КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2021 Т. 13 № 5 С. 947–963
DOI: 10.20537/2076-7633-2021-13-5-947-963

Ки&M

ЧИСЛЕННЫЕ МЕТОДЫ И ОСНОВЫ ИХ РЕАЛИЗАЦИИ

УДК: 519.85

Ускоренные адаптивные по константам сильной выпуклости и Липшица для градиента методы первого порядка

Н. В. Плетнев[1, 2]

[1]Московский физико-технический институт,
Россия, 141701, г. Долгопрудный, Институтский пер., д. 9
[2]Вычислительный центр им. А. А. Дородницына Российской академии наук Федерального исследовательского центра «Информатика и управление» Российской академии наук,
Россия, 119333, Москва, ул. Вавилова, 40

E-mail: nikita.pletnev@phystech.edu

Получено 19.05.2020, после доработки — 03.09.2021.
Принято к публикации 03.09.2021.

Работа посвящена построению эффективных и применимых к реальным задачам методов выпуклой оптимизации первого порядка, то есть использующих только значения целевой функции и ее производных. При построении используется быстрый градиентный метод OGM-G, который является оптимальным по оракульной сложности (числу вычислений градиента целевой функции), но при запуске требует знания констант сильной выпуклости и Липшица градиента для вычисления количества шагов и длины шага, требуемых для достижения заданной точности. Данное требование усложняет практическое использование метода. Предлагаются адаптивный по константе сильной выпуклости алгоритм ACGM, основанный на рестартах OGM-G с обновлением оценки константы сильной выпуклости, и адаптивный по константе Липшица градиента метод ALGM, в котором применение рестартов OGM-G дополнено подбором константы Липшица с проверкой условий гладкости, используемых в методе универсального градиентного спуска. При этом устраняются недостатки исходного метода, связанные с необходимостью знания данных констант, что делает возможным практическое использование. Доказывается, что оценки сложности построенных алгоритмов являются оптимальными с точностью до числового множителя. Для проверки полученных результатов проводятся эксперименты на модельных функциях и реальных задачах машинного обучения.

Ключевые слова: быстрый градиентный метод, адаптивность по константе сильной выпуклости, адаптивность по константе Липшица градиента

COMPUTER RESEARCH AND MODELING
2021 VOL. 13 NO. 5 P. 947–963
DOI: 10.20537/2076-7633-2021-13-5-947-963

Ки&М

NUMERICAL METHODS AND THE BASIS FOR THEIR APPLICATION

UDC: 519.85

Fast adaptive by constants of strong-convexity and Lipschitz for gradient first order methods

N. V. Pletnev[1, 2]

[1]Moscow Institute of Physics and Technology,
9 Institute lane, Dolgoprudny, 141701, Russia
[2]Institution of Russian Academy of Sciences Dorodnicyn Computing Centre of RAS,
40 Vavilov st., Moscow, 119333, Russia

E-mail: nikita.pletnev@phystech.edu

Received 19.05.2020, after completion — 03.09.2021.
Accepted for publication 03.09.2021.

The work is devoted to the construction of efficient and applicable to real tasks first-order methods of convex optimization, that is, using only values of the target function and its derivatives. Construction uses OGM-G, fast gradient method which is optimal by complexity, but requires to know the Lipschitz constant for gradient and the strong convexity constant to determine the number of steps and step length. This requirement makes practical usage very hard. An adaptive on the constant for strong convexity algorithm ACGM is proposed, based on restarts of the OGM-G with update of the strong convexity constant estimate, and an adaptive on the Lipschitz constant for gradient ALGM, in which the use of OGM-G restarts is supplemented by the selection of the Lipschitz constant with verification of the smoothness conditions used in the universal gradient descent method. This eliminates the disadvantages of the original method associated with the need to know these constants, which makes practical usage possible. Optimality of estimates for the complexity of the constructed algorithms is proved. To verify the results obtained, experiments on model functions and real tasks from machine learning are carried out.

Keywords: fast gradient method, adaptivity on the constant for strong convexity, adaptivity on the Lipschitz constant for gradient

Citation: *Computer Research and Modeling*, 2021, vol. 13, no. 5, pp. 947–963 (Russian).

1. Введение

Работа посвящена методам оптимизации первого порядка, то есть методам, использующим лишь значения функции и ее градиента.

Задачи оптимизации функций высокой размерности имеют многообразные приложения, например, в машинном обучении, управлении, экономике и энергетике. Поскольку точное решение данных задач чаще всего невозможно, необходимо применять приближенные методы.

На сложность задачи оптимизации влияет гладкость функции, а также то, является ли она выпуклой. Как известно, для выпуклых функций локальный минимум всегда является глобальным, и необходимое условие экстремума — равенство градиента нулю — становится достаточным. Методы, рассматриваемые в работе, предназначены для решения задачи выпуклой оптимизации.

Методы первого порядка пользуются большой популярностью в связи с относительно невысокой вычислительной сложностью: они требуют вычисления только значения функции, ее градиента и простейших векторных операций.

В настоящее время активно развиваются быстрые градиентные методы, основанные на следующей идее: задается число операций, строятся оптимальные для данного числа операций последовательности коэффициентов, которые используются для получения последовательности точек. Такой подход реализован в статье [Kim, Fessler, 2018]. Построенный в ней метод OGM-G является оптимальным среди методов с фиксированным числом шагов, в статье доказаны оценки для его скорости сходимости и приведен пример функции, для которой улучшение этих оценок невозможно — откуда следует оптимальность. Изложению результатов [Kim, Fessler, 2018] в части оценок, касающихся целей работы, посвящен § 3. Общее описание идеи можно найти в пособии [Гасников, 2019].

Проблема данного подхода заключается в том, что требуемое для достижения заданного результата, например уменьшения нормы градиента вдвое, число итераций неизвестно. Поэтому для эффективного применения подобных методов необходимо оценивать это число.

В пособии [Гасников, 2019], посвященном изложению современного состояния градиентных методов, предлагается способ оценки, но он требует знания константы сильной выпуклости μ и константы Липшица градиента L. Также там указана предложенная Ю. Е. Нестеровым в статье [Нестеров, 1989] идея применения быстрого градиентного метода с оцениванием данного параметра и обновлением его значения при каждом рестарте. В § 5 пособия изложен придуманный Ю. Е. Нестеровым (первоисточник [Nesterov, 2015]) метод адаптивного подбора константы Липшица, известный как универсальный градиентный спуск. На основе данного метода в работе построен быстрый алгоритм, адаптивный как по константе сильной выпуклости, так и по константе Липшица для градиента. На этих идеях построено основное содержание работы — §§ 4 и 5.

В данной работе конструируются методы первого порядка, имеющие оптимальные оценки сходимости и адаптивные по константам сильной выпуклости и Липшица градиента. Они избавлены от указанных недостатков, поэтому могут применяться на практике.

Близкой задаче посвящена статья [Lei, Jordan, 2019]. В ней строится адаптивный по константе сильной выпуклости метод стохастического градиентного спуска. Однако цель полностью не достигнута, так как полученный алгоритм является эффективным лишь с точностью до логарифмического множителя.

В статье [Fercoq, Qu, 2016] также строится метод, адаптивный по константе сильной выпуклости с использованием рестартов, но оценка сложности полученного алгоритма тоже содержит логарифмический множитель. Однако там рассматривается случай для выпуклой функции с негладким слагаемым.

Работа [Barre, Taylor, d'Aspremont, 2020] предлагает метод, адаптивный по константе сильной выпуклости и использующий для вычисления шага норму градиента и невязку по функции. Соответственно, для его применения требуется знание значения функции в оптимуме.

2. Определения и предположения

Решается задача безусловной минимизации:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}). \tag{1}$$

Предполагается, что решение

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \tag{2}$$

существует, а градиент функции $f(\mathbf{x})$ удовлетворяет условию Липшица с константой $L > 0$:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leqslant L\|\mathbf{x} - \mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \tag{3}$$

Здесь и далее используется 2-норма: $\|\mathbf{a}\| = \|\mathbf{a}\|_2 = \sqrt{a_1^2 + \ldots + a_d^2}$.

