# Ускоренный метод Нестерова для децентрализованной распределённой оптимизации на меняющихся со временем сетях

***А. В. Рогозин***[1]
aleksandr.rogozin@phystech.edu
[1]Московский физико-технический институт (государственный университет), г. Долгопрудный

В работе исследуется сходимость методов первого порядка в случае, когда целевая функция меняется во время работы метода. Изменяющаяся во времени целевая функция возникает при рассмотрении задач распределённой децентрализованной оптимизации в случае, когда граф вычислителей периодически меняется (причиной этому могут быть технические неполадки, такие как потеря связи между двумя вычислительными узлами). Основным результатом работы является получение теоретических оценок скорости сходимости для методов градиентного спуска и ускоренного метода Нестерова. Эти результаты получены в предположении, что минимизируемая функция является сильно выпуклой и имеет липшицев градиент, а граф, соответствующий сети вычислительных узлов, меняется конечное число раз.

**Ключевые слова**: *распределённая оптимизация; ускоренный метод; time-varying graph*

**DOI:** 10.21469/22233792.4.5.02

## 1 Введение

Рассмотрим задачу машинного обучения с вектором параметров $y \in \mathbb{R}^d$ и функцией потерь $L(\mathbf{A}, y)$, где $A$ – обучающая выборка. Предположим, что выборка $\mathbf{A}$ не может находиться в памяти одного компьютера из-за своего размера, и поэтому разделена на $n$ частей $\{\mathbf{A_i}\}_{i=1}^n$ и размещена на $n$ различных машинах. Соответствующая задача минимизации эмпирического риска принимает вид

$$L(\mathbf{A}, y) = \sum_{i=1}^{n} L(\mathbf{A_i}, y) \longrightarrow \min_{y \in \mathbb{R}^d}. \tag{1}$$

Так как выборка не может находится в памяти одной машины, нужно модифицировать известные алгоритмы оптимизации так, чтобы они могли работать не на одном компьютере, а на некоторой сети вычислительных узлов.

В дальнейшем будем рассматривать задачи вида

$$\varphi(y) = \sum_{i=1}^{n} \varphi_i(y) \longrightarrow \min_{y \in \mathbb{R}^d}. \tag{2}$$

Здесь каждое $\varphi_i$ есть выпуклая функция, которая может вычисляется на отдельном компьютере. Пусть некоторые машины могут передавать информацию друг другу. Это позволяет построить граф $\mathcal{G} = (V, E)$ для сети компьютеров, на которой ведутся вычисления (вершины графа соответствуют узлам сети, рёбра – связям между узлами). Под *децентрализованным* алгоритмом оптимизации понимается метод, при котором каждый узел под номером $i$ работает только с $\varphi_i$ и обменивается информацией только со своими соседями (то есть со смежными вершинами), но при этом в результате исполнения алгоритма получается некоторый вектор $y \in \mathbb{R}^d$, близкий к решению задачи (2).

Из-за происходящих в сети неполадок, таких как разрыв связи между двумя компьютерами, граф $\mathcal{G}$ может меняться. В работе будут рассматриваться алгоритмы, работающие на последовательности графов $\{\mathcal{G}_k\}_{k=1}^{\infty} = \{(V, E_k)\}_{k=1}^{\infty}$. Предполагается, что все графы $\mathcal{G}_k$ являются связными.

## 2 Постановка задачи

### 2.1 Статичная сеть вычислителей

Сначала рассмотрим случай, когда сеть вычислителей не меняется со временем. Пусть сеть задаётся графом $\mathcal{G} = (V, E)$, где вершины $V$ соответствуют компьютерам, а рёбра $E$ – возможностям обмена информацией между компьютерами. При этом каждое $\varphi$ в задаче (2) хранится на отдельном узле. Задача (2) может быть переписана в эквивалентном виде

$$\sum_{i=1}^{n} \varphi_i(y_i) \longrightarrow \min \;\; s.t.\; y_1 = \ldots = y_n,\; y_i \in \mathbb{R}^d \;\forall i \in V \tag{3}$$

