

УДК 519.85

УСКОРЕННЫЕ ПРОКСИМАЛЬНЫЕ ОБОЛОЧКИ: ПРИМЕНЕНИЕ К ПОКОМПОНЕНТНОМУ МЕТОДУ¹⁾

© 2022 г. А. С. Аникин¹, В. В. Матюхин^{2,*}, Д. А. Пасечнюк²

¹ 664033 Иркутск, ул. Лермонтова, 134, а/я 292, Институт динамики систем и теории управления им. В.М. Матросова Сибирского отделения РАН, Россия

² 141701 Долгопрудный, М.о., Институтский пер., 9, Московский физико-технический институт (национальный исследовательский университет), Россия

*e-mail: vladmatyukh@gmail.com

Поступила в редакцию 16.02.2021 г.
Переработанный вариант 16.02.2021 г.
Принята к публикации 04.08.2021 г.

Статья посвящена одному частному случаю применения универсальных ускоренных проксимальных оболочек для получения вычислительно эффективных ускоренных вариантов методов, использующихся для решения различных частных постановок оптимизационных задач. В данной работе предлагается проксимально ускоренный покомпонентный градиентный метод с эффективной алгоритмической сложностью итерации, позволяющий существенно учитывать разреженность решаемой задачи, и рассматривается пример применения предлагаемого подхода для решения задачи оптимизации функции вида SoftMax, для которой описываемый метод позволяет ослабить зависимость вычислительной сложности решения от размерности n задачи в $\mathcal{O}(\sqrt{n})$ раз, и демонстрирует на практике более быструю по сравнению со стандартными методами сходимость. Библи. 22. Фиг. 2.

Ключевые слова: проксимальный ускоренный метод, каталист, ускоренный покомпонентный метод, SoftMax, LogSumExp.

DOI: 10.31857/S004446692202003X

1. ВВЕДЕНИЕ

Одним из важнейших теоретических результатов в выпуклой оптимизации является разработка ускоренных методов оптимизации [1]. На начальном этапе развития этой концепции было предложено множество ускоренных вариантов различных методов, применяющихся к решению многих задач выпуклой оптимизации, однако каждый такой случай требовал отдельного, частного рассмотрения возможности ускорения, ввиду чего предлагаемые конструкции были существенно различны и не позволяли предполагать способ их обобщения. Важным шагом к разработке универсальной схемы ускорения методов оптимизации стали работы, в которых предлагался и исследовался метод, названный каталист, основанный на идее ускоренного проксимального градиентного метода [2], [3] и позволяющий ускорять другие методы оптимизации, используя их для последовательного решения ряда регуляризованных по Моро-Иосиде вспомогательных задач [4]–[6]. В продолжение этих идей в дальнейшем было предложено множество вариантов применения данного метода и его модификаций [7]–[9]. Среди наиболее свежих, на момент написания данной статьи, результатов были также описаны обобщения обсуждаемого подхода на тензорные методы [10]–[13]. Соответствующее представление ускоренной проксимальной оболочки, если опираться на известные авторам сведения, является наиболее общим из описанных в литературе, и потому в данной работе внимание будет обращено прежде всего именно на методы, предложенные в последней из цитируемых выше работ.

¹⁾ Исследования Д.А. Пасечнюка были поддержаны стипендией А.М. Райгородского в области оптимизации и при финансовой поддержке РФФИ (код проекта 19-31-51001) Научное наставничество. Исследования А.С. Аникина выполнены при финансовой поддержке РФФИ (код проекта 18-29-03071 мк). Работа В.В. Матюхина выполнена при поддержке Минобрнауки РФ (госзадание) № 075-00337-20-03, номер проекта 0714-2020-0005.

Основной мотив данной работы состоит в том, чтобы описать возможности практического применения универсальных ускоренных проксимальных оболочек для конструирования вычислительно и оракульно-эффективных методов оптимизации. Рассмотрим классический покомпонентный метод [14], итерация которого для выпуклой функции $f : \mathbb{R}^n \rightarrow \mathbb{R}$ имеет вид:

$$x_{k+1}^i = x_k^i - \eta \nabla_i f(x_k), \quad i \sim \mathcal{U}\{1, \dots, n\}, \quad \eta > 0.$$

