Санкт-Петербургский Государственный Университет
Математико-механический факультет

Программная инженерия

Турсунова Мунира Бахромовна

Реализация нового ускоренного алгоритма онлайн-обучения в библиотеке scikit-learn

Отчёт по производственной (преддипломной) практике

Научный руководитель:
проф. кафедры СП, д. ф.-м. н. О. Н. Граничин

Рецензент:
с.н.с. ИПМаш РАН, к. ф.-м. н. В. А. Ерофеева

Санкт-Петербург
2023

Оглавление

Введение

Машинное обучение получило широкий резонанс за последние пару десятков лет. Каждый год алгоритмы машинного обучения находят применение в новых областях: медицина, финансы, онлайн образование и многое другое. В последнее время начали набирать популярность новые методы машинного обучения, которые приходят на замену предыдущим. Одним из таких методав является онлайновое машинное обучение. В отличии от традиционнной модели, в онлайновом машинном обучении данные становятся доступны последовательно и используются для обновления предсказания на последующих шагах. Онлайновое обучение является общей техникой, используемой в областях машинного обучения, когда невозможна тренировка по всему набору данных. Например, при работе с большими объемами данных, не помещающихся в память, а также в ситуациях, когда алгоритму приходится динамически приспосабливаться к новой модели данных или когда сами данные образуются как функция от времени, например, при предсказании цен на фондовом рынке.

Одним их классических и наиболее популярных методов как пакетного, так и онлайноввого машинного обучения является метод стохастического градиентного спуска. Несмотря на то, что метод градиентого спуска существует в сообществе машинного обучения уже давно, совсем недавно он привлек значительное внимание в контексте крупномасштабного обучения. С момента появления стохастического градиентного спуска, начало разрабатываться множество новых методов численной оптимизации, которые в большинстве своём являются модификацией градиентного спуска [13, 10, 17, 5, 15, 8, 6, 3, 2, 12, 16, 7, 14]. Тем не менее, на сегодняшний день, среди методов пакетного машинного обучения, наибольшей популярностью пользуются ускоренные градиентные методы вследствии быстрой скорости сходимости. Одним из первых таких методов является метод тяжелого шарика, выдвинутый Б. Поляком в середине прошлого века [13], однако сейчас метод тяжелого шарика редко используется на практике, а вместо него используются такие методы, как быстрый градиентный метод Нестерова, являющийся модификацией метода тяжелого шарика [10, 11], а также метод Адама, также являющийся модификацией градиентного спуска [6]. Другой областью развития градиентного спуска является методы нулевого порядка, которые, в отличии от классичесских градиентных методов, не накладывают определенные ограничения на исследуемые функции, вследствии чего показывают лучшие результаты на зашумленных данных и не требуют дополнительные вычислительные затраты на определение градиента иследумой функции [9].

Поскольку методы онлайнового машинного обучения начали набирать интерес среди исследователей отностительно недавно, большинство существующих на сегодняшний день программных продуктов онлайнового машинного обучения довольно ограничены. Одним из наиболее популярных таких инструментов для является библиотека scikit-learn, в которой из алгоритмов онлайнового машинного обучени реализован только градиентный спуск и пассивный агрессивный регрессор для регрессионных задач.

Все вышеупомянутое указывает на актуальность разработки и реализации новых ускоренных алгоритмов для машинного обучения в реальном времени в библиотеке scikit-learn.

1. Постановка задачи

Целью магисстерской работы является разработка и реализация нового ускоренного алгоритма онлайн-обучения в библиотеке scikit-learn. Для достижения вышеупомянутой цели были выдвинуты следующие задачи:

- Исследовать существующие алгоритмы онлайнового машинного обучения.
- Разработать новый ускоренный SPSA алгоритм для задач трекинга.
- Исследовать существующие программные продукты онлайнового машинного обучения.
- Изучить реализацию библиотеки scikit-learn.
- Имплементироваввть новый алгоритм онлайн-обучения в библиотеке scikit-learn.
- Провести сравнитльное тестирование нового метода с уже реализованными в scikit-learn.

2. Обзор алгоритмов

2.1. Постановка задачи

В статическом варианте методов машинного обучения с учителем, главная задача заключаются в поиске такой функции $y = F(x)$, $F : X \to Y$, которая бы минимизировала функцию потерь $f(F(x), y)$, $V : Y \times Y \to \mathbb{R}$. В данном случае X рассматривается как пространство входных данных, а Y — пространство выходных данных.

В этом случае задача состоит в поиске набора параметров $\theta_0, ..., \theta_t$, которые минимизируют функцию потерь $Y(\theta)$:

$$Y(\theta) = \mathbb{E}f(y, \theta) = \int f(y, \theta) dP(y) \tag{1}$$

Потери $Y(\theta)$ не могут быть минимизированы напрямую, поскольку распределение $dP(y)$ неизвестно. Однако возможно вычислить приближение $Y(\theta)$ по конечному обучающему набору независимых наблюдений $y_1, ..., y_t$.

$$Y(\theta) \approx \hat{Y}_t(\theta) = \frac{1}{t} \sum_{i=1}^{t} f(y_i, \theta) \tag{2}$$

Поскольку практическая часть работы нацелена на решение линейных моделей в scikit-learn, для описания методов далее будет использоваться модели, в которых целевое значение ожидается как линейная комбинация характеристик:

$$F(x, \theta) = \theta_0 + \theta_1 x_1 + ... + \theta_d x_d. \tag{3}$$

2.2. Метод пакетного градиентного спуска

В классическом, пакетном подходе метода градиентного спуска, для вычисления оценки параметров θ_t на каждом шаге используется следующая формула:

$$\theta_{t+1} = \theta_t - \gamma_t \nabla_\theta \hat{Y}_t(\theta_t) = \theta_t - \gamma_t \frac{1}{t} \sum_{i=1}^{t} \nabla_\theta f(y_i, \theta_t) \tag{4}$$

,

где γ_t - положительное число.

