_{21}	D_{22}	D_{11}	D_{12}	D_{21}	D_{22}	D_{11}	D_{12}	D_{21}	D_{22}
D_{11}	0,00	0,55	0,69	0,82	0,00	1,09	0,97	0,96	0,00	1,36	0,83	1,04
D_{12}	0,55	0,00	0,23	0,39	1,09	0,00	1,47	1,14	1,36	0,00	0,78	1,06
D_{21}	0,69	0,23	0,00	0,60	0,97	1,47	0,00	0,51	0,83	0,78	0,00	0,44
D_{22}	0,82	0,39	0,60	0,00	0,96	1,14	0,51	0,00	1,04	1,06	0,44	0,00
D_{ij}	$m = 5$				$m = 6$				$m = 7$			
	D_{11}	D_{12}	D_{21}	D_{22}	D_{11}	D_{12}	D_{21}	D_{22}	D_{11}	D_{12}	D_{21}	D_{22}
D_{11}	0,00	1,33	0,39	1,20	0,00	1,04	0,84	0,86	0,00	1,56	0,67	1,31
D_{12}	1,33	0,00	1,07	0,62	1,04	0,00	0,72	0,92	1,56	0,00	1,80	0,90
D_{21}	0,39	1,07	0,00	0,93	0,84	0,72	0,00	1,06	0,67	1,80	0,00	1,28
D_{22}	1,20	0,62	0,93	0,00	0,86	0,92	1,06	0,00	1,31	0,90	1,28	0,00

оказались меньше критического $t_{кр} = 1{,}97$ при $(n_1 + n_2 - 2) = 192$ степенях свободы и уровне значимости $p = 0{,}05$.

Рассмотрим более подробно каждый отдельный случай путем выражения процента числа раз, когда t-статистика превышает критическое значение $t_{кр}$, где для сравнения одной пары способов замены при $m = 2, 3, ..., 8$ и четырех вариантов генерации получаем 28 измерений.

Как видно из табл. 3, тенденция сохранилась и наиболее сильные различия наблюдаются в способах замены наблюдения со случайным выбором. Также очевидно, что при выборе самого «старого» наблюдения для замены (D_{2i}) число переходов не будет сильно отличаться, вне зависимости от того, является оно частью решения на предыдущей итерации или нет. Это связано с тем, что вероятность получения решения в узловой точке, сформированной первым согласно порядковому номеру выборки наблюдением, мала и уменьшается с ростом n. Данное рассуждение подтверждается полученным в табл. 3 результатом, согласно которому D_{2i} абсолютно однородны.

Согласно полученным в табл. 2 и 3 результатам, между исследуемыми способами замены «старых» наблюдений во время проведения динамического анализа данных отсутствуют существенные различия. Наибольшее отклонение можно наблюдать для D_{1i}, которое в конечном счете не будет оказывать значимого воздействия как на время поиска решения, так и на общую вычислительную сложность. Это связано с тем, что несмотря на выигрыш, достигаемый D_{11} за счет гарантированного близкого расположения к новому решению, разница между ними нивелируется вычислениями для проверки принадлежности наблюдения множеству Z^* и выбора очередного наблюдения, если оно в нем содержалось. В связи с этим будем считать множества, содержащие число переходов между узловыми прямыми, для рассматриваемых способов замены примерно однородными.

Однако данный вывод, несмотря на минимальные различия между D_{ij}, не позволяет однозначно определить оптимальный способ замены, поскольку однородность между сравниваемыми выборками не абсолютна. Воспользуемся методом Монте-Карло и сгенерируем загрязненные и чистую выборки с 10 000 наблюдений при $m = 2, 3, ..., 9$. Поскольку при различных n будут наблюдаться схожие соотношения результатов, рассмотрим среднее время, затраченное на одну итерацию, при $n = 200$. Вычисления проводим на ноутбуке марки Dell G5 5587 с 6-ядерным процессором Intel Core i7-8750H, с тактовой частотой до 2,2 ГГц (гигагерц); программа реализована в среде программирования Microsoft Visual C++.