Одним из многих приложений данного метода является оптимизация функционалов, вычисление одной компоненты градиента которых значимо эффективнее, чем вычисление полного вектора градиента; в частности, многие задачи в случае разреженных постановок удовлетворяют данному условию. Однако оракульная сложность данного метода при условии остановки метода при достижении ε -малости невязки по значению функции составляет $\mathcal{O}\left(n \frac{\bar{L} R^2}{\varepsilon}\right)$, где $R^2 = \|x_0 - x_*\|_2^2$, $\bar{L} = \frac{1}{n} \sum_{i=1}^n L_i$ есть среднее констант Липшица компонент градиента, притом эта оценка не является оптимальной для класса выпуклых задач. Рассмотрим теперь ускоренный покомпонентный метод, предложенный Ю.Е. Нестеровым [15]. То есть оракульная сложность данного метода соответствует оптимальной оценке: $\mathcal{O}\left(n \sqrt{\frac{\bar{L} R^2}{\varepsilon}}\right)$, где $\sqrt{\bar{L}} = \frac{1}{n} \sum_{i=1}^n \sqrt{L_i}$ – среднее квадратных корней из констант Липшица компонент градиента. Вместе с тем ситуация кардинально меняется в случае рассмотрения алгоритмической сложности метода: пусть даже вычисление одной компоненты градиента имеет сложность $\mathcal{O}(s)$, $s \ll n$, сложность итерации ускоренного покомпонентного метода будет составлять $\mathcal{O}(n)$, в отличие от стандартного метода, сложность итерации которого есть $\mathcal{O}(s)$, – содержательно это означает, что степень разреженности задачи при применении ускоренного покомпонентного метода не влияет существенно на сложность алгоритма, и, кроме того, сложность в таком случае квадратично зависит от размерности задачи: вместе это в некоторой степени обесценивает применение покомпонентного метода в данном случае. Таким образом, интересной задачей является построение ускоренного покомпонентного метода, сложность итерации которого, как и в стандартном варианте метода, составляет $\mathcal{O}(s)$, при сохранении оптимальной оракульной сложности – в данной работе это удается осуществить благодаря применению универсальной ускоренной проксимальной оболочки “ускоренный метаалгоритм” [13].

Данная статья состоит из двух основных разделов. В разд. 2 описываются теоретические результаты о сходимости и алгоритмической сложности покомпонентного метода, ускоренного путем применения оболочки “ускоренный метаалгоритм”. В разд. 3 на примере задачи оптимизации функционала вида SoftMax экспериментально проверяется эффективность метода в отношении времени его работы, описываются возможности его вычислительно-эффективной имплементации и осуществляется сравнение со стандартными методами.

2. ТЕОРЕТИЧЕСКИЕ ГАРАНТИИ

Рассмотрим задачу оптимизации вида $\min_{x \in \mathbb{R}^n} f(x)$ функции $f : \mathbb{R}^n \rightarrow \mathbb{R}$, удовлетворяющей следующим условиям:

- 1) f дифференцируема на \mathbb{R}^n ;
- 2) f выпукла на \mathbb{R}^n ;
- 3) $\nabla_i f$ удовлетворяет условию Липшица:

$$\exists L_i \in \mathbb{R} : |\nabla_i f(x + ue_i) - \nabla_i f(x)| \leq L_i |u|, \quad \forall x \in \mathbb{R}^n, \quad u \in \mathbb{R},$$

где e_i есть i -й орт базиса, $i \in \{1, \dots, n\}$;

4) ∇f удовлетворяет условию Липшица с константой L .

Листинг 1. Ускоренный метаалгоритм для метода \mathcal{M}

Вход: $H > 0, x_0 \in \mathbb{R}^n$;
 $\lambda \leftarrow 1/2H$;
 $A_0 \leftarrow 0; v_0 \leftarrow x_0$;
for $k = 0, \dots, \tilde{N} - 1$ **do**
 $a_{k+1} \leftarrow \frac{\lambda + \sqrt{\lambda^2 + 4\lambda A_k}}{2}$;
 $A_{k+1} \leftarrow A_k + a_{k+1}$;
 $\tilde{x}_k \leftarrow \frac{A_k v_k + a_{k+1} x_k}{A_{k+1}}$;
 Посредством запуска метода \mathcal{M}
 найти с точностью ε по аргументу
 решение вспомогательной задачи:
 $v_{k+1} \in \text{Arg} \min_{y \in \mathbb{R}^n} \left\{ f(y) + \frac{H}{2} \|y - \tilde{x}_k\|_2^2 \right\}$;
 $x_{k+1} \leftarrow x_k - a_{k+1} \nabla f(v_{k+1})$;
end
return $v_{\tilde{N}}$.

Обратимся к содержанию работы [13], где предложен общий вариант “ускоренного метаалгоритма” решения задач выпуклой оптимизации для композитных функционалов вида $F(x) = f(x) + g(x)$. Для рассматриваемой постановки задачи такая общность не требуется, достаточно применить частный случай описанной схемы при $p = 1, f \equiv 0$ (используются обозначения соответствующей работы), в котором описанная оболочка принимает вид ускоренного проксимального метода. Псевдокод используемого метода представлен в листинге 1.

Листинг 2. Покомпонентный метод

Вход: $y_0 \in \mathbb{R}^n$;
 $Z \leftarrow \sum_{i=1}^n (H + L_i)$;
 $p_i \leftarrow (H + L_i)/Z, i \in \{1, \dots, n\}$;
Дискретное вероятностное
распределение π с вероятностями p_i ;
for $k = 0, \dots, N - 1$ **do**
 $i \sim \pi\{1, \dots, n\}$;
 $y_{k+1} \leftarrow y_k$;
 $y_{k+1}^i = y_k^i - \frac{1}{H + L_i} \nabla_i F(y_k)$;
end
return y_N .

Прежде чем сформулировать какие-либо результаты о сходимости, следует подробнее рассмотреть вопрос о решении вспомогательной задачи — ее аналитическое решение доступно лишь в редких случаях, и потому необходимо решать ее численными методами, а значит, неточно.