В этом случае, на каждой итерации, для расчета оценки θ_t вычисляется среднее значения градиентов функции потерь $\nabla_\theta f(y_i, \theta)$ по всей обучающей выборке.

2.2.1. Инкрементальный метод стохастического градиентного спуска

В инкрементальном подходе градиентного спуска вместо усреднения градиента потерь по всей обучающей выборке на каждой итерации выбирается случайный пример y_t и параметр θ_t обновляется слудеющим образом:

$$\theta_{t+1} = \theta_t - \gamma_t \nabla_\theta f(y_t, \theta_t) \tag{5}$$

2.3. Ускоренные методы

Ускоренные градиентные методы позвляют значительно уменьшить количество шагов, за которое алгоритм начнет выдавать минимальные помехи за счет двигажения в том же направлении что и на предыдущей итерации. Другими словами, общую формулу итеративного поиска оценки точки минимума можно записать следующим образом:

$$\theta_{t+1} = \theta_t - \gamma_t \nabla_\theta f_t(\theta_t) + \alpha_t(\theta_t - \theta_{t-1})$$

Параметр α_t вычисляется итеративно.

2.3.1. Быстрый стохастический градиентный метод для задач трекинга

Алгоритм быстрого стохастического градиентного метода для задач трекинга (SFGT) основан на ускоренном методе Нестерова [10]. В алгоритме рассматриваютсяс измерения градиентов функций, искаженные аддитивным шумом $\xi_t \in R^q$:

$$Y_t(\theta) = \nabla f_t(\theta) + \xi_t$$

Общая структура алгоритма следующая:

1. Выбираются значения параметров $h > 0$, $\eta \in (0, \mu)$, $\alpha_x \in (0, 1)$ так, что значение α_t, удовлетворяющее неравенству (6), всегда может быть найдено.

2. Выбираются значения $\theta_0 \in \mathbb{R}^q$, $\gamma_0 > 0$.
   Фиксируется значение $v_0 = \theta_0$.
   Вычисляется значение переменной H_1: $H_1 = h - \frac{h^2 L}{2}$.

3. На t-ой итерации ($t \geq 0$):

   (a) Найти $\alpha_t \in [\alpha_x, 1)$ такое, что

   $$H_1 - \frac{\alpha_t^2}{2\gamma_{t+1}} > 0. \tag{6}$$

   (b) Вычислить $\gamma_{t+1} = (1 - \alpha_t)\gamma_t + \alpha_t(\mu - \eta)$.

   (c) Вычислить $x_t = \frac{1}{\gamma_t + \alpha_t(\mu - \eta)}\Big(\alpha_t \gamma_t v_t + \gamma_{t+1}\theta_t\Big)$

   (d) Рассчитать значение $Y_t(x_t)$.

   (e) Найти новую оценку θ_{t+1}: $\theta_t = x_t - hY_t(x_t)$.

   (f) Установить $v_{t+1} = \frac{1}{\gamma_{t+1}}\Big[(1 - \alpha_t)\gamma_t v_t + \alpha_t(\mu - \eta)x_t - \alpha_t Y_t(x_t)\Big]$.

3. Ускоренный SPSA алгоритм для задач трекинга

Ускоренный SPSA алгоритм для задач трекинга позволяет решить задачу поиска последовательности параметров $\theta_t \in \mathbb{R}^q$, которые минимизируют функцию потерь $Y(\theta)$, иными словами позволяет решить задачу отслеживания параметров (1). Алгоритм представляет собой модификацию ускоренного метода Нестерова для задачи отслеживания с применением алгоритма SPSA [4].

3.1. Постановка задачи

Пусть $(\Omega, \mathcal{F}, P)$ – вероятностное пространство, соответствующее выборочному пространству Ω, набору событий $\mathcal{F}$ и вероятностной мере P. $\mathbb{E}$ – математическое ожидание. Пусть $\mathcal{F}_{t-1}$ – σ-алгебра всех вероятностных событий, произошедших до момента времени $t = 1, 2, \ldots$, $\mathbb{E}_{\mathcal{F}_{t-1}}$ условное математическое ожидание относительно σ-алгебры $\mathcal{F}_{t-1}$.

Рассмотрим задачу оптимизации среднего риска $F(\theta)$:

$$\min_{\theta}\{F(\theta) = \mathbb{E}_{\xi_t} f(\theta, \xi_t)\}, \tag{7}$$

где $\theta \in \mathbb{R}^d$ – вектор оценок, ξ_t неопределенность, принадлежащая множеству Ξ. Задача (7) возникает во многих практических приложениях, а также в машинном обучении. Неопределенность представлена неконтролируемой детерминированной последовательностью (например, $\Xi = \mathbb{N}$ и $\xi_t = t$) или случайной последовательностью.В последнем случае предполагается, что распределение вероятности ξ_t существует и может быть известно или неизвестно.