Также считается, что функция $f(\mathbf{x})$ является сильно выпуклой с неизвестной нам константой $\mu > 0$:

$$\frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^*\|^2 \leqslant f(\mathbf{x}) - f(\mathbf{x}^*) \leqslant \frac{1}{2\mu}\|\nabla f(\mathbf{x})\|^2. \tag{4}$$

Первое неравенство напрямую следует из определения, второе доказывается в соответствии с [Гасников, 2019]:

$$f(\mathbf{x}^*) = \min_{\mathbf{y}} f(\mathbf{y}) \geqslant \min_{\mathbf{y}} \left(f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2 \right) = f(\mathbf{x}) - \frac{1}{2\mu}\|\nabla f(\mathbf{x})\|^2.$$

В качестве невязки нахождения точки экстремума функции используется норма градиента. Критерий останова выглядит так:

$$\|\nabla f(\mathbf{x})\| \leqslant \varepsilon. \tag{5}$$

Траекторией метода оптимизации называется последовательность порождаемых им точек.

Рестартом (согласно [Гасников, 2019]) называется перезапуск метода с использованием результата предыдущего запуска в качестве начального значения.

Константа, с которой функция удовлетворяет определению сильной выпуклости в окрестности некоторой части траектории, превосходящая μ, обозначается как μ^{loc}. Аналогично определяется L^{loc}.

Алгоритм называется адаптивным по некоторому параметру, если его применение не требует никаких предположений о значении данного параметра.

3. Исходный алгоритм

3.1. OGM-G — оптимальный неадаптивный метод

В [Kim, Fessler, 2018] показано, что оптимальным в классе методов с заданным числом шагов фиксированной длины является ускоренный градиентный метод OGM-G. Однако он не является адаптивным ни по константе Липшица, ни по константе сильной выпуклости. Данный алгоритм используется в работе для построения оптимальных адаптивных методов.

Algorithm 1. Optimal Gradient Method (OGM-G)

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\mathbf{x}_0 \in \mathbb{R}^d$ — начальная точка, $N \geqslant 1$.

1: $\mathbf{y}_0 := \mathbf{x}_0$.
2: **for** $k = 0, \ldots, N-1$ **do**
3: $\quad \mathbf{y}_{i+1} := \mathbf{x}_i - \frac{1}{L}\nabla f(\mathbf{x}_i)$;
4: $\quad \mathbf{x}_{i+1} := \mathbf{y}_{i+1} + \beta_i(\mathbf{y}_{i+1} - \mathbf{y}_i) + \gamma_i(\mathbf{y}_{i+1} - \mathbf{x}_i)$.
5: **end for**

Output: $\mathbf{x}_N$.

Коэффициенты β_i, γ_i вычисляются по формулам

$$\beta_i = \frac{(\theta_i - 1)(2\theta_{i+1} - 1)}{\theta_i(2\theta_i - 1)}, \quad \gamma_i = \frac{2\theta_{i+1} - 1}{2\theta_i - 1},$$

где последовательность $\{\theta_i\}_{i=0}^N$ строится следующим образом:

$$\theta_i = \begin{cases} \dfrac{1 + \sqrt{1 + 8\theta_1^2}}{2}, & i = 0; \\ \dfrac{1 + \sqrt{1 + 4\theta_{i+1}^2}}{2}, & 1 \leqslant i < N; \\ 1, & i = N. \end{cases}$$

3.2. Оценки

По теореме 2 из [1], при применении OGM-G

$$\|\nabla f(\mathbf{x}^N)\|^2 \leqslant \frac{4L(f(\mathbf{x}^0) - f(\mathbf{x}^*))}{N^2}. \tag{6}$$

Оттуда же

$$f(\mathbf{x}^N) - f(\mathbf{x}^*) \leqslant \frac{L\|\mathbf{x}^0 - \mathbf{x}^*\|^2}{N^2}. \tag{7}$$

Из (4) и (6)

$$\|\nabla f(\mathbf{x}^N)\|^2 \leqslant \frac{4L}{N^2}\frac{1}{2\mu}\|\nabla f(\mathbf{x}^0)\|^2, \tag{8}$$

или

$$\|\nabla f(\mathbf{x}^N)\| \leqslant \sqrt{\frac{2L}{\mu N^2}}\|\nabla f(\mathbf{x}^0)\|. \tag{9}$$

Таким образом, выполнение N итераций гарантирует уменьшение нормы градиента $f(x)$ как минимум вдвое, где

$$N = 2\sqrt{2\frac{L}{\mu}}. \tag{10}$$

Согласно лемме 4 статьи [Kim, Fessler, 2018], полученная оценка является неулучшаемой в худшем случае.

3.3. Недостатки метода

Полученная оценка показывает, что использование OGM-G неявно предполагает, помимо наличия известной константы Липшица, знание константы сильной выпуклости.

Практически во всех реальных случаях применения методов оптимизации ни одно из этих предположений не выполняется. Свойства функции заранее неизвестны, а вычисление констант требует нахождения минимума и максимума собственных значений матрицы Гессе. Эта задача может быть даже сложнее, чем исходная задача оптимизации.

Указанные соображения делают оптимальный теоретически метод неприменимым на практике. Решению данной проблемы посвящен следующий раздел.

4. Адаптивность по константе сильной выпуклости

4.1. ACGM — адаптивный по константе сильной выпуклости метод

Ю. Е. Нестеровым в работе [Нестеров, 1989] предложен способ построения адаптивного по μ алгоритма, основанного на рестартах.

В этом разделе OGM-G используется в качестве «черного ящика», получающего на вход функцию f, начальную точку $\mathbf{x}_0$, константу Липшица L и затравочную (начальную) константу сильной выпуклости μ_0. Число итераций N, используемое методом, вычисляется по формуле (10). В дальнейшем применение OGM-G как шага в алгоритмах будет обозначаться как $OGMG(f, \mathbf{x}_0, L, \mu_0)$.

Algorithm 2. Adaptive by strong Convexity Gradient Method (ACGM)

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\mathbf{x}_0 \in \mathbb{R}^d$ — начальная точка, L, μ_0, β, ε.

for $k \geqslant 0$ **do**
  if $\|\nabla f(\mathbf{x}_k)\| \leqslant \varepsilon$ **then**
    break;
  end if
  $\mu_k := \beta\mu_{k-1}$;
  $\mathbf{x}_k := OGMG(f, \mathbf{x}_{k-1}, L, \mu_k)$;
  if $\|\nabla f(\mathbf{x}_k)\| \leqslant \frac{1}{2}\|\nabla f(\mathbf{x}_{k-1})\|$ **then**
    continue;
  end if
  $\mu_k := \frac{\mu_k}{\beta}$;
  if $\|\nabla f(\mathbf{x}_k)\| < \|\nabla f(\mathbf{x}_{k-1})\|$ **then**
    $\mathbf{x}_{k-1} := \mathbf{x}_k$;
  end if
  goto 6:
end for

Output: $[\mathbf{x}_0 \dots \mathbf{x}_N]$.

ACGM — adaptive by constant of strong convexity gradient method — решает проблему неизвестности μ, инициализируя ее произвольным значением с последующим изменением.

На каждом шаге предполагаемое значение μ умножается на одно и то же $\beta > 1$.

4.2. Оценки

В результате применения ACGM очередное уменьшение вдвое нормы градиента будет выполнено за

$$2\sqrt{2\frac{L}{\mu_k^{init}}} + 2\sqrt{2\frac{L}{\mu_k^{init}/\beta}} + \ldots + 2\sqrt{2\frac{L}{\mu_k^{init}/\beta^m}} = \sqrt{8\frac{L}{\mu_k}}\sum_{i=0}^{m}\frac{1}{\sqrt{\beta^i}} \lesssim \frac{\sqrt{8\beta}}{\sqrt{\beta}-1}\sqrt{\frac{L}{\mu_k}}$$

итераций метода OGM-G, где m — количество повторений цикла на шаге k. Индекс *init* указывает, что в формуле используется не конечное значение переменной, а то, которым она была инициализирована.

При этом μ_k отличается не более чем в β раз от μ_k^{loc}. Использование значения μ, подходящего для всего пространства, могло бы повысить количество операций.