Вспомогательную задачу допустимо решать до выполнения следующего условия останова ([16], Appendix B):

$$\left\| \nabla \left\{ F(y_*) := f(y_*) + \frac{H}{2} \|y_* - \tilde{x}_k\|_2^2 \right\} \right\| \leq \frac{H}{2} \|y_* - \tilde{x}_k\|_2. \quad (1)$$

Ввиду того, что $\|\nabla F(y_*)\|_2 = 0$, а также ввиду $(L + H)$ -липшицевости ∇F , имеем

$$\|\nabla F(y_*)\|_2 \leq (L + H) \|y_* - y_k\|_2. \quad (2)$$

Выписав неравенство треугольника: $\|\tilde{x}_k - y_*\|_2 - \|y_* - y_k\|_2 \leq \|y_* - \tilde{x}_k\|_2$, и воспользовавшись неравенствами (1), (2), получаем окончательное условие останова:

$$\|y_* - y_k\|_2 \leq \frac{H}{3H + 2L} \|\tilde{x}_k - y_*\|_2. \quad (3)$$

Содержательно отсюда следует, что необходимая точность решения вспомогательной задачи по аргументу не зависит от требуемой точности решения общей задачи, что позволяет значимо упростить получение дальнейших результатов.

Рассмотрим теперь основной применяемый для решения вспомогательных задач метод: содержание покомпонентного метода [17] (в частном случае $\gamma = 1$) представлено в листинге 2. Для данного метода в случае рассматриваемых вспомогательных задач справедлива

Теорема 2.1 (см. [14, теорема 6.8]). Пусть F является H -сильно выпуклой относительно $\|\cdot\|_2$. Тогда для последовательности $\{y_k\}_{k=1}^N$, генерируемой покомпонентным методом, выполняется

$$\begin{aligned} \mathbb{E}[F(y_N)] - F(y_*) &\leq \left(1 - \frac{1}{\kappa}\right)^N (F(y_0) - F(y_*)), \\ \kappa &= \frac{H}{Z}, \quad Z = \sum_{i=1}^n (H + L_i). \end{aligned} \quad (4)$$

Используя данный результат, сформулируем утверждение о числе итераций покомпонентного метода, достаточном для выполнения полученного выше условия останова.

Следствие 2.1. Математическое ожидание $\mathbb{E}[y_N]$ точки, являющейся результатом работы покомпонентного метода, удовлетворяет условию (3) достижения достаточной точности решения вспомогательной задачи ускоренного метаалгоритма в том случае, если для числа итераций метода выполнено

$$\begin{aligned} N \geq N(\tilde{\epsilon}) &= \left\lceil \frac{Z}{H} \ln \left\{ \left(1 + \frac{L}{H}\right) \left(3 + \frac{2L}{H}\right)^2 \right\} \right\rceil, \\ \tilde{\epsilon} &= \frac{H}{2} \left(\frac{H}{3H + 2L} \right)^2 \|y_0 - y_*\|_2^2. \end{aligned}$$

Доказательство. Ввиду $(H + L)$ -липшицевой гладкости функции F верно:

$$F(y_0) - F(y_*) \leq \frac{H + L}{2} \|y_0 - y_*\|_2^2.$$

Используя это неравенство вместе с оценкой (4), можем выписать условие достижения заданной точности $\tilde{\epsilon}$ по функции:

$$\frac{H + L}{2} \left(1 - \frac{1}{\kappa}\right)^N \|y_0 - y_*\|_2^2 \leq \tilde{\epsilon}.$$

Также верно $1 - 1/\kappa \leq \exp\{-1/\kappa\}$, и, значит:

$$\frac{H + L}{2} \exp\{\kappa/N\} \|y_0 - y_*\|_2^2 \leq \tilde{\epsilon}.$$

Логарифмируя и подставляя выражение для κ , получаем выражение для числа итераций от $\tilde{\varepsilon}$:

$$N(\tilde{\varepsilon}) = \left\lceil \frac{Z}{H} \ln \left\{ \frac{(H+L)\|y_0 - y_*\|_2^2}{2\tilde{\varepsilon}} \right\} \right\rceil. \quad (5)$$

Ввиду H -сильной выпуклости функции F верно $\|\bar{y}_N - y_*\|_2^2 \leq \frac{2}{H}(F(\bar{y}_N) - F(y_*))$, где $\bar{y}_N = \mathbb{E}[y_N]$. Функция F выпукла, следовательно, по неравенству Йенсена, $F(\bar{y}_N) \leq \mathbb{E}[F(y_N)]$, и отсюда, вместе с (3) получаем достаточное условие достижения решения вспомогательной задачи:

$$\mathbb{E}[F(y_N)] - F(y_*) \leq \frac{H}{2} \left(\frac{H}{3H+2L} \right)^2 \|y_0 - y_*\|_2^2.$$

Подставляя в формулу (5) вместо $\tilde{\varepsilon}$ выражение из правой части данного неравенства, непосредственно приходим к выражению из утверждения.