Разница в среднем времени обработки одной итерации (табл. 4) действительно небольшая и составляет сотые или десятые доли миллисекунды, но даже такая разница может оказаться существенной при стремительном росте размера анализируемой выборки. Очевидно, что сохранение решения на предыдущей итерации алгоритма гарантирует более быстрый спуск к новому решению и, как показало исследование, этот выигрыш будет сильнее, чем проигрыш вследствие необходимости проверки наблюдения на принадлежность множеству Z^*. Наименьшее среднее время при обработке одной итерации алгоритма при различных m стабильно принадлежало способу D_{11}, поэтому в дальнейших расчетах использовали именно его.

В [15] показано, что выигрыш во времени оценивания параметров регрессионных моделей у динамической реализации относительно статической достигается за счет более близкого расположения первого приближения алгоритма спуска

Таблица 3. Процентное выражение превышения критического значения t-статистикой

Table 3. Percentage of exceeding the critical value by t-statistic

	D_{11}	D_{12}	D_{21}	D_{22}
D_{11}	0,00	0,18	0,07	0,11
D_{12}	0,18	0,00	0,11	0,04
D_{21}	0,07	0,11	0,00	0,00
D_{22}	0,11	0,04	0,00	0,00

Таблица 4. Среднее время (мс) обработки одной итерации динамической реализации алгоритма спуска при $n = 200$

Table 4. Average processing time of one iteration of the dynamic implementation of the descent algorithm for $n = 200$ in msec

D	m							
	2	3	4	5	6	7	8	9
D_{11}	1,99	4,13	7,27	11,35	16,41	22,64	30,76	40,03
D_{12}	2,04	4,18	7,33	11,72	16,94	23,36	31,58	42,03
D_{21}	2,03	4,18	7,35	11,44	16,50	22,96	31,02	40,66
D_{22}	2,04	4,26	7,48	11,71	16,97	23,53	31,94	41,32

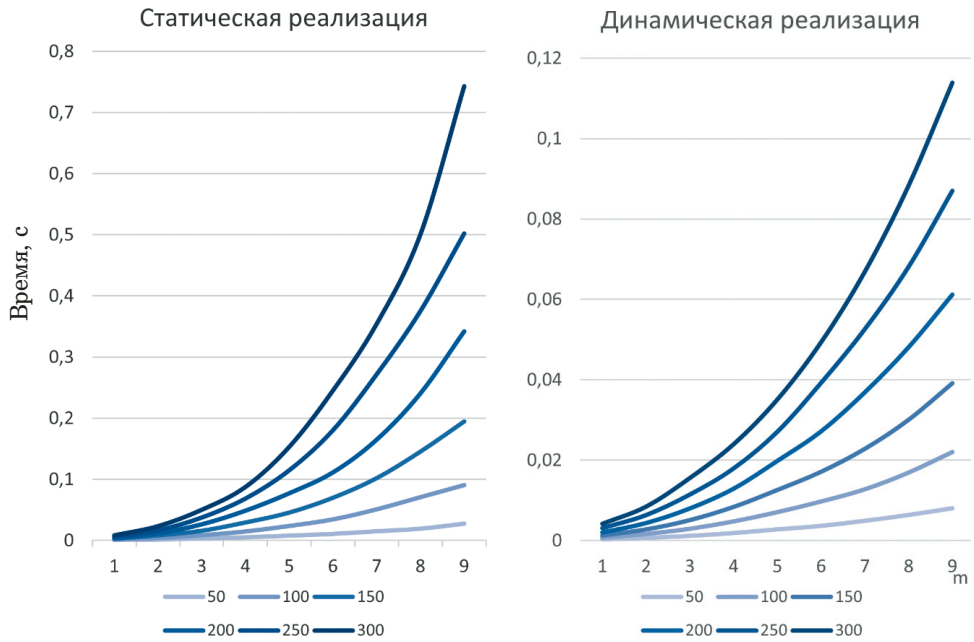

Рис. 4. Среднее время (в секундах), затраченное алгоритмом градиентного спуска на обработку данных при $m = 2, 3, ..., 9$ и $n = 50, 100, ..., 300$ для его статической и динамической реализации

Fig. 4. Average time (in sec) spent by the gradient descent algorithm for data processing at $m = 2, 3, ..., 9$ and $n = 50, 100, ..., 300$ for its static and dynamic implementation

к искомому решению. Сравним время, затраченное алгоритмами на поиск решения, при этом в качестве эталонного будем считать время, потраченное МНК на те же операции. Используем ранее сгенерированные выборки и вычислим среднее время, затраченное алгоритмами на их обработку при $n = 50, 100, ..., 300$.