Действительно, если последовательные s точек траектории ACGM лежат в области, в которой $f(\mathbf{x})$ сильно выпукла с константой $\mu^{loc} \geqslant \beta^s\mu$, то в данных точках ACGM применяется с $\mu_0 \geqslant \frac{\mu^{loc}}{\beta} \geqslant \beta^{s-1}\mu$. Тогда количество обращений к вычислению градиента для каждого уменьшения его нормы вдвое оказывается не более $2\sqrt{2\frac{L}{\beta^{s-1}\mu}}$ — то есть в $\beta^{\frac{s-1}{2}}$ раз меньше, чем при $\mu_k \equiv \mu$.

Суммарное количество итераций при работе ACGM с использованием критерия останова (5) оценивается следующим образом. Требуется выполнить $K = \log_2\frac{\|\nabla f(\mathbf{x}^0)\|}{\varepsilon}$ шагов. Каждый шаг содержит $O\left(\sqrt{\frac{L}{\mu}}\right)$ итераций, поэтому алгоритм завершит работу, выполнив $O\left(\sqrt{\frac{L}{\mu}}\log_2\frac{\|\nabla f(\mathbf{x}^0)\|}{\varepsilon}\right)$ итераций, то есть вычислений $f(\mathbf{x})$ и $\nabla f(\mathbf{x})$.

Как показано выше, полученная оценка по порядку величины может быть уточнена. Если $\mu_k \geqslant \frac{\mu_k^{loc}}{\beta}$, то каждый шаг ACGM содержит не более $\frac{\sqrt{8\beta}}{\sqrt{\beta}-1}\sqrt{\frac{L}{\mu_k}} \leqslant \frac{\sqrt{8}\beta}{\sqrt{\beta}-1}\sqrt{\frac{L}{\mu_k^{loc}}}$ итераций; соответственно, общее количество итераций не превосходит

$$\frac{\sqrt{8}\beta}{\sqrt{\beta}-1}\sum_{k=0}^{K-1}\sqrt{\frac{L}{\mu_k^{loc}}}.$$

Данное вычисление основано на идее оценки из [Гасников и др., 2018].

Минимизация зависящего от β коэффициента дает $\beta = 4$.

Это значение является оптимальным лишь с точки зрения худшего случая, когда $\mu_k = \frac{\mu_k^{loc}}{\beta}$. В реальных случаях, поскольку данное равенство является лишь теоретически возможным предельным случаем, значение коэффициента может оказаться меньше данного, но в любом случае оно превосходит $\inf\limits_{\beta>1}\frac{\sqrt{\beta}}{\sqrt{\beta}-1} = 1$.

Таким образом, доказаны теоремы о сходимости построенного метода.

Теорема 1. *Алгоритм ACGM с оптимальным* $\beta = 4$ *и* $\mu_0 > \max\limits_k \mu_k^{loc}$ *достигает точки* $\mathbf{x}_N$, *удовлетворяющей критерию останова* (5), *за* $N \leqslant C\sum\limits_{k=0}^{K-1}\sqrt{\frac{L}{\mu_k^{loc}}}$ *вычислений градиента, где* $K = \log_2\frac{\|\nabla f(\mathbf{x}^0)\|}{\varepsilon}$, $C = \frac{\sqrt{8}\beta}{\sqrt{\beta}-1} = 8\sqrt{2}$.

Данная теорема имеет лишь теоретический смысл, поскольку оценка μ_k^{loc} крайне затруднительна. Следующая теорема содержит менее точную, но более удобную оценку.

Так как $\mu_k^{loc} \geqslant \mu$ при всех k, каждое слагаемое в сумме из теоремы 1 не превосходит $\sqrt{\frac{L}{\mu}}$, откуда сразу следует

Теорема 2. *Алгоритм ACGM с оптимальным* $\beta = 4$ *и* $\mu_0 > \max\limits_k \mu_k^{loc}$ *достигает точки* $\mathbf{x}_N$, *удовлетворяющей критерию останова* (5), *за* $N \leqslant CK\sqrt{\frac{L}{\mu}}$ *вычислений градиента, где* $K = \log_2 \frac{\|\nabla f(\mathbf{x}^0)\|}{\varepsilon}$, $C = \frac{\sqrt{8}\beta}{\sqrt{\beta}-1} = 8\sqrt{2}$.

4.3. К выбору оптимального μ_0. Случай $\mu_0 < \mu^{loc}$

Для упрощения вычислений пусть $\mu_0 = \frac{\mu^{loc}}{4^k}$. Тогда по формуле (10) на первом шаге ACGM будет выполнено $M = 2\sqrt{2\frac{L \cdot 4^k}{\mu^{loc}}} = 2^k N$ итераций. При этом, согласно (9), норма градиента умножится не более чем на $\sqrt{\frac{2L}{\mu^{loc} M^2}} = \frac{1}{2^{k+1}}$.

Для достижения такого результата требуется $k+1$ рестартов, то есть $(k+1)N$ итераций при использовании OGM-G.

При использовании ACGM i-й рестарт выполняется с $\mu = \mu_0 \beta^i$, то есть потребует $\frac{N}{2^i}$ итераций. Суммарное количество не превосходит $2N$.

Поскольку $2^k > 2$, применение заниженного значения константы сильной выпуклости приводит к значительному увеличению количества итераций.

Соответственно, оптимальным будет такой выбор μ_0, что $\mu_0 > \mu_k^{loc}$ при всех k. Таким является, например, $\mu_0 = L$.

4.4. Иллюстрации

Работа алгоритма иллюстрируется на двух функциях: регуляризованная штрафная функция логистической регрессии, возникающая из приложений, и квадратичная форма с плохой обусловленностью. Их свойства будут рассмотрены в разделе «Эксперименты» (§ 6). На рисунке 1 для логистической регрессии и рисунке 2 для квадратичной формы приведены график зависимости логарифма нормы градиента от номера итерации и траектория метода.

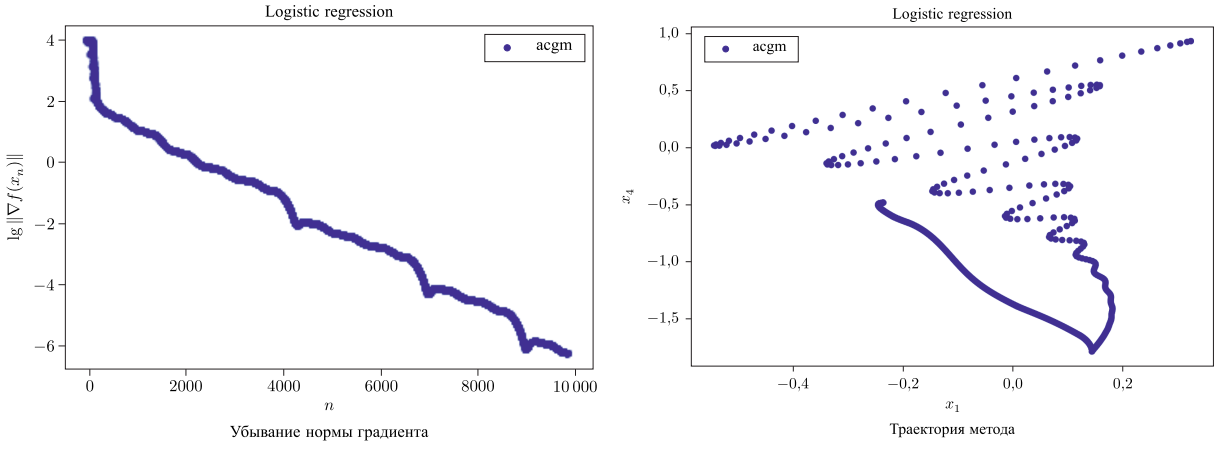

Рис. 1. Работа ACGM на функции логистической регрессии: зависимость нормы градиента от количества итераций n и проекция траектории метода на плоскость двух первых координат

Графики показывают, что метод сходится — логарифм невязки убывает приблизительно линейно по количеству итераций.