Такая формулировка имеет следующий смысл: каждому узлу выдаётся локальная копия вектора $y$ и накладываются ограничения на то, чтобы все эти локальные копии были одинаковыми. Введём лапласиан графа $\mathcal{G}$:

$$[W]_{ij} = \begin{cases} -1, & \text{если } (i,j) \in E, \\ \deg(i), & \text{если } i = j, \\ 0, & \text{иначе.} \end{cases}$$

Если граф $\mathcal{G}$ является неориентированным и связным, то можно показать, что $W$ является неотрицательно определённой симметричной матрицей, а также

$$\ker(W) = \ker(\sqrt{W}) = \operatorname{span}(\mathbf{1}_n)$$

Обозначим $Y = [y_1, ..., y_n]$ и получим, что условие $y_1 = ... = y_n$ эквивалентно ограничению $YW = 0$. Поэтому задачу (3) можно переписать в виде

$$\Phi(Y) = \sum_{i=1}^{n} \varphi_i(y_i) \longrightarrow \min_{Y\sqrt{W}=0}. \tag{4}$$

Заметим, что $Y$ является матрицей размера $d \times n$, а $y$ – это вектор $\mathbb{R}^d$. Распределённый алгоритм основан на рассмотрении задачи, двойственной к (4):

$$f(X) = \max_{Y \in \mathbb{R}^{d\times n}} \big[ -\Phi(Y) - \langle X, Y\sqrt{W}\rangle \big] = \max_{Y \in \mathbb{R}^{d\times n}} \big[ -\Phi(Y) - \langle Y, X\sqrt{W}\rangle \big], \tag{5}$$
$$f(X) \longrightarrow \min_{X \in \mathbb{R}^{d\times n}}.$$

Для удобства обозначим $Z = Z(X) = -X\sqrt{W} \in \mathbb{R}^{d\times n},\; Z = [z_1 \;...\; z_n]$ и получим

$$\begin{aligned} f(X) &= \max_{Y \in \mathbb{R}^{d\times n}} \Big( \langle Y, Z\rangle - \Phi(Y) \Big) \\ &= \max_{Y \in \mathbb{R}^{d\times n}} \Big[ \sum_{i=1}^{n} \big( \langle y_i, z_i\rangle + \varphi_i(y_i) \big) \Big] \\ &= \sum_{i=1}^{n} \max_{y_i \in \mathbb{R}^d} \Big( \langle y_i, z_i\rangle - \varphi_i(y_i) \Big). \end{aligned} \tag{6}$$

Также введём обозначения

$$\begin{aligned} Y(X) &= \arg\max_{Y\in\mathbb{R}^{d\times n}} \Big[ -\langle X, Y\sqrt{W}\rangle - \Phi(Y)\Big], \\ Z &= -X\sqrt{W}, \\ \tilde{Y}(Z) &= \arg\max_{Y\in\mathbb{R}^{d\times n}} \Big[\langle Z, Y\rangle - \Phi(Y)\Big], \\ \tilde{Y}(Z) &= \big(\tilde{y}_1(z_1), ..., \tilde{y}_n(z_n)\big). \end{aligned}$$

и сразу заметим, что

$$\tilde{Y}(Z) = \tilde{Y}(-X\sqrt{W}) = Y(X).$$

Также заметим, что

$$f(X) = \Phi^*(Z)$$

где $\Phi^*$ – сопряжённая по Фенхелю функция к $\Phi$.

Подсчитать градиент $f(X)$ можно по формуле Демьянова-Данскина [2–4]:

$$\nabla f(X) = -Y(X)\sqrt{W} = -\tilde{Y}(-X\sqrt{W})\sqrt{W} = -\tilde{Y}(Z)\sqrt{W}.$$

Знание градиента $f(X)$ в задаче (6) позволяет применять методы первого порядка. Рассмотрим для примера градиентный спуск:

$$\begin{aligned} X^{k+1} &= X^k - \alpha\nabla f(X), \\ X^{k+1} &= X^k + \alpha Y(X^k)\sqrt{W}. \end{aligned}$$

Переходя к обозначениям $Z^k$, получаем

$$Z^{k+1} = Z^k - \alpha\tilde{Y}(Z^k)W. \tag{7}$$

Рассмотрим, как вычисляется $\tilde{Y}(Z)$ более подробно.