Как видно из рис. 4, рост затраченного времени при динамической реализации более сглаженный, чем при статической. В более явном виде данная тенденция отражена в табл. 5, в которой приведен выигрыш динамической реализации алгоритма относительно статической в процентном соотношении.

В табл. 6 показано, во сколько раз в среднем время вычислений статического варианта алгоритма больше времени вычислений динамического варианта алгоритма, а также время вычислений динамического варианта алгоритма больше времени вычислений с помощью МНК. Исходя из полученных данных, можно подтвердить тезис о более сглаженном характере графиков в динамике, что говорит о снижении влияния роста числа коэффициентов m и числа наблюдений n. Следует отметить, что наибольший выигрыш достигается с ростом n и m, но увеличение выигрыша постепенно замедляется.

Согласно полученным результатам (см. табл. 6), наблюдается сильная зависимость времени анализа от m и слабая — от n. В результате в рамках одного столбца мы видим как увеличение отношения времени, так и его незначительное уменьшение. Снижение показателя может свидетельствовать о приближении к пределу выигрыша в рамках столбца, соответствующего числу коэффициентов m.

Выигрыш во времени поиска решения при помощи алгоритма градиентного спуска в динамике относительно статики составляет $(\lg n)^{0,66} m^{0,47}$ с коэффициентом детерминации

Таблица 5. Выигрыш (в процентном отношении) среднего времени поиска решения динамической реализации алгоритма спуска относительно статической при $m = 2, 3, ..., 9$ и $n = 100, 200, 300$

Table 5. Gain (in terms of percentage) in the average time of searching for a solution of the dynamic implementation of the descent algorithm relative to the static one at $m = 2, 3, ..., 9$ and $n = 100, 200, 300$

n	m							
	2	3	4	5	6	7	8	9
100	43	59	66	67	70	72	74	76
200	46	62	68	71	76	77	79	80
300	49	64	69	74	77	80	81	82

функции зависимости $R^2 = 0{,}872$. Степени при коэффициентах можно интерпретировать следующим образом — чем меньшее значение они принимают, тем ближе друг к другу сравниваемые показатели. Аналогичное соотношение динамического анализа к МНК оказалось равным $(\lg n)^{0{,}97}m^{0{,}96}$ с $R^2 = 0{,}868$. Следовательно, исходя из отношения значения степеней можем сделать вывод о среднем выигрыше времени анализа выборки динамики относительно статики примерно на 36,6 % или о приближении на одну треть ко времени вычисления МНК относительно исходного времени при статической реализации.

Отметим, что для вероятностно-статистических моделей линейной регрессии (1) важно, являются ли независимые переменные детерминированными или случайными. Свойства оценок зависят от выбора между этими двумя вариантами, как показано, например, в [21]. В работе представлены алгоритмы получения оценок, их вероятностно-статистические свойства не рассматривались. Для эффективного применения методов оценивания такие свойства в дальнейшем должны быть изучены, например, — получены доверительные интервалы для оценок.

Заключение

Описан алгоритм реализации МНМ на основе градиентного спуска по узловым прямым для динамических задач оценивания регрессионных моделей. Анализ различных вариантов замены «старого» наблюдения на новое показал отсутствие различий в вычислительной сложности. Это позволяет при выборе наилучшего варианта ориентироваться на особенности используемой среды программирования.

Поскольку вычислительная сложность в виде качественной оценки числа операций дает лишь грубую оценку вычислительных затрат, выполнен сравнительный анализ фактического времени вычислений динамического варианта алгоритма и статического, а также алгоритма, реализующего МНК. Исследования показали, что динамический вариант алгоритма градиентного спуска по узловым прямым в среднем в $(\lg n)^{0{,}66}m^{0{,}47}$ раз выигрывает по быстродействию по сравнению со статическим вариантом. Для распространенных практических ситуаций динамический вариант алгоритма градиентного спуска по узловым прямым приближен по быстродействию к МНК, например, при $n \le 100$ и $m \le 5$ он проигрывает по быстродействию не более чем в 6 раз. Это позволяет использовать предложенный алгоритм градиентного спуска по узловым прямым на практике в динамических задачах оценивания широкого класса систем.