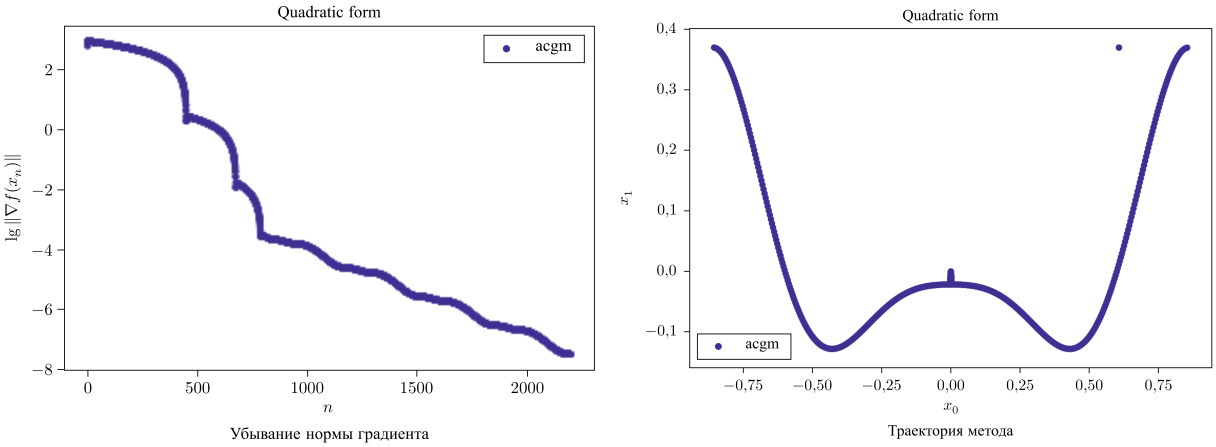

Рис. 2. Работа ACGM на квадратичной функции: зависимость нормы градиента от количества итераций n и траектория метода на координатной плоскости

5. Адаптивность по константе Липшица

5.1. Универсальный градиентный спуск — адаптивный по константе Липшица неускоренный метод

Algorithm 3. Universal Gradient Method (UGM)

Input: $f \in \mathcal{F}(Q)$, $\mathbf{x}_0 \in Q$, L_0, ε.

1: **for** $k \geqslant 0$ **do**
2: **if** $\|\nabla f(x_k)\| \leqslant \varepsilon$ **then**
3: **break**;
4: **end if**
5: $L_{k+1} := \frac{L_k}{2}$;
6: $\mathbf{x}_{k+1} := \arg\min\limits_{\mathbf{x}\in Q} \left\{ f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + L_{k+1} V(\mathbf{x}, \mathbf{x}_k) \right\}$;
7: **if** $f(\mathbf{x}_{k+1}) \leqslant f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + L_{k+1} V(\mathbf{x}_{k+1}, \mathbf{x}_k)$ **then**
8: **continue**;
9: **else**
10: $L_{k+1} := 2L_{k+1}$;
11: **goto 6**;
12: **end if**
13: **end for**

Output: $[x_0 \dots x_N]$.

В замечании 2.1 пособия [Гасников, 2019] показано, что в качестве $V(\mathbf{x}, \mathbf{y})$ подходит функция $V(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$.

Поскольку рассматривается задача безусловной оптимизации, $Q = \mathbb{R}^d$. Градиент минимизируемого выражения равен $\nabla f(\mathbf{x}_k) + L_{k+1}(\mathbf{x} - \mathbf{x}_k)$, поэтому формула шага 6 преобразуется к виду $\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{1}{L_{k+1}} \nabla f(\mathbf{x}_k)$, а условие перехода к следующему шагу — к виду $f(\mathbf{x}_{k+1}) \leqslant f(\mathbf{x}_k) - \frac{1}{2L_{k+1}} \|\nabla f(\mathbf{x}_k)\|^2$.

5.2. OGM-GL — адаптивный по константе Липшица вариант OGM-G

Универсальный градиентный спуск решает проблему неизвестности константы Липшица, но обладает недостатками простейшего градиентного метода, так как для функций с большим числом обусловленности матрицы Гессе направление градиента значительно отличается от направления на экстремум. Поэтому применение универсального градиентного метода на практике неэффективно.

Предлагается следующий вариант OGM-G, адаптивный по константе Липшица, основанный на подборе L аналогично тому, как это делается в универсальном градиентном методе, с проверкой условия строки 7 при каждом вычислении $\mathbf{y}_{i+1}$.

Если условие нарушено, то вычисление последовательностей $\mathbf{x}_i$ и $\mathbf{y}_i$ начинается сначала с тем же количеством шагов и увеличенным значением L.

Algorithm 4. OGM-G Lipschitz

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\mathbf{x}_0 \in \mathbb{R}^d$ — начальная точка, $N \geqslant 1$.

1: $L := \frac{L}{2}$;
2: $\mathbf{y}_0 := \mathbf{x}_0$;
3: **for** $i = 0, \ldots, N-1$ **do**
4: $\quad \mathbf{y}_{i+1} := \mathbf{x}_i - \frac{1}{L}\nabla f(\mathbf{x}_i)$;
5: $\quad$ **if** $f(\mathbf{y}_{i+1}) > f(\mathbf{x}_i) - \frac{1}{2L}\|\nabla f(\mathbf{x}_i)\|^2$ **then**
6: $\quad\quad L := 2L$;
7: $\quad\quad$ **goto 3**
8: $\quad$ **end if**
9: $\quad \mathbf{x}_{i+1} := \mathbf{y}_{i+1} + \beta_i(\mathbf{y}_{i+1} - \mathbf{y}_i) + \gamma_i(\mathbf{y}_{i+1} - \mathbf{x}_i)$.
10: **end for**

Output: $\mathbf{x}_N$, L_{end}.

Коэффициенты β_i, γ_i вычисляются по тем же формулам, что и в алгоритме 1.

5.3. ALGM — адаптивный по константам Липшица и сильной выпуклости алгоритм

Алгоритм построен на том же принципе, что и ACGM, только вместо OGM-G используется адаптивный по L OGM-GL. μ изменяется для сохранения отношения $\frac{L}{\mu}$ и N.

5.4. Оценки

Алгоритм подбора L гарантирует, согласно комментарию к алгоритму универсального градиентного спуска в пособии [Гасников, 2019], выполнение условия $L_{k+1} \geqslant \frac{L^{loc}}{2}$, то есть $L^{loc} \leqslant 2L_{k+1}$. Поскольку $\mathcal{F}_{L'}(\mathbb{R}^d) \subset \mathcal{F}_{L''}(\mathbb{R}^d)$ при $L' < L''$, для OGM-GL выполнены оценки сходимости, доказанные для OGM-G. Поэтому можно применять OGM-GL как составную часть алгоритма, подобного ACGM.

Один запуск OGM-GL требует не более $2N$ вычислений функции и N вычислений градиента на каждое увеличение L_k, а суммарное количество вычислений функции и градиента за один запуск составляет $O\left(N\left(\log_2 \frac{L_{end}}{L_{init}/2} + 1\right)\right)$, то есть $O\left(\sqrt{\frac{L}{\mu}}\log_2 \frac{4L_k^{end}}{L_k^{init}}\right)$, так как $L_{loc} \leqslant L$, $\mu_{loc} \geqslant L$.

Числовой множитель составляет $\sqrt{8}$ для количества вычислений градиента и $2\sqrt{8}$ для количества вычислений значения функции.

Верхняя оценка количества итераций для каждого уменьшения нормы градиента вдвое определяется аналогично вычислению из доказательства теоремы о сходимости ACGM (расчет

Algorithm 5. Adaptive by Lipschitz constant Gradient Method (ALGM)

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\mathbf{x}_0 \in \mathbb{R}^d$ — начальная точка, L_0, μ_0, β, ε.

1: **for** $k \geqslant 0$ **do**
2: **if** $\|\nabla f(\mathbf{x}_k)\| \leqslant \varepsilon$ **then**
3: **break**;
4: **end if**
5: $\mu_k := \beta\mu_{k-1}$;
6: $\mathbf{x}_k, L_k := OGMGL\left(f, \mathbf{x}_{k-1}, L_{k-1}, \left\lceil \sqrt{\frac{8L_{k-1}}{\mu_k}} \right\rceil, \varepsilon\right)$;
7: $\mu_k := \mu_k \cdot \frac{L_k}{L_{k-1}}$;
8: **if** $\|\nabla f(\mathbf{x}_k)\| \leqslant \frac{1}{2}\|\nabla f(\mathbf{x}_{k-1})\|$ **then**
9: **continue**;
10: **end if**
11: $\mu_k := \frac{\mu_k}{\beta}$;
12: **if** $\|\nabla f(\mathbf{x}_k)\| < \|\nabla f(\mathbf{x}_{k-1})\|$ **then**
13: $\mathbf{x}_{k-1} := \mathbf{x}_k$;
14: **end if**
15: **goto 6**:
16: **end for**

Output: $[\mathbf{x}_0 \dots \mathbf{x}_N]$.