Теперь, когда полностью прояснен вопрос о требуемой точности и оракульной сложности решения вспомогательной задачи с помощью покомпонентного метода, можно перейти к результатам о сходимости ускоренного метаалгоритма. Для используемого условия останова (3) метода, решающего вспомогательную задачу, справедлив следующий результат о сходимости ускоренного метаалгоритма:

Теорема 2.2 (см. [13, теорема 1]). *При $H > 0$ для последовательности $\{v_k\}_{k=1}^{\tilde{N}}$, генерируемой ускоренным метаалгоритмом, использующим для решения вспомогательной задачи некоторый не стохастический метод, выполняется*

$$f(v_{\tilde{N}}) - f(x_*) \leq \frac{48}{5} \frac{H \|x_0 - x_*\|_2^2}{\tilde{N}^2}. \quad (6)$$

На основании последнего утверждения можно сформулировать теорему о сходимости ускоренного метаалгоритма в случае применения стохастического метода и, в частности, покомпонентного градиентного спуска.

Теорема 2.3. *При $H > 0$ для некоторого $0 < \delta < 1$ точка $v_{\tilde{N}}$, являющаяся результатом работы ускоренного метаалгоритма, использующего для решения вспомогательной задачи покомпонентный метод, решающий вспомогательную задачу N_δ итераций, удовлетворяет условию*

$$\Pr(f(v_{\tilde{N}}) - f(x_*) < \varepsilon) \geq 1 - \delta$$

в случае, если

$$N_\delta \geq N \left(\frac{\tilde{\varepsilon}\delta}{\tilde{N}} \right) = \left\lceil \frac{Z}{H} \ln \left\{ \frac{\tilde{N}}{\delta} \left(1 + \frac{L}{H} \right) \left(3 + \frac{2L}{H} \right)^2 \right\} \right\rceil, \quad (7)$$

$$\tilde{N} \geq \left\lceil \frac{4\sqrt{15}}{5} \sqrt{\frac{H \|x_0 - x_*\|_2^2}{\varepsilon}} \right\rceil.$$

Доказательство. В следствии 2.1 представлена оценка числа итераций, достаточного для выполнения следующего условия на математическое ожидание значения функционала в результирующей точке метода:

$$\mathbb{E}[F(y_{N(\tilde{\varepsilon})})] - F(y_*) \leq \tilde{\varepsilon}.$$

Воспользуемся неравенством Маркова и получим формулировку данного условия в терминах оценки вероятности больших отклонений [18]: заведомо выберем допустимое значение вероятности невыполнения поставленного условия, так чтобы $0 < \delta/\tilde{N} < 1$, где \tilde{N} выражается из (6); тогда

$$\Pr(F(y_{N(\tilde{\varepsilon}/\tilde{N})}) - F(y_*) \geq \tilde{\varepsilon}) \leq \frac{\delta}{\tilde{N}} \frac{\mathbb{E}[F(y_{N(\tilde{\varepsilon}/\tilde{N})})] - F(y_*)}{\tilde{\varepsilon} \cdot \delta/\tilde{N}} = \frac{\delta}{\tilde{N}}.$$

Поскольку вероятность того, что полученное решение некоторой отдельно взятой вспомогательной задачи не будет удовлетворять поставленному условию, равна δ/\tilde{N} , значит, вероятность то-

го, что за \tilde{N} итераций ускоренного метаалгоритма условие будет не выполнено хотя бы для одной из задач, есть $\tilde{N} \cdot \delta / \tilde{N} = \delta$, откуда и следует доказываемое утверждение.

Далее, объединяя оценки, приводимые в теореме 9, можем получить асимптотическую оценку на общее число итераций покомпонентного метода, достаточное для решения рассматриваемой оптимизационной задачи с некоторой заданной точностью, а также оценку оптимального значения параметра H :

Следствие 2.2. Для того чтобы точка $v_{\tilde{N}}$, являющаяся результатом работы ускоренного метаалгоритма, удовлетворяла условию

$$Pr(f(v_{\tilde{N}}) - f(x_*) < \epsilon) \geq 1 - \delta,$$

достаточно выполнить в сумме

$$\hat{N} \geq \tilde{N} \cdot N_\delta = \mathcal{O}\left(\frac{Z \|x_0 - x_*\|_2}{\sqrt{H}} \frac{1}{\epsilon^{1/2}} \log\left\{\frac{1}{\epsilon^{1/2}\delta}\right\}\right) \tag{8}$$

итераций покомпонентного метода для решения вспомогательной задачи. При этом оптимально значение параметра H регуляризации вспомогательной задачи следует выбирать как $H \simeq \frac{1}{n} \sum_{i=1}^n L_i$ (\simeq обозначает равенство с точностью до малого множителя порядка \log).

Доказательство. Выражение для \hat{N} можно получить путем прямой подстановки одной из оценок, приводимых в (7), в другую, и их последующего умножения. Если исключить из рассмотрения малый множитель порядка $\log(L/H)$, константа в оценке будет зависеть от H как:

$$\sqrt{H} \cdot \frac{Z/n}{H} = \sqrt{H} \left(1 + \frac{\frac{1}{n} \sum_{i=1}^n L_i}{H}\right).$$

Минимизируя представленное выражение по H , получаем указанный результат.

Рассмотрим теперь подробнее вопрос об алгоритмической сложности предложенного ускоренного метода покомпонентного градиентного спуска. Очевидным является

Утверждение 2.1. Алгоритмическая сложность рассматриваемого метода составляет

$$T = \mathcal{O}(\tilde{N}(T_{\text{out}} + N_\delta T_{\text{inn}})),$$

где T_{out} – амортизированная оценка сложности вычислений, производимых на итерации ускоренного метаалгоритма, T_{inn} – амортизированная оценка сложности итерации покомпонентного метода.