$$\begin{aligned} \tilde{Y}(Z) &= \arg\max_{Y\in\mathbb{R}^{d\times n}} \Big[\langle Z, Y\rangle - \Phi(Y)\Big] \\ &= \arg\max_{y_1\in\mathbb{R}^d,\ldots,y_n\in\mathbb{R}^d} \left[\sum_{i=1}^n \langle z_i, y_i\rangle - \varphi_i(y_i)\right], \\ \tilde{y}_i(z_i) &= \arg\max_{y\in\mathbb{R}^d} \left[\langle z_i, y_i\rangle - \varphi_i(y_i)\right]. \end{aligned}$$

Из последнего выражения следует, что $i$-ый столбец $\tilde{Y}(Z)$ может вычисляться локально на $i$-ом узле. Пусть $z_i$ (т.е. $i$-ый столбец $Z$) хранится на $i$-ом узле. Тогда обновление $z_i^k$ согласно правилу (7) на $i$-ом узле происходит по формуле

$$z_i^{k+1} = z_i^k - \alpha\left(\tilde{Y}(Z^k)W\right)_i.$$

Заметим, что в силу структуры матрицы $W$ $i$-ому узлу нужно получить величины $\tilde{y}_i(z_i^k)$ только от узлов, являющихся его соседями. Иными словами, для того, чтобы сделать

градиентный шаг (7), каждому компьютеру нужно обменяться информацией *только со своими соседями*. В этом и заключается децентрализованный алгоритм.

### 2.2 Свойства двойственнной задачи

Так как децентрализованное задачи (4) основано на запуске децентрализованного метода оптимизации на двойственной функции $f(X)$, заданной в (5), то для анализа сходимости понадобится знание свойств $f(X)$, таких как константа сильной выпуклости и константа Липшица для градиента. В работе [9] получен результат, связывающий сильную выпуклость и гладкость функции и её сопряжённой по Фенхелю. Сформулируем этот результат в качестве леммы.

**Лемма 1.** Пусть $f$ – замкнутая сильно выпуклая функция. Тогда $f$ является $\mu$-сильно выпуклой относительно $\|\cdot\|$ тогда и только тогда, когда $f^*$ является $L$-гладкой (т.е. имеет градиент с константой Липшица $L$) относительно $\|\cdot\|_*$.

Следующая теорема обобщает результаты леммы 1 на матрицы.

**Теорема 1.** Пусть $\sigma_{\max}$ – наибольшее, а $\tilde{\sigma}_{\min}$ – наименьшее ненулевое сингулярные числа матрицы $W$, являющейся Лапласианом графа $\mathcal{G}$. Предположим, что $\Phi(Y)$ является $L_\Phi$-гладкой и $\mu_\Phi$-сильно выпуклой относительно нормы Фробениуса $\|\cdot\|_F$. Тогда $f(X) = \max\limits_{Y\in\mathbb{R}^{d\times n}}\left(-\langle X\sqrt{W}, Y\rangle - \Phi(Y)\right)$ является сильно выпуклой с константой $\mu_f = \frac{\sqrt{\tilde{\sigma}_{\min}(W)}}{L_\Phi}$ на подпространстве $(\ker W)^\perp$ и имеет липшицев градиент с константой $L_f = \frac{\sqrt{\sigma_{\max}(W)}}{\mu_\Phi}$ на всём $\mathbb{R}^{d\times n}$.

### 2.3 Переменная во времени сеть

Наконец, перейдём к специфичной задаче, которой посвящена данная работа. Предположим, что граф $\mathcal{G}$ время от времени меняется, причём набор вершин остаётся неизменным, а рёбра могут появляться и исчезать. При этом предполагается, что граф $\mathcal{G}$ остаётся связным. Таким образом, мы приходим к последовательности графов $\{\mathcal{G}_k\}_{k=1}^{\infty}$.