для количества обращений к градиенту функции; j — номер перезапуска OGM-GL в пределах одного шага ALGM):

$$\sum_{j=0}^{J} \sqrt{8\frac{L}{\mu_k^{init}/\beta^j}}\left(2 + \log_2 \frac{L_{kj}}{L_{k,j-1}}\right) \leqslant \sqrt{8\frac{L}{\mu_k}} \sum_{j=0}^{J} \frac{1}{\sqrt{\beta^j}}\left(2 + \left(\log_2 \frac{L_k}{L_{k-1}}\right)_+\right) \lesssim$$

$$\lesssim \frac{\sqrt{8\beta}}{\sqrt{\beta}-1}\sqrt{\frac{L}{\mu_k}}\left(2 + \left(\log_2 \frac{L_k}{L_{k-1}}\right)_+\right) \leqslant \frac{\sqrt{8}\beta}{\sqrt{\beta}-1}\sqrt{\frac{L}{\mu}}\left(2 + \left(\log_2 \frac{L_k}{L_{k-1}}\right)_+\right).$$

Положительная срезка логарифма появляется для того, чтобы оценка выполнялась даже в том случае, если $L_k = \frac{L_{k-1}}{2}$. Алгоритм построен так, что за запуск OGM-GL константа Липшица может уменьшиться не более чем в два раза.

Количество шагов ALGM не превышает $K = \log_2 \frac{\|\nabla f(\mathbf{x}^0)\|}{\varepsilon}$. Поскольку для $t \geqslant -1$ выполнено свойство $(t)_+ \leqslant t + 1$, суммарное количество вычислений градиента не превосходит $\frac{\sqrt{8}\beta}{\sqrt{\beta}-1}\sqrt{\frac{L}{\mu}}\left(3K + \log_2 \frac{L}{L_0}\right)$; оценка количества вычислений функции отличается только числовым множителем и превышает полученное значение вдвое.

Как и для ACGM, числовой множитель минимален при $\beta = 4$. Таким образом, получена

Теорема 3. *Траектория ALGM* (*алгоритма* 5) *с оптимальным* $\beta = 4$ *содержит точку* $\mathbf{x}_N$, *удовлетворяющую критерию останова* (5), *после выполнения* $N \leqslant C\sqrt{\frac{L}{\mu}}\left(3K + \log_2 \frac{L}{L_0}\right)$ *вычислений градиента и* $2C\sqrt{\frac{L}{\mu}}\left(3K + \log_2 \frac{L}{L_0}\right)$ *вычислений функции, где* $K = \log_2 \frac{\|\nabla f(\mathbf{x}^0)\|}{\varepsilon}$, $C = \frac{\sqrt{8}\beta}{\sqrt{\beta}-1} = 8\sqrt{2}$.

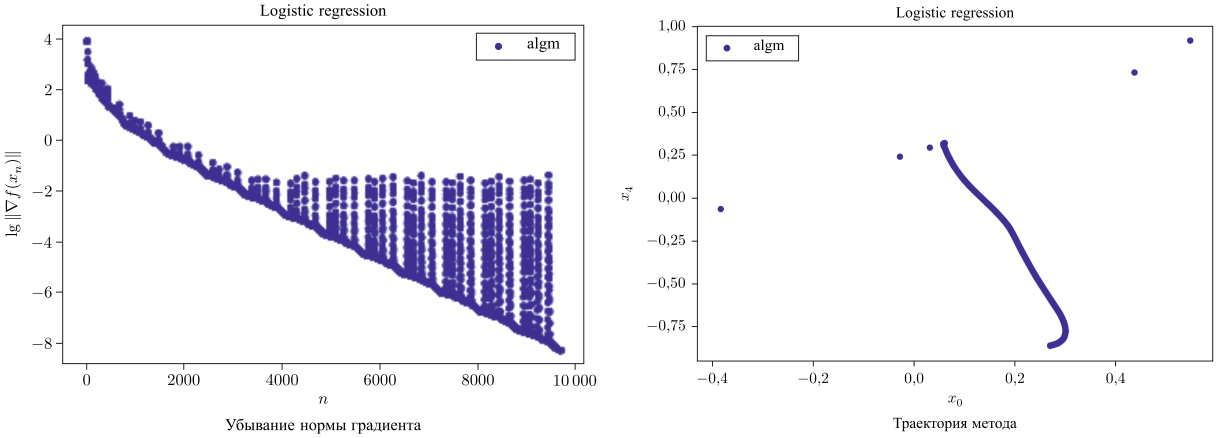

Рис. 3. Работа ALGM на функции логистической регрессии: зависимость нормы градиента от количества итераций n и проекция траектории метода на плоскость двух первых координат

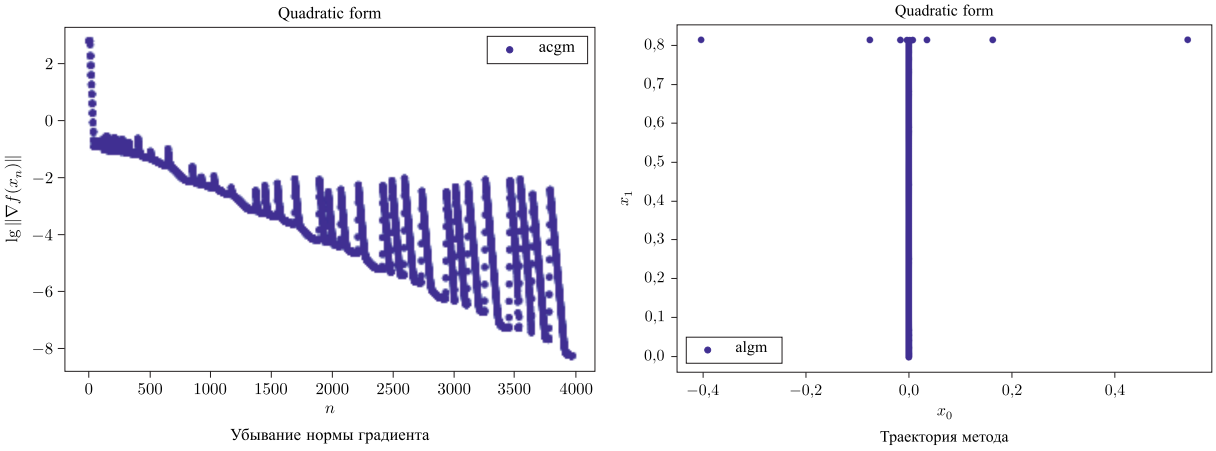

Рис. 4. Работа ALGM на квадратичной функции: зависимость нормы градиента от количества итераций n и траектория метода на координатной плоскости

5.5. Иллюстрации

Графики на рисунках 3 (для логистической регрессии) и 4 (для квадратичной формы) показывают, что нельзя говорить о сходимости в обычном смысле — логарифм невязки имеет выбросы. Они связаны с тем, что при изменении L соответствующие подпоследовательности точек, генерируемые OGM-GL, оказываются вне траектории монотонного убывания. Однако последовательность точек, порождаемая методом, имеет подпоследовательность, сходящуюся к экстремуму. Это и устанавливает теорема 3: за указанное число вызовов оракула метод гарантированно породит точку с невязкой в заданных пределах, но это не значит, что после нее все точки тоже будут иметь малую невязку.

6. Эксперименты

6.1. Тестовые функции

При решении задач бинарной классификации наиболее популярным методом является логистическая регрессия. Она сводится к следующей задаче оптимизации:

$$L(\mathbf{w}, \mathbf{X}, \mathbf{y}) = \sum_{i=1}^{N} \log(1 + \exp(-y_i \mathbf{x}_i^T \mathbf{w})) + \frac{C}{2}\|\mathbf{w}\|^2 \longrightarrow \min_{\mathbf{w}},$$

$$\mathbf{X} \in \mathbb{R}^{N \times M}, \quad \mathbf{x} \in R^M, \quad \mathbf{w} \in \mathbb{R}^M, \quad \mathbf{y} \in \{-1, 1\}^N.$$

Здесь $\mathbf{X}$ — матрица признаков, $\mathbf{y}$ — вектор ответов (его элементы — ± 1), C — параметр регуляризации.