Финансирование

Работа выполнена при финансовой поддержке гранта РФФИ, проект № 20-41-660008 р_а.

Таблица 6. Отношения среднего времени анализа статической и динамической реализации, а также динамической реализации и МНК: $m = 2, 3, ..., 9$ и $n = 50, 100, ..., 300$

Table 6. Ratios of the average analysis time: *a* — static to dynamic implementations; *b* — dynamic to least squares $m = 2, 3, ..., 9$ and $n = 50, 100, ..., 300$

Статика/Динамика					Динамика/МНК				
n	m				n	m			
	2	3	4	5		2	3	4	5
50	1,66	2,25	2,73	2,72	50	1,46	2,69	3,72	4,90
100	1,75	2,44	2,91	3,08	100	1,63	3,27	4,87	6,10
150	1,76	2,53	3,04	3,46	150	2,11	3,65	5,74	7,71
200	1,84	2,64	3,17	3,71	200	2,58	4,16	6,30	8,92
250	1,83	2,59	3,27	3,82	250	3,81	4,84	7,48	9,95
300	1,95	2,74	3,21	3,85	300	3,49	5,66	8,68	11,28
n	m				n	m			
	6	7	8	9		6	7	8	9
50	2,81	2,91	3,24	3,00	50	6,31	7,60	9,27	9,49
100	3,34	3,54	3,90	4,17	100	8,21	10,23	11,80	14,46
150	3,59	3,94	4,63	4,69	150	8,68	12,58	13,59	17,42
200	4,11	4,26	4,83	4,90	200	9,72	14,72	17,74	21,22
250	4,21	4,60	5,27	5,68	250	12,29	15,53	17,48	22,59
300	4,36	5,07	5,39	5,67	300	13,94	16,11	21,25	25,12

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.

