# Применение алгоритма "Полоска" в задаче онлайнового машинного обучения

М. М. Липкович[1], Д. В. Миронов[2]

Рассмотрена задача онлайнового обучения моделей, где данные для обучения становятся доступными не сразу, а поступают в последовательном порядке. Для решения поставленной задачи используется алгоритм «Полоска», предложенный В.А. Якубовичем. Эффективность метода показана в его сравнении со стандартной моделью, обученной стохастическим градиентным спуском.

**Ключевые слова:** онлайновое машинное обучение, линейные модели, алгоритм "Полоска".

## 1. Введение

В классической постановке машинного обучения предполагается наличие набора данных, доступного к моменту обучения моделей. В работе рассматривается случай, когда такого предзаданного набора данных нет, и данные для обучения приходят последовательно. Такая постановка задачи называется онлайновым машинным обучением [1]. Потребность в моделях такого рода возникает, например, при работе с большими данными когда вычислительно неосуществимо совершить обучение на всем наборе, и в областях где данные меняются во времени, в следствие чего модели необходимо под них подстраиваться. Одним из наиболее популярных методов такого обучения является стохастический градиентный спуск, где в каждый момент времени модель обучается на пакете из одного семпла [2].

В данной работе предложено использовать алгоритм «Полоска» [3]. Этот алгоритм сводит задачу нахождения оптимальных параметров к решению системы бесконечных неравенств. В работе будет рассмотрен вариант построения бесконечной системы неравенств, которая хорошо ложится в парадигму онлайнового машинного обучения. Применимость

[1] *Липкович Михаил Маркович* — старший научный сотрудник ИПМаш РАН, доцент кафедры теоретической кибернетики мат.-мех. ф.-та СПбГУ, e-mail: lipkovich.mikhail@gmail.com.

Mikhail Lipkovich — senior researcher at IPME RAS, associate professor at Saint Petersburg State University, Department of Theoretical Cybernetics

[2] *Миронов Дмитрий Викторович* — студент магистр мат.-мех. ф.-та СПбГУ по программе «Прикладная математика и информатика», e-mail: dmitrvikizz@gmail.com.

Mironov Dmitrii — master's student at Saint Petersburg State University, "Applied mathematics and informatics" program

алгоритма будет показана в его сравнении с логистической регрессией, обученной стохастическим градиентным спуском.

## 2. Описание алгоритма «Полоска»

Рассмотрим модель, предложенную В. А. Якубовичем [3].

Пусть дан набор данных $\{x_i, y_i = s(x_i)\},\ i = 1, \ldots, m$, где $x_i \in \mathbb{X} \subset \mathbb{R}^n$ - векторы признаков, $s(x)$ - неизвестная функция из $\mathbb{X}$ в $\{1, -1\}$. Задана задача бинарной классификации, т.е. требуется восстановить функцию $s$ по имеющемуся набору данных.

Пусть $\{a_j(x)\},\ j = 1, \ldots, N, \ldots$ – полный в $L^2(\mathbb{X})$ набор ограниченных функций. В алгоритме «Полоска» строится классификатор вида

$$\sigma(x) = \operatorname{sign}\left(\sum_{j=1}^{N} \theta_j a_j(x)\right) = \operatorname{sign}(s_N(x, u)), \tag{1}$$

где $u$ – вектор обучаемых параметров с элементами $\theta_1, ..., \theta_N$.

Строя опознающую систему исходя из условий минимизации функции квадрата ошибки приходят к следующему неравенству:

$$\left|\sum_{k=1}^{N} \alpha_{hk}^{(m)} \theta_k - \alpha_h^{(m)}\right| < \varepsilon, \qquad h = 1, 2, ..., N \tag{2}$$

$$\alpha_h^{(m)} = \frac{1}{m} \sum_{j=1}^{m} s(x_j) a_h(x_j), \qquad \alpha_{hk}^{(m)} = \frac{1}{m} \sum_{j=1}^{m} a_k(x_j) a_h(x_j). \tag{3}$$

Для применения алгоритма «Полоска» необходимо циклически продолжить (2) до бесконечной системы неравенств. Один из способов этого достичь - рассмотреть бесконечную по $h$ систему, сделав все функции циклическими по $h$. В этом случае, весь датасет должен быть доступен сразу. Однако, нас интересует возможность применения данного алгоритма к онлайновому машинному обучению, поэтому вместо этого предлагается рассмотреть систему, бесконечную по $m$. Тогда на каждое добавление семпла данных будут приходиться $N$ неравенств типа (2).