В экспериментах $\mathbf{X} \in \mathbb{R}^{1100\times 1000}$, $\mathbf{y} \in \{-1,\ 1\}^{1100}$, $\mathbf{w}_0 \in \mathbb{R}^{1000}$ генерируются случайно. $C = 1$.

$$\frac{\partial^2 L}{\partial \mathbf{w}^2} = C\mathbf{I}_{1000} + \sum_{i=1}^{1000} \sigma(y_i \mathbf{x}_i^T \mathbf{w})(1 - \sigma(y_i \mathbf{x}_i^T \mathbf{w}))\mathbf{x}_i \mathbf{x}_i^T,$$

где $\sigma(z) = \frac{1}{1+e^z}$, поэтому функция является сильно выпуклой. Каждое слагаемое является неотрицательно определенной матрицей, а первое — положительно определенной. Поэтому константа сильной выпуклости оценивается снизу: $\mu \geqslant 1$.

Константа Липшица для градиента равна наибольшему собственному значению матрицы Гессе, и ее оценка зависит от случайных величин. Однако два множителя, содержащих $\mathbf{w}$, в произведении не превосходят $\frac{1}{4}$. Поэтому в пределах одного эксперимента значение константы Липшица не меняется.

Косвенно оценить константу Липшица позволяет тот факт, что методы, требующие ее знания, расходятся при подстановке в них $L = 10\,000$ и сходятся при подстановке $L = 100\,000$. Поэтому истинное значение превосходит 10 000, но не превосходит 100 000. Также оценка может быть получена из результатов работы ALGM или OGM-GL.

В качестве второй тестовой функции взята квадратичная форма с $L = 1000$, $\mu = 0{,}1$:

$$f(x_0,\ x_1) = 500x_0^2 + 0{,}05x_1^2.$$

6.2. Проверка теоремы о сходимости ACGM

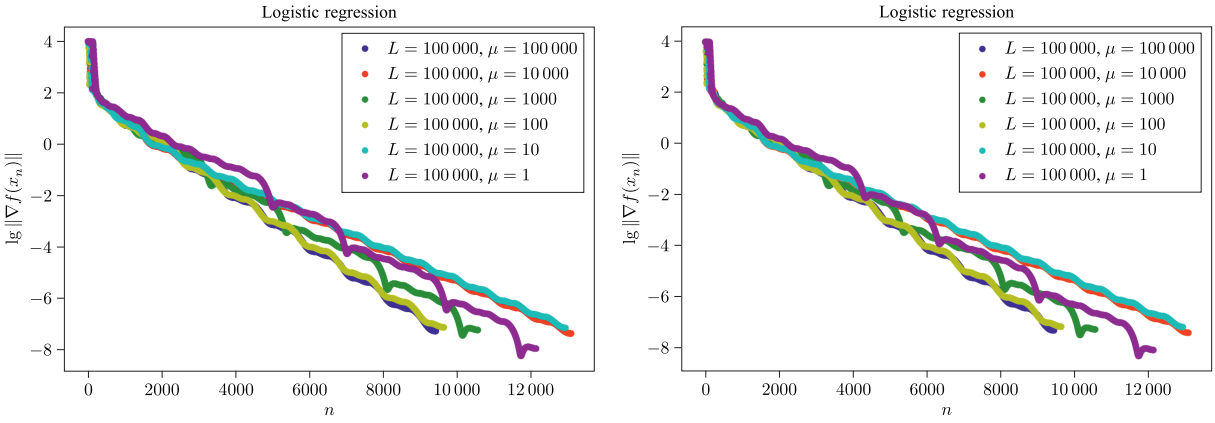

Рис. 5. Сходимость ACGM на функции логистической регрессии: убывание нормы градиента в зависимости от номера итерации n для разных затравочных значений константы сильной выпуклости μ_0

Показанные на рисунках 5 и 6 запуски с разным значением μ_0 для разных случайных значений $\mathbf{X}$, $\mathbf{y}$, $\mathbf{w}_0$ показывают приблизительно линейную скорость убывания логарифма невязки в зависимости от количества вызовов оракула. Заметные заострения на графике для $\mu = 0{,}1$ в случае квадратичной формы вызваны переходами через истинное значение — для этой функции оно как раз составляет 0,1.

Также рассматриваются квадратичные формы $\frac{1}{2}(Lx_0^2 + \mu x_1^2)$ с разными L, μ. На рисунке 7 зависимость количества вызовов оракула от константы Липшица при $\mu = 1$ отражена на левом графике, а от константы сильной выпуклости при $L = 10^6$ — на правом.

Эти графики подтверждают линейную зависимость количества вызовов от $\sqrt{\frac{L}{\mu}}$, установленную теоремой 2.

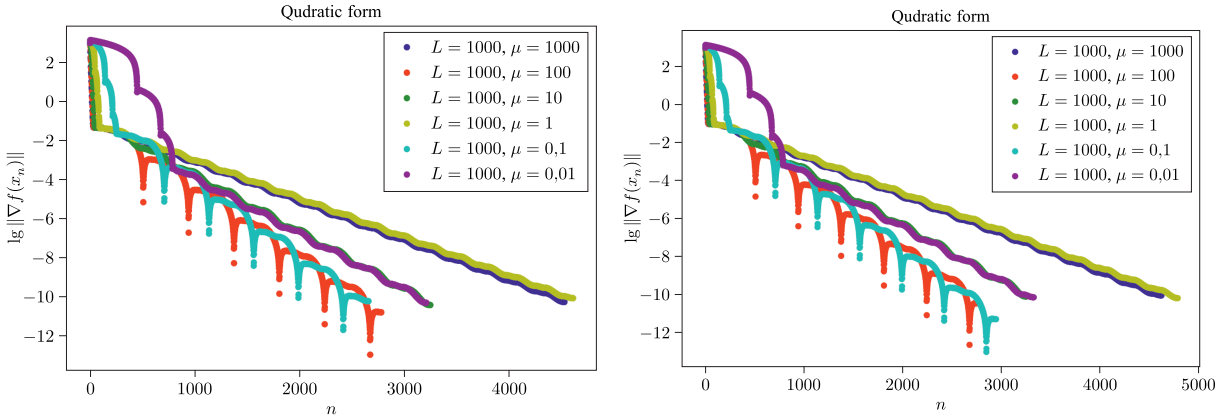

Рис. 6. Сходимость ACGM на квадратичной функции: убывание нормы градиента в зависимости от номера итерации n для разных затравочных значений константы сильной выпуклости μ_0

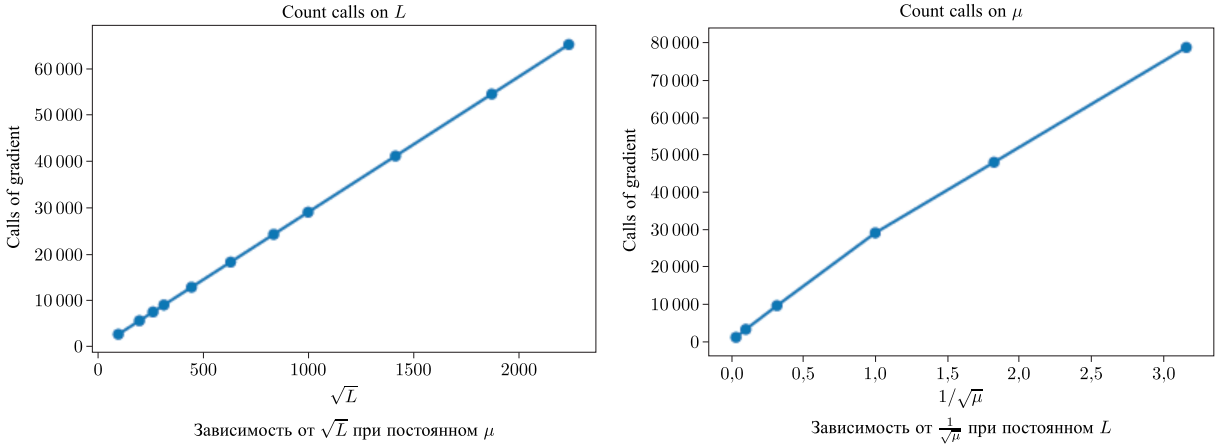

Рис. 7. Линейная зависимость количества вызовов оракула от $\sqrt{\frac{L}{\mu}}$. Теорема 2 подтверждается

6.3. Сравнение ACGM и OGM-G

OGM-G принимает количество итераций на вход, а ACGM работает до достижения условия остановки. Поэтому для сравнения эффективности используется модификация OGM-G, которая повторяет выполнение алгоритма с теми же заданными L и μ до выполнения условия остановки. Результаты сравнения показаны на рисунке 8.