В алгоритмме рассматривается оптимизация нулевого порядка, когда имеются только зашумленные измерения функции, которые нужно оптимизировать. В отличие от большинства существующих решений, алгоритм не делает никаких статистических предположений относительно помех. Также предполагается, что параметр θ не может быть измерен напрямую. Таким образом, вводится последовательность точек измерения $\mathbf{x}_1, \mathbf{x}_2, \ldots$, выбранных согласно плану наблюдения. ТЗначения $y_1, y_2, \ldots$ функций $f_{\xi_t}(\cdot)$ наблюдаемы в каждый момент времени $t = 1, 2, \ldots$ с аддитивными внешними неизвестными, но ограниченными (unknown-but-bounded) помехами v_t

$$y_t = f_{\xi_t}(\mathbf{x}_t) + v_t. \tag{8}$$

Здесь и далее мы заменяем обозначение $f(\theta, \xi_t)$ на $f_{\xi_t}(\cdot)$, подчеркивая, что ξ_t является неконтролируемой последовательностью. Также, минимизатор θ $F(\theta)$ может меняться со временем. Формально *нестационарная задача оптимизации среднего риска* выглядит следующим образом: оценить изменяющуюся во времени точку минимума θ_t функции

$$F_t(\theta) = \mathbb{E}_{\mathcal{F}_{t-1}} f_{\xi_t}(\theta) \to \min_{\theta_t}. \tag{9}$$

Предположение 1. Функция $F_t(\cdot)$ сильно выпукла:

$$\forall \mathbf{x} \in \mathbb{R}^d \;\; \langle \mathbb{E}_{\mathcal{F}_{t-1}} \nabla f_{\xi_t}(\mathbf{x}), \mathbf{x} - \theta_t \rangle \geq \mu \|\mathbf{x} - \theta_t\|^2,$$

где θ_t – точка минимума функции

Предположение 2. Для любого $\forall \xi \in \Xi$, градиент $\nabla f_\xi(\mathbf{x})$ удовлетворяет условию Липшица: $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$

$$\|\nabla f_\xi(\mathbf{x}_1) - \nabla f_\xi(\mathbf{x}_2)\| \leq L \|\mathbf{x}_1 - \mathbf{x}_2\|$$

с константой $L > 0$.

Предположение 3. Для всех $n \geq 0$ и $\forall \mathbf{x} \in \mathbb{R}^d$, $a, b > 0$, изменение точки минимума ограничено:

$$\text{a)}\;\; \mathbb{E}_{\mathcal{F}_{n-1}} |f_{\xi_n}(\mathbf{x}) - f_{\xi_{n+1}}(\mathbf{x})| \leq a \mathbb{E}_{\mathcal{F}_{n-1}} \|\nabla f_{\xi_n}(\mathbf{x})\| + b,$$
$$\text{b)}\;\; \mathbb{E}_{\mathcal{F}_{t-1}} \|\nabla f_{\xi_t}(\mathbf{x}) - \nabla f_{\xi_{t-1}}(\mathbf{x})\| \leq c.$$

Предположение 4. Для $n = 1, 2, \ldots$, последовательное разности помех $\tilde{v}_n = v_{2n} - v_{2n-1}$ ограничены: $|\tilde{v}_n| \leq c_v < \infty$, or $\mathbb{E}(\tilde{v}_n)^2 \leq c_v^2$, когда последовательность $\{\tilde{v}_n\}$ случайная.

Предположение 5: Для любого $n = 1, 2, \ldots$,

a) Δ_n и ξ_{2n-1}, ξ_{2n} (если они случайны) не зависят от σ-алгебры $\mathcal{F}_{2n-2}$.
b) Если $\xi_{2n-1}, \xi_{2n}, \tilde{v}_n$ случайны, тогда случайные векторы Δ_n и элементы $\xi_{2n-1}, \xi_{2n}, \tilde{v}_n^i$ независимы.
d) Для $n = 1, 2, \ldots,$ векторы $\mathbf{z} = \Delta_n$ наряду с одновременным возмущением симметричных функций распределения $\mathrm{P}_n(\cdot)$ удовлетворяют следующим условиям:

$$\int \mathbf{z}\mathrm{P}_n(d\mathbf{z}) = 0, \quad \int \|\mathbf{z}\|^2 \mathrm{P}_n(d\mathbf{z}) \leq c_\Delta^2.$$

Предположения 1 и 2 часто встречатся в задачах оптимизации. Предположение 3 используется в задачах с нестационарными данными.

Пусть $\Delta_n \in R^d$, $n = 1, 2, \ldots$ - независимая случайная велчина, т.е., *simultaneous test perturbation*, взятая из распределения Бернулли. Каждая компонента вектора независимо принимает значение $\pm\frac{1}{\sqrt{d}}$ с вероятностью $\frac{1}{2}$.

Замечание: итерационный процесс разбиается на блоки, а n показывает номер текущего блока. Например, если $n = 1$, алгоритм (**??**) обеспечивает расчеты для моментов времени $t = 2n - 1$ и $t = 2n$ тогда как переменные с нижним индексом n вычисляются только один раз и используются в оба момента времени.

3.2. Описание алгоритма

Рассмотрим алгоритм с двумя наблюдениями функций $f_{\xi_t}(\cdot)$ для построения последовательностей точек измерения $\{\mathbf{x}_t\}$ и оценок $\{\widehat{\theta}_t\}$:

1. Выбирется начальное состояние вектора оценок $\hat{\theta}_0 \in \mathbb{R}^d$, и параметров $\gamma_0 > 0$, $h > 0$, $\beta > 0$, $\eta \in (0, \mu)$, $\alpha_0 \in (0, 1)$. Также определяются переменные $z_0 = \hat{\theta}_0$ и $H = h - \frac{h^2 L}{2}$. На каждом шаге n, выбириается $\alpha_x \in (0, 1)$, такое что условие (10) выполнается, где $\alpha_n \in [\alpha_x, 1)$:
$$H - \frac{\alpha_n^2}{2\gamma_n} > 0, \tag{10}$$
и $\gamma_n = (1 - \alpha_{n-1})\gamma_{n-1} + \alpha_{n-1}(\mu - \eta - 1)$.