Оттолкнувшись от него, сформулируем результат о вычислительной сложности метода.

Теорема 2.4. Пусть сложность вычисления одной компоненты градиента f составляет $\mathcal{O}(s)$. Тогда алгоритмическая сложность рассматриваемого метода есть

$$T = \mathcal{O}\left(sn \cdot \sqrt{\frac{\bar{L} \|x_0 - x_*\|_2^2}{\epsilon}} \log\left\{\frac{1}{\epsilon^{1/2}\delta}\right\}\right), \quad \bar{L} = Z/n = \frac{1}{n} \sum_{i=1}^n L_i.$$

Доказательство. Переформулируем оценку из утверждения 2.1 следующим образом:

$$T = \mathcal{O}(\hat{N} \cdot T_{\text{iter}}),$$

где T_{iter} – амортизированная оценка сложности элементарной итерации ускоренного метаалгоритма, т.е. итерации, которая может быть и внутренней итерацией покомпонентного метода, и основной итерацией метаалгоритма. Сложность основной итерации метаалгоритма определяется в первую очередь вычислением полного градиента f , а сложность этой процедуры (из условия теоремы) есть $\mathcal{O}(sn)$. В то же время, ввиду $Z = n\bar{L}$, верно также $N_\delta = \tilde{\mathcal{O}}(n)$, где символ $\tilde{\mathcal{O}}(\cdot)$ означает то же, что и $\mathcal{O}(\cdot)$, но с возможным присутствием множителей порядка $\log(\cdot)$. Поскольку основная итерация метаалгоритма выполняется через каждые N_δ элементарных итераций, где N_δ постоянно, из этого с помощью любого из методов амортизационного анализа тривиально получается амортизированная оценка сложности основной итерации метаалгоритма, составляющая $\mathcal{O}(s)$. Сложность итерации покомпонентного метода (если вместо копирования значений точки вы-

Таблица 1. Сравнение эффективности методов

Метод	Итеративная сложность	Вычислительная сложность	Источник
FGM	$\mathcal{O}(sn)$	$\mathcal{O}\left(sn \cdot \frac{1}{\varepsilon^{1/2}} \cdot \sqrt{L}\right)$	[1]
CDM	$\mathcal{O}(s)$	$\mathcal{O}\left(sn \cdot \frac{1}{\varepsilon} \cdot \bar{L}\right)$	[14]
ACDM	$\mathcal{O}(n)$	$\mathcal{O}\left(n^2 \cdot \frac{1}{\varepsilon^{1/2}} \cdot \sqrt{\bar{L}}\right)$	[15]
Catalyst CDM	$\mathcal{O}(s)$	$\tilde{\mathcal{O}}\left(sn \cdot \frac{1}{\varepsilon^{1/2}} \cdot \sqrt{\bar{L}}\right)$	данная работа

полнять операции “на месте”, что для данной конструкции вполне допустимо) определяется вычислением одной компоненты градиента, и составляет также $\mathcal{O}(s)$. Отсюда получаем $T_{\text{iter}} = \mathcal{O}(s)$. Используя оценку (8) и подставляя оптимальное значение H , получаем приведенную сложность метода.

Заметим также, что сложность метода по памяти при этом составляет $\mathcal{O}(n)$, также как и сложность предварительных вычислений (для покомпонентного метода нет необходимости выполнять их каждый раз заново).

Сравним полученные для предложенного метода оценки с оценками других методов, которые могут быть использованы для решения задач в описываемой постановке: быстрого градиентного метода (FGM), классического покомпонентного спуска (CDM), ускоренного покомпонентного спуска в варианте Ю.Е. Нестерова (ACDM) и предложенного в данной работе подхода (Catalyst CDM). Оценки приведены в табл. 1. Как можно видеть из приведенных асимптотических оценок вычислительной сложности, предложенный метод позволяет достигать эффективной в отношении характера зависимости от размерности задачи n и требуемой точности ε скорости сходимости, не уступающей другим методам, при некоторой плате за это в виде логарифмического множителя в оценке.

Заметим, кроме того, что несмотря на существенное сходство оценок, между двумя наиболее эффективными методами в табл. 1 (FGM и Catalyst CDM) существует также различие в константах, характеризующих гладкость функции: L – в FGM и \bar{L} – в Catalyst CDM, тем самым поведение рассматриваемого метода для различных задач напрямую зависит от характера их покомпонентной гладкости. В общем случае нельзя утверждать, что одна из констант асимптотически существенно выгоднее другой, однако в ряде частных случаев возможно выписать оценки значений констант явно, и часто оказывается, что \bar{L} “меньше” L . Наиболее существенен выигрыш в том случае, если справедливы соотношения $L = \mathcal{O}(n)$, $\bar{L} = \mathcal{O}(1)$ – тогда в оценке вычислительной сложности предлагаемого метода удастся редуцировать фактор порядка $\mathcal{O}(\sqrt{n})$ по сравнению с быстрым градиентным методом. В следующем разделе будет рассмотрен пример постановки задачи, в которой данный случай имеет место.

3. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

В данном разделе описывается характер практического поведения метода, на примере следующей оптимизационной задачи для функционала, имеющего вид SoftMax-функции:

$$\min_{x \in \mathbb{R}^n} \left\{ f(x) = \gamma \ln \left(\sum_{j=1}^m \exp \left(\frac{[Ax]_j}{\gamma} \right) \right) - \langle b, x \rangle \right\}, \quad (9)$$

где $b \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$. Задачи такого вида существенно важны для многих приложений, в частности, они возникают в задачах энтропийно-линейного программирования в качестве двойственной задачи [4], [22], в том числе в задаче оптимального транспорта, а также исполняют роль сглаженной аппроксимации функции \max (что и дало функционалу название SoftMax) и, соответ-

ственно, нормы $\|\cdot\|_\infty$, что может быть востребовано в некоторых постановках задачи PageRank или при решении системы линейных уравнений. Притом во всех описанных задачах важным частным случаем являются разреженные постановки, в случае которых матрица A разрежена, т.е. такова, что среднее число ненулевых элементов в строке A_j не превосходит некоторого $s \ll n$ (будет удобно также предполагать возможность для одной из строк A_j являться полностью неразрезанной).

Сформулируем свойства, которыми обладает функция f [19]:

- 1) f дифференцируема;
- 2) ∇f удовлетворяет условию Липшица с константой $L = \max_{j=1,\dots,m} \|A_j\|_2^2$;
- 3) $\nabla_i f$ удовлетворяют покомпонентному условию Липшица с константами $L_i = \max_{j=1,\dots,m} |A_{ji}|$.

Начнем уточнение свойств с первого пункта. Выпишем выражение для i -й компоненты градиента данного функционала:

$$\nabla_i f(x) = \frac{\sum_{j=1}^m A_{ji} \exp([Ax]_j)}{\sum_{j=1}^m \exp([Ax]_j)}.$$

Как можно видеть, наивное вычисление этого выражения может занимать время, сравнимое с вычислением градиента в целом, что будет значительно влиять на вычислительную сложность, а значит, и на время работы метода. Однако в то же время многие члены в этом выражении могут перевычисляться либо редко, либо покомпонентно, и использоваться при совершении шага метода как члены его дополнительной последовательности, так что сложность итерации будет оставаться эффективной, и применение покомпонентного метода будет оправдано. Для удобства описания используемых вычислительных приемов запишем шаг покомпонентного метода в виде

$$y_{k+1} = y_k + \delta e_i,$$

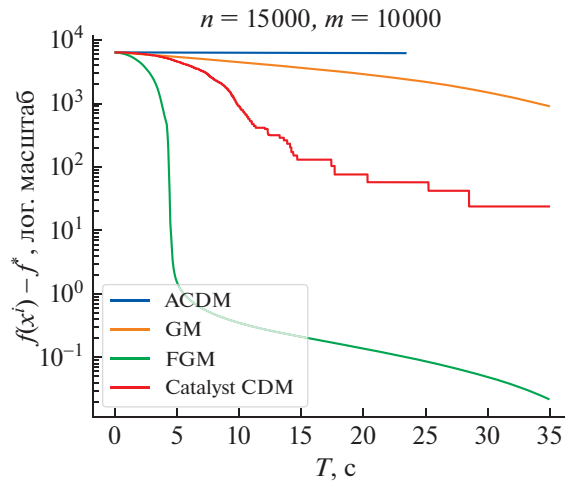
где δ – размер шага, умноженный на соответствующую компоненту градиента, e_i есть i -й орт базиса.

1. Будем хранить набор значений $\{\exp([Ay_k]_j)\}_{j=1}^m$, используемых для вычисления суммы в числителе. Обновление этих значений после осуществления шага метода имеет сложность $\mathcal{O}(s)$, ввиду того что $Ay_{k+1} = Ay_k + \delta A_i$, и вместе с тем A_i имеет не более s ненулевых компонент, а значит, потребуется вычисление s корректирующих множителей и умножение на них соответствующих значений из набора.

2. Как можно видеть из п. 1, производить умножение разреженных векторов следует за $\mathcal{O}(s)$, учитывая лишь ненулевые компоненты. В смысле программной реализации это означает необходимость использования для кэшируемых значений и для строк матрицы A разреженного представления, т.е. хранение лишь пар индекс-значение ненулевых элементов. Тогда, очевидно, сложность арифметических операций для таких векторов будет пропорциональна сложности цикла с элементарными арифметическими операциями, число итераций которого равно числу ненулевых элементов (в языке программирования python, например, такой формат хранения реализуется в методе `scipy.sparse.csr_matrix` [20]).

3. Аналогично, будем хранить значение $\sum_{j=1}^m \exp([Ay_k]_j)$, являющееся знаменателем. Его обновление осуществляется с той же сложностью, что и обновление последовательности из п. 1 (путем вычисления суммы ненулевых слагаемых, прибавляемых к каждому значению из набора).

4. Поскольку вычисление указанного выражения требует вычисления значений экспонент, может происходить переполнение типов. Для решения этой проблемы стандартно применяется `exp-normalize trick` [21]. Однако для его применения следует также хранить значение $\max_{j=1,\dots,m} [Ay_k]_j$. Вместе с тем нет необходимости знать именно это значение, или, иначе, знать его точно – достаточно лишь его приближения, чтобы значения в показателях экспонент были малы, так что перевычислять эту величину можно гораздо реже: например, раз в m итераций метода, в результате чего амортизированная сложность будет равна $\mathcal{O}(s)$.