Если начальное значение μ совпало с истинным, то OGM-G оказывается эффективнее ACGM, что и показывает левый график. Однако это обстоятельство не делает возможным эффективное применение OGM-G ввиду неизвестности этой константы. Если же $\mu_0 < \mu$, то, хоть оба алгоритма работают, ACGM достигает целевого значения быстрее — на правом графике.

Начальное значение $\mu_0 = L$ приводит к монотонной сходимости ACGM для обеих тестовых функций.

6.4. Проверка теоремы о сходимости ALGM

Сходимость ALGM показана на рисунках 10 и 11. Для квадратичной формы (рисунок 11) правый график построен по результатам тех же измерений, что и левый, но на меньшем диапазоне. Во всех случаях графики показывают линейное убывание минимальной достигнутой невязки с ростом числа операций.

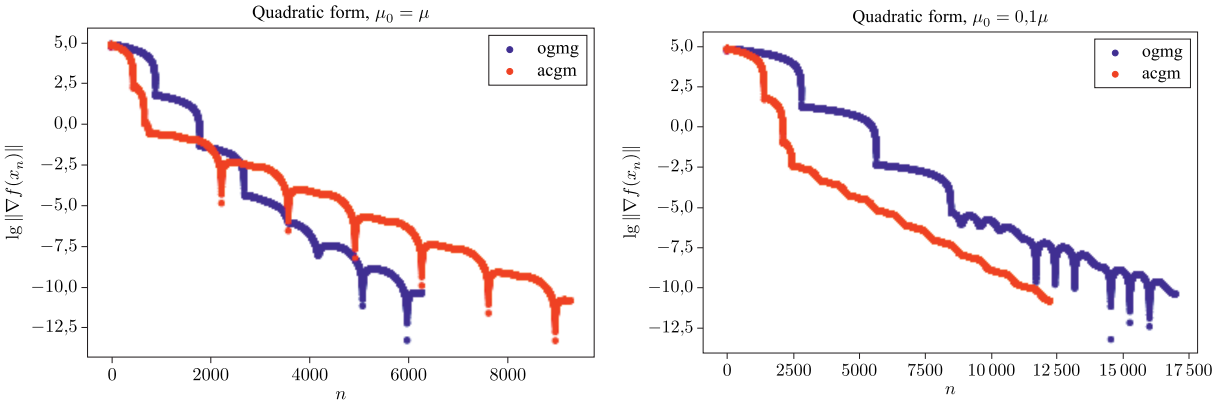

Рис. 8. Сходимость ACGM и OGM-G: убывание нормы градиента в зависимости от номера итерации n при разных затравочных значениях константы сильной выпуклости μ_0 для квадратичной функции

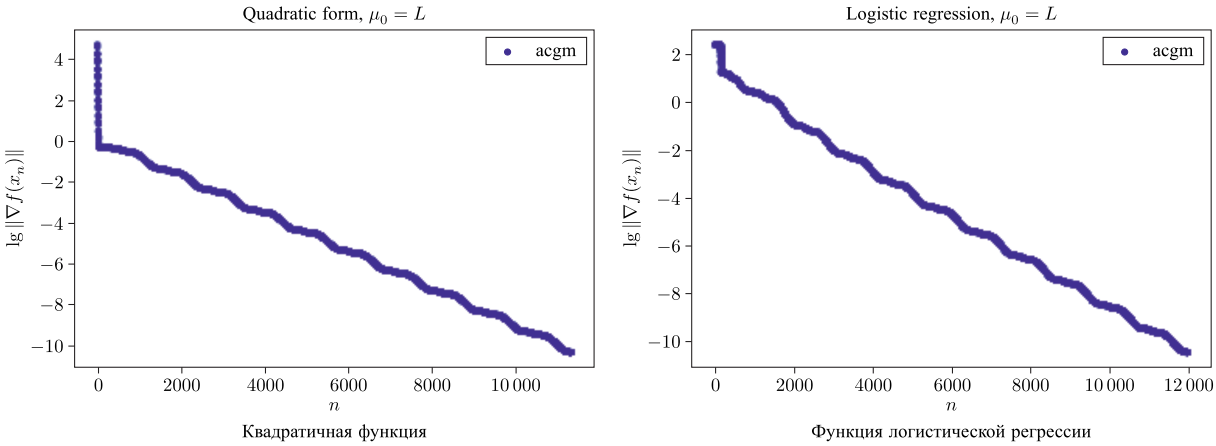

Рис. 9. Сходимость ACGM: убывание нормы градиента в зависимости от номера итерации n при затравочном значении константы сильной выпуклости $\mu_0 = L$

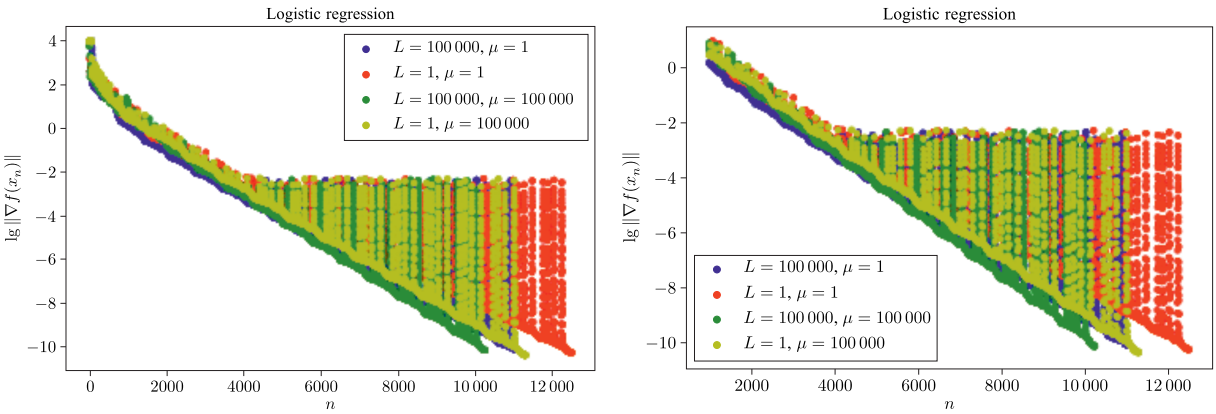

Рис. 10. Сходимость ALGM на функции логистической регрессии: убывание нормы градиента в зависимости от номера итерации n для разных затравочных значений констант Липшица L и сильной выпуклости μ_0

Также рассматриваются квадратичные формы $\frac{1}{2}(Lx_0^2 + \mu x_1^2)$ с разными L, μ. На рисунке 12 зависимость количества вызовов оракула от константы Липшица при $\mu = 1$ отражена на левом графике, а от константы сильной выпуклости при $L = 10^5$ — на правом.