2. На шаге $n \geq 1$:

   (a) Вычислить

$$\widetilde{\mathbf{x}}_{2n-2} = \frac{1}{\gamma_{n-1} + \alpha_n(\mu - \eta - 1)} \Big(\alpha_n \gamma_{n-1} \mathbf{z}_{2n-2} + \gamma_n \widehat{\theta}_{2n-2} \Big),$$
$$\widetilde{\mathbf{x}}_{2n-1} = \widetilde{\mathbf{x}}_{2n-2}.$$

   (b) Вычислить

$$\mathbf{x}_{2n} = \widetilde{\mathbf{x}}_{2n-2} + \beta \Delta_n, \ \mathbf{x}_{2n-1} = \widetilde{\mathbf{x}}_{2n-2} - \beta \Delta_n.$$

   (c) Найти градиент:

$$\mathbf{g}_{2n} = \Delta_n \frac{y_{2n} - y_{2n-1}}{2\beta}.$$

   (d) Установить оценки:

$$\widehat{\theta}_{2n-1} = \widehat{\theta}_{2n-2}, \ \widehat{\theta}_{2n} = \widetilde{\mathbf{x}}_{2n-1} - h \mathbf{g}_{2n}.$$

   (e) Найти новое значение параметра:

$$\mathbf{z}_{2n} = \gamma_n^{-1} \Big[(1 - \alpha_n) \gamma_{n-1} \mathbf{z}_{2n-2} + \alpha_n (\mu - \eta - 1) \widetilde{\mathbf{x}}_{2n-1} - \alpha_n \mathbf{g}_{2n}) \Big].$$

Замечание: Если константы, фигурирующие в предположениях 1-3, неизвестны, мы можем установить для них значения из наихудшего сценария.

3.3. Доказательство

Следующая теорема показывает верхнюю границу ошибки оценки.

Теорема 1. Пусть $\{A_n\}$ и $\{Z_n\}$ последовательности из $\mathbb{R}$ такие, что:

$$A_0 = 0, \quad A_{n+1} = (1 - \alpha_n)[(1 - \lambda_n) a + A_n],$$
$$Z_0 = 0, \quad Z_{n+1} = (1 - \lambda_{n+1})(b + ac) + A_{n+1} c.$$

. Если Предположения 1–5 выполняются, вышеописанный алгоритм генерирует последовательность оценок $\{\widehat{\theta}_n\}_{n=0}^{\infty}$ такую, что:

$$\mathbb{E}_{\mathcal{F}_{n-1}} f_{\xi_n}(\widehat{\theta}_n) - f_{\xi_n}(\theta_n) \leq$$
$$\lambda_n(\phi_0(\theta_0) - f_{\xi_n}(\widehat{\theta}_n) + \Phi) + D_n,$$

где

$$D_0 = 0, \quad D_{n+1} = (1-\alpha_n)D_n + (1-\alpha_n)Z_n + \frac{\alpha_n c^2}{2\eta} +$$
$$H_2 + b(1+\alpha_n) - Hc^2 + \frac{a + \alpha_n \bar{a} - 2Hc}{2\epsilon^2}.$$

и $\phi_0(\mathbf{x}) = f_0(\widehat{\theta}_0) + \frac{\gamma_0}{2}\|\mathbf{x} - \widehat{\theta}_0\|^2$, $\Phi = \frac{\gamma_0 c^2}{2\mu^2}$, $\lambda_0 = 1$, $\bar{a} = (\frac{\alpha_n}{2\gamma_{n+1}}(2L\beta + c) + a)$, $H_2 = \frac{\alpha_n(2L\beta+c)^2(\alpha_n+\gamma_{n+1})}{8\gamma_{n+1}} + \frac{h^2 L(2L\beta+c)^2}{8}$, $\lambda_t \to 0$, $\gamma_0 > 0$.

Доказательство: Доказательство теоремы представлено в материалах конференции «The 2022 American Control Conference» [1].

4. Обзор программных продуктов

4.1. Программные продукты для онлайнового обучения

Несмотря на то, что методы онлайнового машинного обучения начали набирать популярность относительно недавно, на сегодняшний день существует немалое количество инструментов подходящих для решения задач онлайнового обучения: Vowpal Wabbit, PyTorch, TensorFlow, Scikit-Multiflow, River, DASK, Jubatus, LIBFFM, LIBLINEAR, LIBOL, MOA, scikit-learn, Spark Streaming, SofiaML, StreamDM, Tornado, VFML.

Хотя вышеупомянутый список доволньно объемный, во многих из перечисленных инструментов не реализованы алгоритмы онлайн обучения.

PyTorch К примеру, такие широкоиспользуемые библиотеки как PyTorch и TensorFlow еще не реализовали классические алгоритмы онлайнового обучения.

TensorFlow

River

Scikit-multi-flow

Jubatus

Creme Framework

Spark Streaming и StreamDM Spark Streaming и StreamDM не занимаются онлайн-обучением, а вместо этого мини-пакетируют данные в фиксированные интервалы времени.

Scikit-Multiflow Инкрементальное обучение в Scikit-Multiflow и dask реализовано на базе Scikit-learn, к тому же данные инструменты нацелены скорее на оптимизацию методов из Scikit-learn путем их распараллеливания.