ЛИТЕРАТУРА

1. Вероятность и математическая статистика: Энциклопедия / Гл. ред. Ю. В. Прохоров. — М.: Большая Российская энциклопедия, 1999. — 910 с.
2. **Arkes J.** Regression Analysis: A Practical Introduction. — New York: Taylor & Francis Group, 2019. — 363 p.
3. **Hoffmann J. P.** Linear Regression Models: Applications in R. — Boca Raton: CRC Press, 2022. — 437 p.
4. **Демиденко Е. З.** Линейная и нелинейная регрессия. — М.: Финансы и статистика, 1981. — 302 с.
5. **Мудров В. И., Кушко В. Л.** Методы обработки измерений. Квазиправдоподобные оценки. — М.: Радио и связь, 1983. — 304 с.
6. **Bloomfield P., Steiger W. L.** Least Absolute Seviations: Theory, Applications, and Algorithms. — Boston – Basel – Stuttgart: Birkhauser, 1983. — 349 p.
7. **Birkes D., Dodge Y.** Alternative Methods of Regression. — John Wiley & Sons, 1993. — 239 p.
8. **Armstrong R. D., Kung D. S.** Algorithm AS 132: Least absolute value estimates for a simple linear regression problem / Applied Statistics 1978. Vol. 7. P. 363 – 366. DOI: 10.2307/2347181
9. **Wesolowsky G. O.** A new descent algorithm for the least absolute value regression problem / Communications in Statistics, Simulation and Computation. 1981. Vol. 10. N 5. P. 479 – 491. DOI: 10.1080/03610918108812224
10. **Акимов П. А., Матасов А. И.** Уровни неоптимальности алгоритма Вейсфельда в методе наименьших модулей / Автоматика и телемеханика. 2010. № 2. С. 4 – 16.
11. **Krzic A. S., Sersic D.** L1 Minimization Using Recursive Reduction of Dimensionality / Signal Processing. 2018. Vol. 151. P. 119 – 129. DOI: 10.1016/j.sigpro.2018.05.002
12. **Wei Xue, Wensheng Zhang, Gaohang Yu.** Least absolute deviations learning of multiple tasks / Journal of Industrial & Management Optimization. 2018. N 14(2). P. 719 – 729. DOI: 10.3934/jimo.2017071
13. **Тырсин А. Н.** Алгоритмы спуска по узловым прямым в задаче оценивания регрессионных уравнений методом наименьших модулей / Заводская лаборатория. Диагностика материалов. 2021. Т. 87. № 5. С. 68 – 75. DOI: 10.26896/1028-6861-2021-87-5-68-75
14. **Акимов П. А., Матасов А. И.** Итерационный алгоритм для L_1-аппроксимации в динамических задачах оценивания / Автоматика и телемеханика. 2015. № 5. С. 7 – 26. DOI: 10.1134/S000511791505001X
15. **Тырсин А. Н., Голованов О. А.** Динамическое регрессионное моделирование на основе градиентного спуска по узловым прямым / Современные наукоемкие технологии. 2021. № 10. С. 88 – 93. DOI: 10.17513/snt.38859
16. **Тырсин А. Н., Азарян А. А.** Точное оценивание линейных регрессионных моделей методом наименьших модулей на основе спуска по узловым прямым / Вестник ЮУрГУ. Серия «Математика. Механика. Физика». 2018. Т. 10. № 2. С. 47 – 56. DOI: 10.14529/mmph180205
17. **Местников С. В., Эверстова Г. В.** Преобразование Жордана-Гаусса и линейная оптимизация. — Якутск: Издательский дом СВФУ, 2019. — 160 с.
18. **Михайлов Г. А., Войтишек А. В.** Численное статистическое моделирование. Методы Монте-Карло. — М.: Издательский центр «Академия», 2006. — 368 с.
19. **Tukey J. W.** A Survey of Sampling from Contaminated Distribution / Contributions to Probability and Statistics. — Stanford: Stanford Univ. Press, 1960. P. 443 – 485.
20. **Хьюбер П.** Робастность в статистике / Пер. с англ. — М.: Мир, 1984. — 304 с.
21. **Орлов А. И.** Многообразие моделей регрессионного анализа (обобщающая статья) / Заводская лаборатория. Диагностика материалов. 2018. Т. 84. № 5. С. 63 – 73. DOI: 10.26896/1028-6861-2018-84-5-63-73