Фиг. 1. Сходимость методов для задачи SoftMax (9) с равномерно разреженной случайной матрицей.

Итак, в дальнейших рассуждениях можно полагать, что итерация покомпонентного метода при решении соответствующей вспомогательной задачи имеет амортизированную сложность $\mathcal{O}(s)$.

Далее, рассмотрим подробнее вопрос о величине констант гладкости данного функционала.

Можно выписать асимптотические формулы для L и $\bar{L} = \frac{1}{n} \sum_{i=1}^n L_i$:

$$L = \max_{j=1, \dots, m} \|A_j\|_2^2 = \mathcal{O}(n), \quad \bar{L} = \frac{1}{n} \sum_{i=1}^n \max_{j=1, \dots, m} |A_{ji}| = \mathcal{O}(1).$$

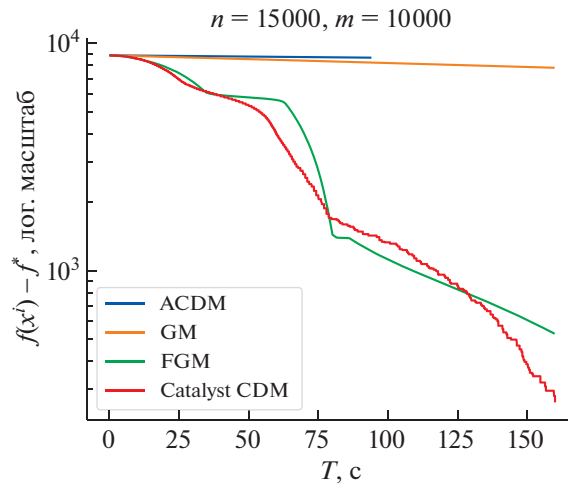
Используя эти оценки, уточним вычислительную сложность методов FGM и Catalyst CDM в применении к данной задаче:

$$T_{\text{FGM}} = \mathcal{O}\left(sn^{3/2} \cdot \frac{1}{\epsilon^{1/2}}\right), \quad T_{\text{CCDM}} = \tilde{\mathcal{O}}\left(sn \cdot \frac{1}{\epsilon^{1/2}}\right).$$

Таким образом, в теории, применение метода Catalyst CDM для решения данной задачи позволяет, по сравнению с FGM, редуцировать в асимптотической оценке вычислительной сложности множитель порядка $\mathcal{O}(\sqrt{n})$. Практически, это означает, что предложенный метод разумно применять для задач большой размерности.

Сравним теперь работу предложенного в статье метода (Catalyst CDM) с рядом альтернативных подходов: градиентным спуском (GM), быстрым градиентным методом (FGM), покомпонентным спуском (CDM) и ускоренным покомпонентным спуском (ACDM), на примере задачи (9) с искусственно сгенерированной двумя различными способами матрицей A . На фиг. 1 и 2 представлены графики сходимости рассматриваемых методов: по оси абсцисс отложено время работы методов в секундах, по оси ординат — невязка по функции в логарифмическом масштабе (f_* находится путем поиска соответствующей точки x_* с помощью метода FGM, настроенного на точность, заведомо значительно превосходящую возможную для достижения на выбранном временном промежутке).

На фиг. 1 представлен случай, для которого все элементы матрицы A являются нормально распределенными случайными величинами из дискретного равномерного распределения $A_{ji} \in \mathcal{U}\{0, 1\}$, при этом число ненулевых элементов составляет $s \approx 0.2m$, и параметр $\gamma = 0.6$ (так же как и во втором случае). В такой постановке предложенный метод демонстрирует более быструю сходимость по сравнению со всеми сравниваемыми методами, за исключением FGM. В то же время, в постановке, отраженной на фиг. 2, при которой число ненулевых элементов, по сравнению с первым случаем, увеличено до $s \approx 0.75m$, а матрица генерируется неравномерно в соответствии с правилом: $0.9m$ строк с $0.1n$ ненулевых элементов и $0.1m$ строк с $0.9n$ ненулевых элементов, а также одна и строк матрицы является полностью неразрезанной, предложенный метод



Фиг. 2. Сходимость методов для задачи SoftMax (9) с неоднородно разреженной случайной матрицей.

сходится быстрее FGM. Это объясняется тем, что в этом случае $L = n$, тогда как \bar{L} по-прежнему остается достаточно мало, в результате чего константа в предложенном методе оказывает заметно меньшее влияние на вычислительную сложность, чем в случае FGM. Из результатов эксперимента также можно отметить, что гораздо существеннее степени разреженности задачи на эффективность предложенного метода влияет характер ее покомпонентной гладкости.