Двойственная задача (5) зависит от матрицы $W$. Так как граф вычислительной сети меняется, вместе с ним будет меняться $W$, и, следовательно, двойственная функция $f(X)$. А именно, мы будем иметь дело с последовательностью функций

$$f_k(X) = \Phi^*(-X\sqrt{W_k}) = \max_{Y\in\mathbb{R}^{d\times n}}\left(-\left\langle X, Y\sqrt{W_k}\right\rangle - \Phi(Y)\right). \tag{8}$$

Рассмотрим, как будет выглядеть градиентный спуск в этом случае. Аналогично 7 получаем

$$Z^{k+1} = Z^k - \alpha\nabla f_k(X_k)$$
$$Z^{k+1} = Z^k - \alpha\tilde{Y}(Z^k)W_k$$

Таким образом, после запуска градиентного метода целевая функция может меняться, и $k$-ый шаг будет производится для функции $f_k(X)$. Основной результат работы описывает, как работают градиентный спуск и ускоренный метод Нестерова в случае, когда целевая функция время от времени изменяется.

## 3 Результаты

В этом разделе нам понадобится

**Определение 1.**

$$\theta_{\max} = \max_{k \geqslant 0} \{\sigma_{\max}(W_k)\}, \tag{9a}$$

$$\theta_{\min} = \min_{k \geqslant 0} \{\tilde{\sigma}_{\min}(W_k)\}. \tag{9b}$$

Заметим, что связных графов на $n$ вершинах имеется конечное число, поэтому максимум и минимум, фигурирующие в (9), определены корректно, причём $\theta_{\max} < \infty$, $\theta_{\min} > 0$. Также сформулируем основное предположение относительно функции $\Phi(Y)$.

**Предположение 1.** Функция $\Phi(Y)$ является $\mu_\Phi$-сильно выпуклой и $L_\Phi$-гладкой.

Заметим, что матрица $W$, задающая ограничения, меняется от итерации к итерации, но в каждый момент времени $YW_k = 0$ равносильно $y_1 = ... = y_n$ (т.е. меняется способ задания ограничений, но не само множество ограничений). Значит, функция $f$ меняется, но её точка минимум и минимальное значение остаются постоянными из-за сильной двойственности. Учитывая это и теорему 1, получаем

**Утверждение 1.** Пусть выполнено предположение 1. Все функции $\{f_k\}_{k=1}^{\infty}$ являются сильно выпуклыми с константой $\mu_f$ на подпространстве $(\ker W)^\perp$, имеют липшицев градиент с константой $L_f$ на всём $\mathbb{R}^{d \times n}$, а также общую точку минимума $X^*$ и минимальное значение $f^*$, где

$$\mu_f = \frac{\sqrt{\theta_{\min}(W)}}{L_\Phi}, \tag{10a}$$

$$L_f = \frac{\sqrt{\theta_{\max}(W)}}{\mu_\Phi}. \tag{10b}$$

### 3.1 Градиентный спуск

Рассмотрим, как работает неускоренный градиентный спуск на меняющейся со временем функции $f$:

$$X^{k+1} = X^k - \frac{1}{L_f} \nabla f_k(X^k) \tag{11}$$

где $L_f$ определено в (10).

**Теорема 2.** Пусть $\{X^k\}$ – последовательность, генерируемая градиентным спуском (11), и предположение 1 выполняется. Тогда для всякого $k > 0$:

$$\|X^{k+1} - X^*\|_2 \leqslant e^{-\frac{\mu_f}{L_f} k} \|X^0 - X^*\|_2.$$

### 3.2 Ускоренный метод Нестерова

В этом разделе рассмотрим ускоренный метод Нестерова для сильно выпуклых задач:

$$Y^{k+1} = X^k - \frac{1}{L_f} \nabla f_k(X^k), \tag{12a}$$

$$X^{k+1} = \left(1 + \frac{\sqrt{\varkappa_f} - 1}{\sqrt{\varkappa_f} + 1}\right) Y^{k+1} - \frac{\sqrt{\varkappa_f} - 1}{\sqrt{\varkappa_f} + 1} Y^k, \tag{12b}$$

Следующий результат гарантирует линейную сходимость данного метода при определённых условиях.