Обозначим $\delta_h^{(m)} = \sum\limits_{k=1}^{N} \alpha_{hk}^{(m)} \theta_k^{(m)} - \alpha_h^{(m)}$, где верхний индекс $m$ теперь обозначает номер итерации и рассмотрим следующий алгоритм обновления весов:

$$\theta_k^{m+1} = \begin{cases} \theta_k^m, \text{ если } \left|\delta_h^{(m)}\right| < \varepsilon, \\ \theta_k^m - \delta_h^m \alpha_{hk}^{(m)} \left[\sum\limits_{k=1}^{N} \left(\alpha_{hk}^{(m)}\right)^2\right]^{-1}, \text{ если } \left|\delta_h^{(m)}\right| \geqslant \varepsilon. \end{cases} \tag{4}$$

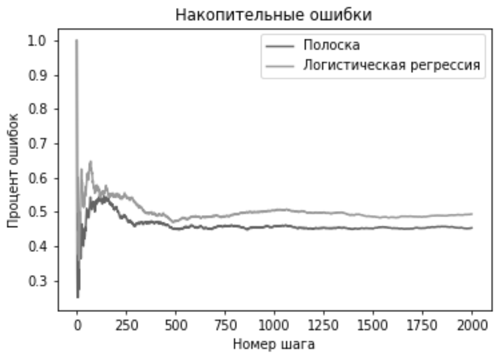
**Теорема.** Данный алгоритм является конечно-сходящимся

*Доказательство.* Основна на общем результате из [3].

## 3. Результаты моделирования

Для проверки применимости алгоритма «Полоска» к поставленной задаче сравним его с логистической регрессией, обученной методом стохастического градиентного спуска. Сравнения будет происходить на наборе данных "HIGGS Data Set"[1] [4], состоящем из 11000000 семплов по 28 признаков.

Обеим моделям будут последовательно предъявляться семплы из датасета. На каждом семпле будем выполнять прогноз, и запоминать общее число ошибок к данному шагу. Далее, этот семпл будет предъявляться для обучения. Такой подход сравнения онлайновых моделей был предложен в [5]. Графики процентов накопительных ошибок для первых 2000 семплов представлены ниже. Алгоритм «Полоска» с самого начала начинает совершать меньшее количество ошибок. Итоговый процент ошибок для Полоски» - 38%, а для регрессии - 44%. Таким образом, предложенный алгоритм обладает потенциалом, и стоит заняться его улучшением и распространением на другие задачи.




## 4. Благодарности

Исследование выполнено за счет гранта Российского научного фонда (проект №21-71-00144)

[1] https://archive.ics.uci.edu/ml/datasets/HIGGS

## Список литературы

[1] S. Hoi, Doyen Sahoo, Jing Lu, P. Zhao, “Online Learning: A Comprehensive Survey”, *Neurocomputing*, **459** (2021), 249–289.

[2] Guillaume Bouchard and Théo Trouillon and Julien Perez and Adrien Gaidon, “Online Learning to Sample”, *arXiv: Learning*, 2015.

[3] Якубович В. А., “Рекуррентные конечно-сходящиеся алгорифмы решения систем неравенств”, *Доклады Академии наук СССР*, **166**:6 (1966), 1308-1312.

[4] Baldi, P., P. Sadowski, and D. Whiteson, “Searching for exotic particles in high-energy physics with deep learning”, *Nature Communications*, **5** (2014).

[5] Ma, Justin and Saul, Lawrence K. and Savage, Stefan and Voelker, Geoffrey M., “Identifying Suspicious URLs: An Application of Large-Scale Online Learning”, *ICML ’09: Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, 681–688.


**Application of "Stripe" algorithm for online machine learning**


**Lipkovich M., Mironov D.**



The present paper considers online learning problem where training data becomes available in a sequential order. "Stripe" algorithm proposed by V.A. Yakubovich is used for online learning. Its applicability for online machine learning is confirmed by comparison with a traditional model trained using stochastic gradient descent.



*Keywords:* online machine learning, linear models, "Stripe" algorithm



## References

[1] S. Hoi, Doyen Sahoo, Jing Lu, P. Zhao, “Online Learning: A Comprehensive Survey”, *Neurocomputing*, **459** (2021), 249–289.

[2] Guillaume Bouchard and Théo Trouillon and Julien Perez and Adrien Gaidon, “Online Learning to Sample”, *arXiv: Learning*, 2015.

[3] Yakubovich V. A., “Recurrent finitely convergent algorithms for solving systems of inequalities (In Russian)”, *Dokl. Akad. Nauk SSSR*, **166**:6 (1966), 1308-1312.

[4] Baldi, P., P. Sadowski, and D. Whiteson, “Searching for exotic particles in high-energy physics with deep learning”, *Nature Communications*, **5** (2014).

[5] Ma, Justin and Saul, Lawrence K. and Savage, Stefan and Voelker, Geoffrey M., “Identifying Suspicious URLs: An Application of Large-Scale Online Learning”, *ICML ’09: Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, 681–688.