Matlab Statistics and Machine Learning Toolbox™ позволяют вам реализовать добавочное обучение для классификации или регрессии. Как и другие функции машинного обучения Statistics and Machine Learning Toolbox, точка входа в добавочное обучение — это добавочный объект обучения, который вы передаете функциям с данными для реализации добавочного обучения. В отличие от других функций машинного обучения, данные не требуются для создания добавочного объекта обучения. Однако объект добавочного обучения указывает, как обрабатывать поступающие данные, например, когда подгонять модель, измерять показатели производительности или выполнять оба действия, в дополнение к параметрической форме модели и параметрам, характерным для конкретной задачи.

Vowpal Wabbit Vowpal Wabbit – быстрая система онлайнового обучения с внешней памятью и открытым кодом с некоторым набором поддерживаемых техник обучения машин, со взвешиванием важности и выбором различных функций потерь и алгоритмов оптимизации.

LIBOL Библиотека LIBOL представляет коллекцию линейных онлайн-моделей, обученных методами градиентного спуска первого и второго порядка, в данный момент не поддерживается.

Scikit-learn Онлайн-обучение в Scikit-learn реализовано в виде перцептрона, классификатора методом статистического градиентного спуска, наивного байесовского классификатора для классификационных задач и регрессией стохастическим градиентным спуском, пассивно агрессивного регрессора для регрессионных задач.

Для реализации была выбрана библиотека Scikit-learn ввиду следующих причин: удобного API для разработки новых методов, наличия реализации классических алгоритмов онлайнового обучения, популярности самой библиотеки, а также во время исследования предметной области было выявлено, что данная библиотека является одной из наиболее широкоиспользуемых в области онлайнового машинного обучения.

4.2. Библиотека Scikit-learn

Scikit-learn – это один из наиболее широко используемых пакетов Python для Data Science и Machine Learning. Библиотека Scikit-learn — один из самых распространенных выборлв для решения задач классического машинного обучения. Архитектура библиотеки Scikit-learn сосотит из 4 основных компонентов (Рис.1):

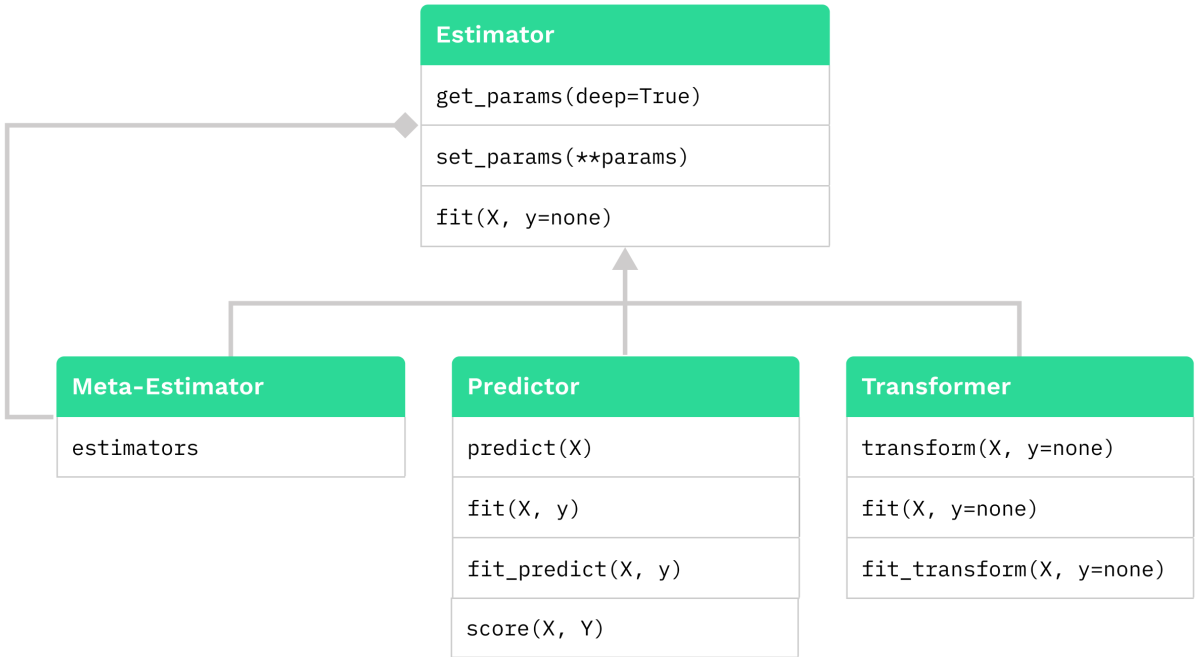

Рис. 1: UML диаграмма компонентов библиотеки Scikit-learn

- Estimators – каждый алгоритм, как для обучения так и для обработки данных, в scikit-learn реализован с помощью объекта Estimator. Это позволяет создать универсальный способ инициализации алгоритмов и получения гипер-параметров и обученных параметров.

- Predictor – это Estimators, реализующие методы прогнозирования и оценки. Например, алгоритмы, для классификации, регрессии, кластеризации и т.д.

- Transformeer – это Estimators, которые реализуют методы, связанные с преобразованием данных. В основном они используются для изменения или фильтрации данных на этапе препроцессинга перед передачей данных в предсказатели.

- Meta-estimators – это Estimators, который позволяет объединять один или несколько Estimators в единый объект.

Одним из наиболее широкоиспользуемых алгоримтмов онлайн-обучения в scikit-learn является инкрементальный стохастический градентный спуск, который реализован в библиотеке с помощью классов SGDClassifier и SGDRegressor. Вышеупомянутые классы были использованы для сравнительного тестирования нового алгоритмма.

5. Реализация

5.1. Архитектура

В классическом пакетном варианте обучения с учителем, данные делятся на тренировочные и тестовые. Первые доступны модели сразу во всем объеме, вследствии чего она обучается на всей выборке и вычисляется параметры модели Θ (пример линейной модели показан в (3)). Далее, имея вычесленный параметр Θ, делаюттся предсказания на тестовом наборе и вычисляется погрешность. Схема класссического пакетного подхода изображена на Рис. 2.