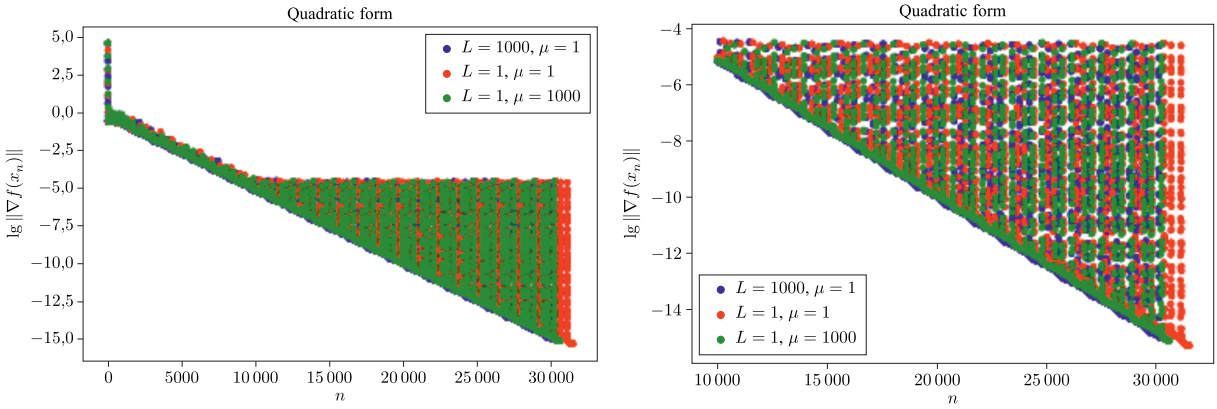

Рис. 11. Сходимость ALGM на квадратичной функции: убывание нормы градиента в зависимости от номера итерации n для разных затравочных значений констант Липшица L и сильной выпуклости μ_0

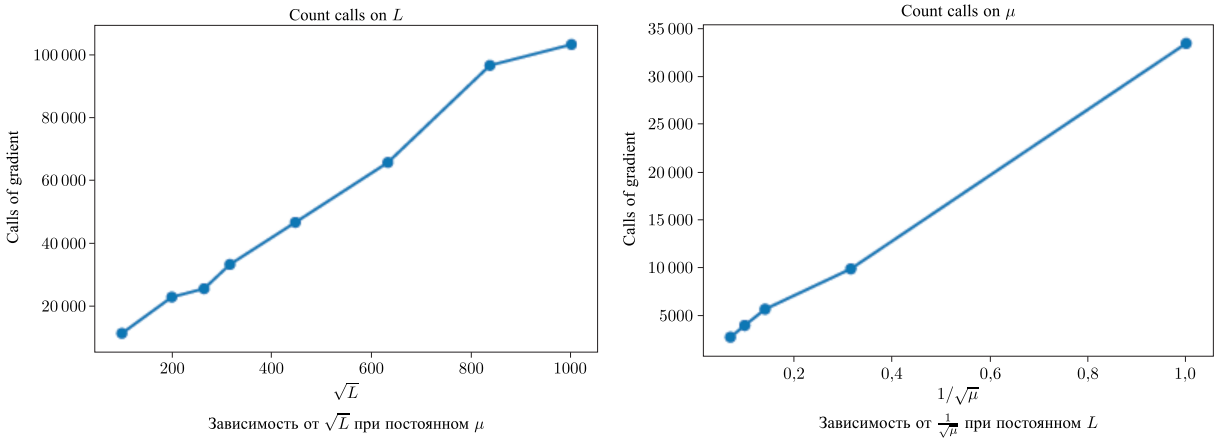

Рис. 12. Линейная зависимость количества вызовов оракула от $\sqrt{\frac{L}{\mu}}$. Теорема 3 подтверждается

Эти графики подтверждают приближенно линейную зависимость требуемого количества вызовов от $\sqrt{\frac{L}{\mu}}$, установленную теоремой 3.

6.5. Сравнение ACGM и ALGM

Из графиков на рисунке 13 видно, что ALGM сходится хоть и не монотонно, но быстрее ACGM.

7. Заключение

В работе построены адаптивные по константе сильной выпуклости и константе Липшица для градиента методы оптимизации первого порядка путем улучшения быстрого градиентного метода OGM-G.

Адаптивный по константе сильной выпуклости алгоритм ACGM имеет оптимальную оценку сходимости (см. теоремы 1 и 2). Адаптивный по константе Липшица градиента алгоритм ALGM имеет оценку сходимости, которая оптимальна с точностью до логарифмического множителя (см. теорему 3). Полученные результаты — продвижение в направлении разработки полностью адаптивных методов, не требующих на вход никаких параметров. Как правило, на практике такие методы работают лучше неадаптивных.

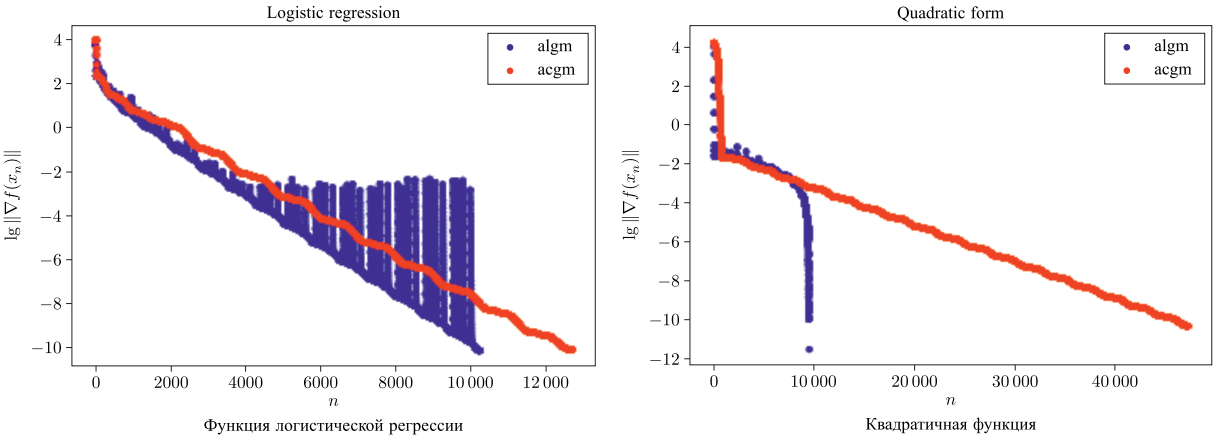

Рис. 13. Сравнение скорости сходимости ACGM и ALGM для разных функций: убывание нормы градиента в зависимости от номера итерации n

Хочу выразить особую благодарность моему научному руководителю Гасникову Александру Владимировичу, доктору физико-математических наук, профессору МФТИ, за полезные советы при подготовке данной статьи.

Список литературы (References)

Гасников А. В. Современные численные методы оптимизации. Метод универсального градиентного спуска // e-print, 2019. URL: https://arxiv.org/pdf/1711.00394.pdf
Gasnikov A. V. Sovremeniye chisleniye methody optimizacii [Universal gradient descent]. E-print. 2019. URL: https://arxiv.org/pdf/1711.00394.pdf (in Russian).

Гасников А. В., Гасникова Е. В., Нестеров Ю. Е., Чернов А. В. Об эффективных численных методах решения задач энтропийно-линейного программирования // Журнал выч. математики и мат. физики. — 2016. — Т. 56, № 4.
Gasnikov A. V., Gasnikova E. V., Nesterov Yu. E., Chernov A. V. Ob effectivnyh chislennyh metodah resheniya zadach entropiyno-lineinogo programmirovaniya [About effective computational methods of solution problems of entropial-linear programming] // ZhVM & MF [Comp. Math. & Math. Phys.]. — 2016. — Vol. 56, no. 4 (in Russian).

Нестеров Ю. Е. Введение в выпуклую оптимизацию. — М.: МЦНМО, 2010. — 262 с.
Nesterov Yu. E.. Vvedeniye v vypukluyu optimizatsiyu [Introductory lectures on convex optimization]. — Moscow: MCCME, 2010. — 262 p. (in Russian).

Нестеров Ю. Е. Эффективные методы в нелинейном программировании. — М.: Радио и связь, 1989. — 301 с.
Nesterov Yu. E.. Effectivnyye metody v nelineynom programmirovanii [Effective methods in non-linear pogramming]. — Moscow: Radio and communication, 1989. — 301 p. (in Russian).

Barre M., Taylor A., d'Aspremont A. Complexity Guarantees for Polyak Steps with Momentum // Proceedings of Thirty Third Conference on Learning Theory. — 2020. — Vol. 125. — P. 452–478. — URL: https://proceedings.mlr.press/v125/barre20a.html

Fercoq O., Qu Zh. Restarting accelerated gradient methods with a rough strong convexity estimate. E-print. 2016. URL: https://arxiv.org/pdf/1609.07358.pdf

Kim D., Fessler J. A. Optimizing the Efficiency of First-order Methods for Decreasing the Gradient of Smooth Convex Functions. E-print. 2018. URL: https://arxiv.org/pdf/1803.06600v2.pdf

Lei L., Jordan M. I. On the Adaptivity of Stochastic Gradient-Based Optimization. E-print. 2019. URL: https://arxiv.org/pdf/1904.04480v2.pdf

Nesterov Yu. E.. Universal gradient methods for convex optimization problems // Math. Program. — 2015. — Vol. 152. — P. 381–404. — https://doi.org/10.1007/s10107-014-0790-0