REFERENCES

1. Probability and mathematical statistics: Encyclopedia. — Moscow: Bol'shaya Rossiiskaya Éntsiklopediya, 1999. — 910 p. [in Russian].
2. **Arkes J.** Regression Analysis: A Practical Introduction. — New York: Taylor & Francis Group, 2019. — 363 p.
3. **Hoffmann J. P.** Linear Regression Models: Applications in R. — Boca Raton: CRC Press, 2022. — 437 p.
4. **Demidenko E. Z.** Linear and non-linear regression. — Moscow: Finansy i Statistika, 1981. — 302 p. [in Russian].
5. **Mudrov V. I., Kushko V. L.** Measurement processing methods. Quasi-plausible estimates. — Moscow: Radio i svyaz', 1983. — 304 p. [in Russian].
6. **Bloomfield P., Steiger W. L.** Least Absolute Seviations: Theory, Applications, and Algorithms. — Boston – Basel – Stuttgart: Birkhauser, 1983. — 349 p.
7. **Birkes D., Dodge Y.** Alternative Methods of Regression. — John Wiley & Sons, 1993. — 239 p.
8. **Armstrong R. D., Kung D. S.** Algorithm AS 132: Least absolute value estimates for a simple linear regression problem / Applied Statistics 1978. Vol. 7. P. 363 – 366. DOI: 10.2307/2347181
9. **Wesolowsky G. O.** A new descent algorithm for the least absolute value regression problem / Communications in Statistics, Simulation and Computation. 1981. Vol. 10. N 5. P. 479 – 491. DOI: 10.1080/03610918108812224
10. **Akimov P. A., Matasov A. I.** Nonoptimality levels of the Weisfeld algorithm in the method of the least modules / Avtom. Teplomekh. 2010. N 2. P. 4 – 16 [in Russian].
11. **Krzic A. S., Sersic D.** L1 Minimization Using Recursive Reduction of Dimensionality / Signal Processing. 2018. Vol. 151. P. 119 – 129. DOI: 10.1016/j.sigpro.2018.05.002
12. **Wei Xue, Wensheng Zhang, Gaohang Yu.** Least absolute deviations learning of multiple tasks / Journal of Industrial & Management Optimization. 2018. N 14(2). P. 719 – 729. DOI: 10.3934/jimo.2017071
13. **Tyrsin A. N.** Algorithms for descending along nodal lines in the problem of estimating regression equations by the method of least modules / Zavod. Lab. Diagn. Mater. 2021. Vol. 87. N 5. P. 68 – 75 [in Russian]. DOI: 10.26896/1028-6861-2021-87-5-68-75
14. **Akimov P. A., Matasov A. I.** Iterative algorithm for L_1-approximation in dynamic estimation problems / Avtom. Teplomekh. 2015. N 5. P. 7 – 26 [in Russian]. DOI: 10.1134/S000511791505001X
15. **Tyrsin A. N., Golovanov O. A.** Dynamic regression modeling based on gradient descent along nodal lines / Sovr. Naukoem. Tekhnol. 2021. N 10. P. 88 – 93 [in Russian]. DOI: 10.17513/snt.38859
16. **Tyrsin A. N., Azaryan A. A.** Accurate least moduli estimation of linear regression models based on nodal descent / Vestn. YuUrGU. Ser. Mat. Mekh. Fiz. 2018. Vol. 10. N 2. P. 47 – 56 [in Russian]. DOI: 10.14529/mmph180205
17. **Mestnikov S. V., Everstova G. V.** Jordan – Gauss transform and linear optimization. — Yakutsk: Izd. dom SVFU, 2019. — 160 p. [in Russian].
18. **Mikhailov G. A., Voitishek A. V.** Numerical statistical modeling. Monte-Carlo methods. — Moscow: Akademiya, 2006. — 368 p. [in Russian].
19. **Tukey J. W.** A Survey of Sampling from Contaminated Distribution / Contributions to Probability and Statistics. — Stanford: Stanford Univ. Press, 1960. P. 443 – 485.
20. **Huber P.** Robustness in statistics. — Moscow: Mir, 1984. — 304 p. [Russian translation].
21. **Orlov A. I.** Diversity of the models for regression analysis (generalizing article) / Zavod. Lab. Diagn. Mater. 2018. Vol. 84. N 5. P. 63 – 73 [in Russian]. DOI: 10.26896/1028-6861-2018-84-5-63-73

A Message from the Directors of the BIPM and the BIML
World Metrology Day — 20 May 2023

Bureau International des Poids et Mesures

OIML

Martin Milton
Director of the BIPM

Anthony Donnellan
Director of the BIML

Measurements supporting the global food system

Food is a major concern for every one of us. Providing access to safe and affordable food remains one of the major challenges for governments worldwide. This is also the goal of farmers and food producers who trade products through distributors and retailers to consumers at international, national and local levels. In 2021, this trade was worth 22 trillion USD and accounted for approximately 20% of all global trade.

To trade internationally and to access markets for high-value products, producers must be able to show that they meet food standards. Additionally, governments need to ensure safety and fair trade especially in local markets for food. All of this is supported by reliable measurements of the quantity and quality of the primary and processed food products involved.

Our focus for World Metrology Day in 2023 is on the many measurement challenges that must be addressed to make the global food system work. For example:

— the quantity of food bought and sold is measured according to its mass or volume. These measurements range from the large volumes of grain and wheat traded internationally down to rapid online weighing measurements to ensure pre-packaged goods are labelled correctly;

— the effective storage and packaging of food depends on the accurate control of the temperature and humidity of its storage environment;

— the quality and authenticity of food is determined by measuring its chemical composition. This requires measurements to ensure that it contains the stated levels of vitamins through to measurements of its isotopic composition to validate the origin of high-value foods such as honey or wine; and

— the safety of food is ensured by careful measurement to detect the presence of chemical contamination such as pesticide residues and heavy metals or biological contamination such as mycotoxins.

It is now recognised that the depletion of natural resources and the impact of climate change pose major challenges to the global food system such that the goal of a world with zero hunger and universal access to clean water was included amongst the Sustainable Development Goals set by the United Nations.

We again look forward to celebrating World Metrology Day with our stakeholders around the world.