4. ЗАКЛЮЧЕНИЕ

В данной работе предлагается вариант покомпонентного метода, ускоренного с помощью универсальной проксимальной оболочки “ускоренный метаалгоритм”. Проведенный теоретический анализ предложенного метода позволяет утверждать, что зависимость его вычислительной сложности от размерности задачи и требуемой точности решения не уступает прочим методам, используемым для оптимизации выпуклых липшицево-гладких функций, а оценка вычислительной сложности сравнима с оценкой быстрого градиентного метода. При этом в предложенной схеме сохраняются свойства, характерные для классического покомпонентного метода, в том числе возможность использования свойств покомпонентной гладкости функции. Приведенные численные эксперименты подтверждают практическую эффективность метода, и также подчеркивают особенную релевантность предложенного подхода для часто возникающей в различных приложениях задачи оптимизации функции вида SoftMax.

Данная статья представляет результат работы над проектом, предложенным А.В. Гасниковым в рамках проектной смены (Ссылка на сайт проектной смены: <https://sochisirius.ru/obucheniye/graduates/smena673/3258>) “Современные методы теории информации, оптимизации и управления” Сириус 2–23 августа 2020 г. Авторы выражают благодарность организатору проектной смены А.С. Ненашеву за создание комфортных условий для работы.

СПИСОК ЛИТЕРАТУРЫ

1. *Nesterov Y.E.* Lectures on convex optimization // Springer. 2018. Volume 137.
2. *Parikh N., Boyd S.* Proximal algorithms // Foundations and Trends in Optimization. 2014. V. 1. № 3. P. 127–239.
3. *Rockafellar R.T.* Monotone operators and the proximal point algorithm // SIAM Journal on Control and Optimization. 1976. V. 14. № 5. P. 877–898.
4. *Чернов А.В.* Прямо-двойственный метод решения задачи энтропийно-линейного программирования // Интеллектуальные системы. Теория и приложения. 2016. Т. 20. № 1. С. 39–59.
5. *Lin H., Mairal J., Harchaoui Z.* A universal catalyst for first-order optimization // Advances in Neural Information Processing Systems. 2015. V. 28. P. 3384–3392.
6. *Lin H., Mairal J., Harchaoui Z.* Catalyst acceleration for first-order convex optimization: from theory to practice // The Journal of Machine Learning Research. 2017. V. 18. № 1. P. 7854–7907.

7. *Ivanova A., Pasechnyuk D., Grishchenko D., Shulgin E., Gasnikov A., Matyukhin V.* Adaptive catalyst for smooth convex optimization // arXiv preprint arXiv:1911.11271, 2019.
8. *Kulunchakov A., Mairal J.* A generic acceleration framework for stochastic composite optimization // *Advances in Neural Information Processing Systems*. 2019. P. 12556–12567.
9. *Paquette C., Lin H., Drusvyatskiy D., Mairal J., Harchaoui Z.* Catalyst acceleration for gradient-based non-convex optimization // arXiv preprint arXiv:1703.10993, 2017
10. *Bubeck S., Jiang Q., Lee Y.T., Li Y., Sidford A.* Near-optimal method for highly smooth convex optimization // *Conference on Learning Theory*. 2019. P. 492–507.
11. *Doikov N., Nesterov Y.* Contracting proximal methods for smooth convex optimization // *SIAM Journal on Optimization*. 2020. V. 30. № 4. P. 3146–3169.
12. *Monteiro R., Svaiter B.F.* An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods // *SIAM Journal on Optimization*. 2013. V. 23. № 2. P. 1092–1125.
13. *Гасников А.В., Двинских Д.М., Двуреченский П.Е., Камзолов Д.И., Матюхин В.В., Пасечнюк Д.А., Тупица Н.К., Чернов А.В.* Ускоренный метаалгоритм для задач выпуклой оптимизации // *Ж. вычисл. матем. и матем. физ.* 2021. Т. 61 № 1. С. 20–31.
14. *Bubeck S.* Convex optimization: Algorithms and complexity // arXiv preprint arXiv:1405.4980, 2014.
15. *Nesterov Y., Stich S.U.* Efficiency of the accelerated coordinate descent method on structured optimization problems // *SIAM Journal on Optimization*. 2017. V. 27. № 1. P. 110–123.
16. *Kamzolov D., Gasnikov A., Dvurechensky P.* On the optimal combination of tensor optimization methods // arXiv preprint arXiv:2002.01004, 2020.
17. *Nesterov Yu.* Efficiency of coordinate descent methods on huge-scale optimization problems // *SIAM Journal on Optimization*. 2012. V. 22. № 2. P. 341–362.
18. *Anikin A., Dvurechensky P., Gasnikov A., Golov A., Gornov A., Maximov Yu., Mendel M., Spokoiny V.* Modern efficient numerical approaches to regularized regression problems in application to traffic demands matrix calculation from link loads // *Proc. of International Conference ITAS-2015*. 2015.
19. *Гасников А.В.* Современные численные методы оптимизации. Метод универсального градиентного спуска. МФТИ, 2018. С. 181.
20. Python Scipy documentation: https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csr_matrix.html
21. *Blanchard Pierre, Higham Desmond J., Higham Nicholas J.* Accurately Computing the Log-Sum-Exp and Softmax Functions // *MIMS Preprint*. 2019.
22. *Гасников А.В., Гасникова Е.В., Нестеров Ю.Е., Чернов А.В.* Об эффективных численных методах решения задач энтропийно-линейного программирования // *Ж. вычисл. матем. и матем. физ.* 2016. Т. 56. № 4. С. 523–534.