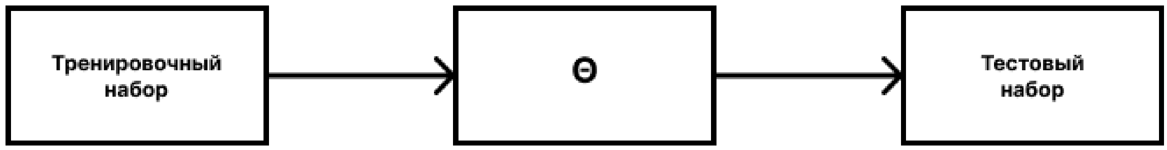

Рис. 2: Классическая схема пакетного обучения

В инкерементаальном подходе обучения с учителем данные становятся доступны постепенно, вследствии чего параметр модели Θ вычисляется инкрементально, так, что на каждой итерации пользователю стаановится доступны лишь часть новых измерений из тренировочного набора. Схема класссического пакетного подхода изображена на Рис. 3.

Онлайновое-обучение в scikit-learn реализовано неявно как mini-batch (ммини-пакетное) обучение с помощью метода partial_fit. Обучающий набор делится на пакеты (mini-batch) и подается в Estimators.

На Рис.4 представлена диаграмма классов UML для реализованных алгоритмов. Новый алгоритм Accelerated Simultaneous Perturbation Stochastic Approximation Algorithm for Tracking (ASPSAT) реализован с помощью классов ASPSATEstimator, ASPSATRegressor, ASPSATClassiffier. В классе ASPSATEstimator вычисляются параметры для алгоритма,

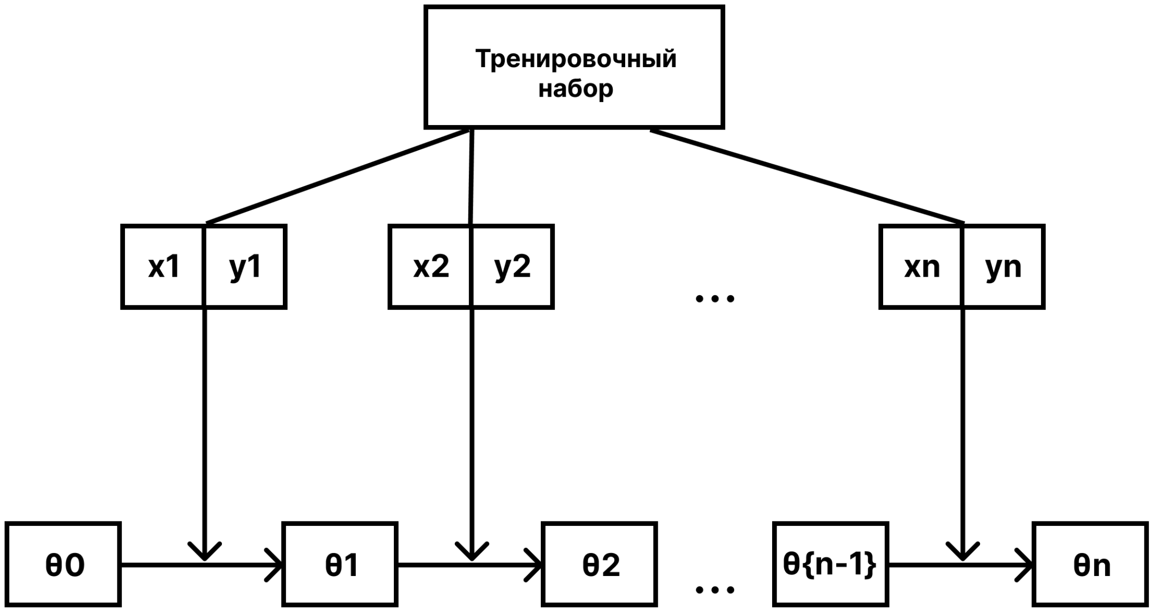

Рис. 3: Схема инкрементального обучения

описсанном в секции 3.2. Классы ASPSATRegressor и ASPSATClassiffier реализуют задачи регрессии и задачи классификации соответственно.

Поскольку сам алгоритм, описанный в секции 3.2, является инкрементальным, решающим задачу трекинга, инкрементальности в реализации классов можно добавиться используя ASPSATRegressor и ASPSATClassi с параметром max_iter (отвечает за число прогонов модели), равным 1. Однако для реализации инкрементального подхода в SGDRegressor и SGDClassifier был использован метод partial_fit, позволяющий подавать модели тренировочные данные по одной паре за раз.