**Теорема 3.** Пусть $\{f_k(X)\}_{k=1}^{\infty}$ – последовательность функций, для которых выполнено предположение 1. Кроме того, пусть изменения графа происходят в моменты $n_1 < ... < n_m$. Тогда последовательность $Y^k$, генерируемая методом Нестерова (12), имеет следующее свойство: для всякого $N > n_m$ выполнено

$$f_N(Y^N) - f^* \leqslant \left(\frac{L_f}{\mu_f}\right)^m \frac{L_f + \mu_f}{2} \frac{R^2}{(1+\gamma_f)^N},$$

где $\gamma_f = \frac{1}{\sqrt{\varkappa_f}-1}$ and $\|X^0 - X^*\|_2 \leqslant R$.

## 4 Заключение

Децентрализованная оптимизация на меняющихся со временем сетях приводит к возникновению переменной во времени целевой функции. Поведение существующих методов оптимизации в этом случае может меняться. В данной работе было теоретически установлено, что для градиентного спуска линейная сходимость сохраняется вне зависимости от числа изменений вычислительного графа. Для ускоренного метода Нестерова была доказана линейная сходимость в случае, когда граф меняется не более, чем конечное число раз.

## Литература


[1] *Nikaido Hukukane* Convex Structures and Economic Theory. / Academic Press, 1968

[2] *Danskin John* The Theory of Max-Min and its Applications to Weapons Allocation Problems. / Springer-Verlag, 1967

[3] *Демьянов В., Малоземов В.* Введение в минимакс / Наука, 1972

[4] *Bertsekas D.* Convex Optimization Theory / Athena Scientific, 2009

[5] *Bansal N., Anupam G.* Potential-Function Proofs for First-Order Methods // arXiv:1712.04581, 2007

[6] *Nedić A., Ozdaglar A.* Cooperative distributed multi-agent optimization. // Convex Optimization in Signal Processing and Communications, 2009

[7] *Bach F., Scaman K., Bubeck S., Lee Y.T. and Massoulié L* Optimal algorithms for smooth and strongly convex distributed optimization in networks // arXiv:1702.08704, 2017

[8] *Гасников А.* Универсальный градиентный спуск // arXiv:1711.00394

[9] *Kakade S.M., Shalev-Shwartz S., Tewari A.* On the duality of strong convexity and strong smoothness: learning applications and matrix regularization // http://ttic.uchicago.edu/ shai/papers/KakadeShalevTewari09.pdf


*Поступила в редакцию 05.02.2019*

# Accelerated Nesterov Method for Decentralized Distributed Optimization on Time-Varying graphs

***Alexander Rogozin***[1]

aleksandr.rogozin@phystech.edu

[1]MIPT, Dolgoprudniy

The paper is focused on first-order methods in case when the aim function changes from one iteration to another. This problem is motivated by distributed optimization on networks which can periodically change because of technical malfunctions such as a loss of connection between two nodes. The main results of the paper include theoretical guarantees for linear convergence of distributed gradient descent and distributed Nesterov accelerated method on strongly convex smooth objective functions under the assumption that the network has a finite number of changes.

**DOI:** 10.21469/22233792.4.5.02

## References

[1] Nikaido Hukukane *Convex Structures and Economic Theory.* / Academic Press, 1968

[2] Danskin John *The Theory of Max-Min and its Applications to Weapons Allocation Problems.* / Springer-Verlag, 1967

[3] Demianov V., Malozemov V. *Introduction into Min-Max* / Nauka, 1972

[4] Bertsekas D. *Convex Optimization Theory* / Athena Scientific, 2009

[5] Bansal N., Anupam G. *Potential-Function Proofs for First-Order Methods // arXiv:1712.04581*, 2007

[6] Nedic A., Ozglar A. *Cooperative distribution multi-agent optimization. // Convex Optimization in Signal Processing and Communications*, 2009

[7] Bach F., Scaman K., Bubeck S., Lee Y.T. and Massoulié L *Optimal algorithms for smooth and strongly convex distributed optimization in networks // arXiv:1702.08704*, 2017

[8] Gasnikov A. *Universal gradient descent // arXiv:1711.00394*

[9] Kakade S.M., Shalev-Shwartz S., Tewari A. *On the duality of strong convexity and strong smoothness: learning applications and matrix regularization // http://ttic.uchicago.edu/ shai/papers/KakadeShalevTewari09.pdf*

*Received February 05, 2019*