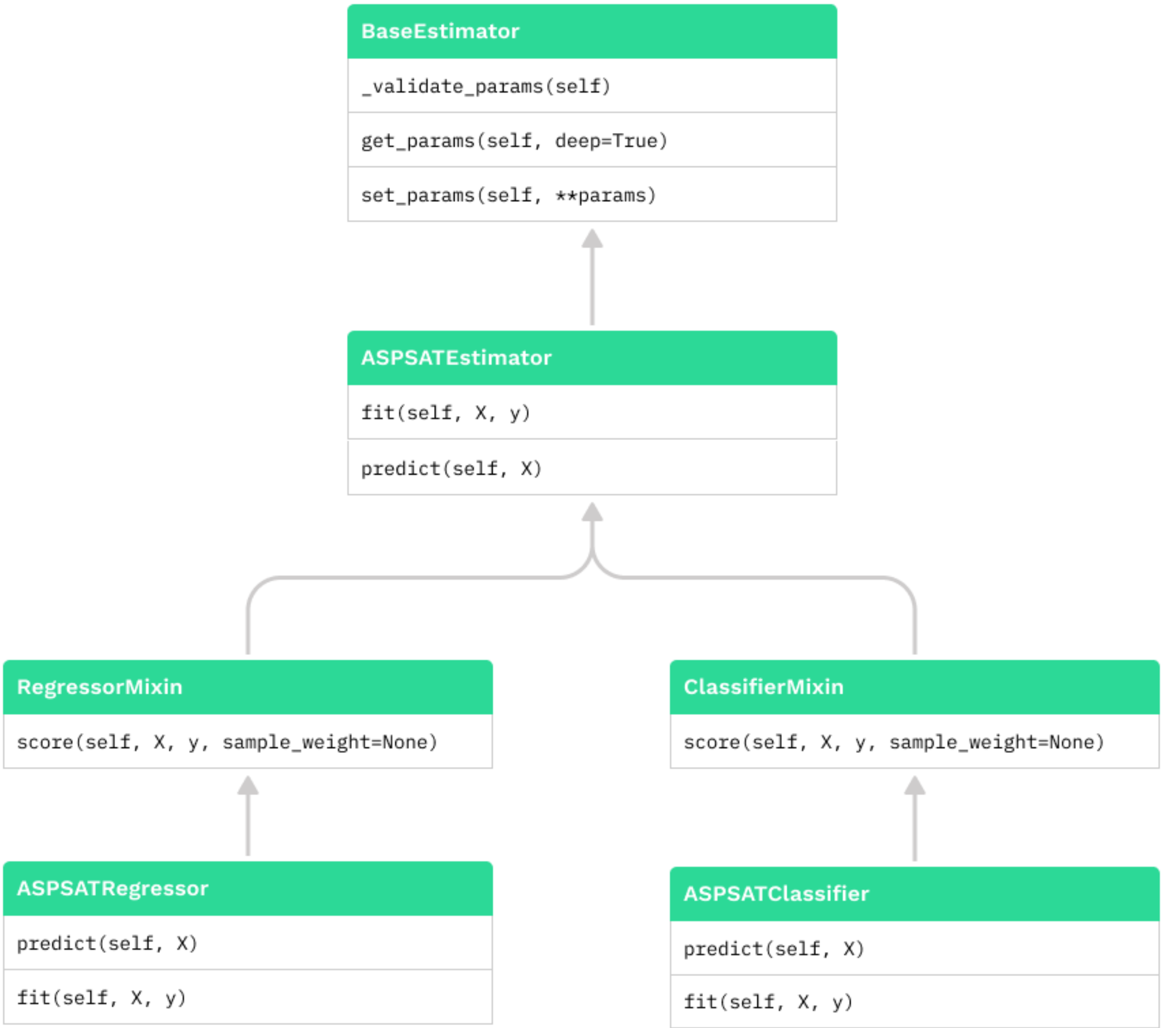

Рис. 4: UML диграмма классов

5.2. Апробация

Тестирование нового алгоритма проводилось на наборе данных о ценах на жилье в Бостоне из библиотеки scikit-learn. Для сравнения использовался метод SGDRegressor и метод partial_fit. Для того, чтобы данные поступали последовательно, по одному измерению за раз, размер пакета (batch size) был равен 1.

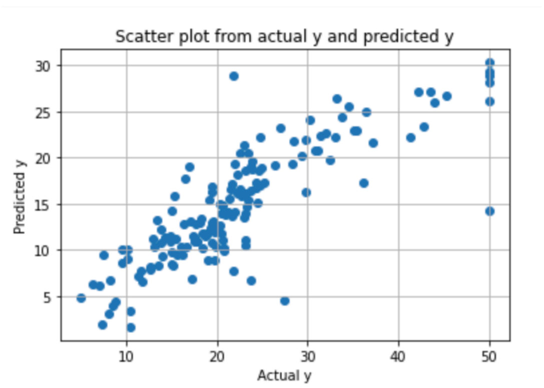

Рис. 5: SGDRegressor

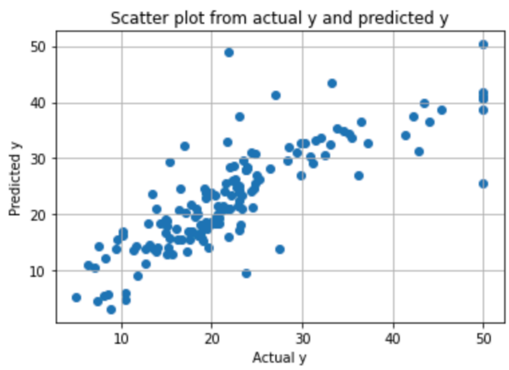

Рис. 6: Accelerated SPSA for tracking

На Рис.5 и Рис.6 представлены диаграммы рассеивания для действительного и предсказанного значения SGDRegressor из scikit-learn и нового реализованного метода.

Номер эксперимента	SGDRegressor	Accelerated SPSA for tracking
1	88.4670	33.5007
2	75.7340	21.1006
3	68.0885	23.2673
4	78.4397	24.0229
5	77.3681	22.2961
6	92.7304	35.5597

Таблица 1: RMSE

В таблице 1 показаны значения среднеквадратичной ошибки RMSE для вышеописанных алгоритмов на различных запусках.

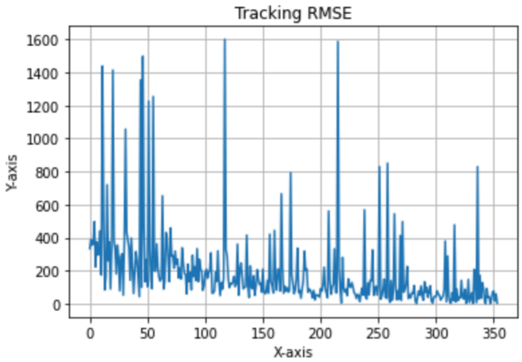

Рис. 7: SGDRegressor

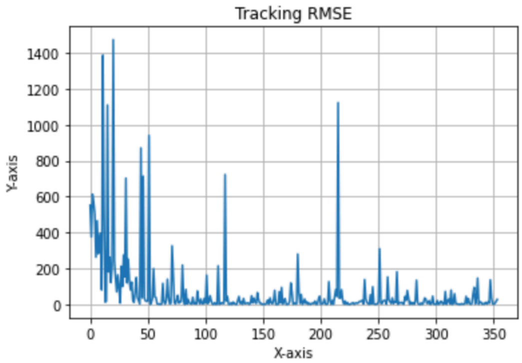

Рис. 8: Accelerated SPSA for tracking

На Рис.7 и Рис.8 представлены графики изменения среднеквадратичной ошибки RMSE на 350 итерациях для действительного и предсказанного значения SGDRegressor из scikit-learn и реализованного нового SPSA алгоритма для задач трекинга.

Заключение

На текущий момент сделано следующее:

- Исследованы существующие алгоритмы онлайнового машинного обучения.
- Разработан новый ускоренный SPSA алгоритм для задач трекинга.
- Исследованы существующие программные продукты онлайнового машинного обучения.
- Изучена реализация библиотеки scikit-learn.
- Имплементирован новый алгоритм в библиотеке scikit-learn.
- Получены результаты сравнительного тестирования для нового алгоритма.

Дополнительно:

- Теоретическая часть работы была представлена на конференциях «The 2022 American Control Conference» [1] и на XXIV конференции молодых ученых «Навигация и управление движением» [18].
- Работа была выполнена частично в рамках гранта «Мультиагентное адаптивное управление в сетевых динамических системах с применением к группам робототехнических устройств в условиях неопределенностей» от РНФ, 2021-2023.

Список литературы

[1] Erofeeva Victoria, Granichin Oleg, Tursunova Munira, Sergeenko Anna, and Jiang Yuming. Accelerated Simultaneous Perturbation Stochastic Approximation for Tracking Under Unknown-but-Bounded Disturbances // IEEE. — 2022. — Access mode: https://www.researchgate.net/publication/363301306_Accelerated_Simultaneous_Perturbation_Stochastic_Approximation_for_Tracking_Under_Unknown-but-Bounded_Disturbances.

[2] Blum R. G. Multidimensional stochastic approximation methods. — The Annals of Mathematical Statistics, 1954. — Access mode: https://www.researchgate.net/publication/38367267_Multidimensional_Stochastic_Approximation_Methods.

[3] Eweda E. and Macchi O. Tracking error bounds of adaptive nonsta- tionary filtering. — Automatica, vol. 21, no. 3, 1985. — Access mode: https://www.semanticscholar.org/paper/Tracking-error-bounds-of-adaptive-nonstationary-Eweda-Macchi/009d3d405ea305831b1397dbaf47047406891a61.

[4] Granichin O. and Amelina N. Simultaneous perturbation stochastic approximation for tracking under unknown but bounded disturbances // IEEE Transactions on Automatic Control. — 2015. — Access mode: https://ieeexplore.ieee.org/document/6908991.

[5] J. Kiefer J. Wolfowitz. Stochastic estimation of the maximum of a regression function. — The Annals of Mathematical Statistics, 1952. — Access mode: https://projecteuclid.org/euclid.aoms/1177729392.

[6] Kingma D. and Ba J. Adam: A method for stochastic optimization. — International Conference on Learning Representations (ICLR), 2014. — Access mode: https://www.researchgate.net/publication/269935079_Adam_A_Method_for_Stochastic_Optimization.

[7] Kushner H. J. and Huang H. Asymptotic properties of stochastic approximations with constant coefficients. — SIAM Journal on Control and Optimization, vol. 19, no. 1, 1981. — Access mode: https://www.researchgate.net/publication/243092849_Asymptotic_Properties_of_Stochastic_Approximations_with_Constant_Coefficients.

[8] Kushner H. J. and Yin G. G. Stochastic Approximation and Recursive Algorithms and Applications. — Springer Science and Business Media, 2003. — Access mode: https://link.springer.com/book/10.1007/b97441.

[9] Nelder J. A. and Mead R. A Simplex Method for Function Minimization // The Computer Journal. — 1965. — Access mode: https://academic.oup.com/comjnl/article-abstract/7/4/308/354237?redirectedFrom=fulltext.

[10] Nesterov Y.E. A method of solving a convex programming problem with convergence rate o(1/k2). — Soviet Mathematics Doklady, 1983. — Access mode: https://ci.nii.ac.jp/naid/10029946121/.

[11] Nesterov Y.E. Introductory Lectures on Convex Optimization: A Basic Course. — Springer Science and Business Media, 2013. — Access mode: https://www.springer.com/gp/book/9781402075537.

[12] Polyak B. T. Some methods of speeding up the convergence of iteration methods. — USSR Computational Mathematics and Mathematical Physics, 1964. — Access mode: https://www.sciencedirect.com/science/article/abs/pii/0041555364901375?via.

[13] Polyak B. T. Introduction to Optimization. — Optimization Software, 1987. — Access mode: https://www.researchgate.net/publication/268248877_Introduction_to_optimization_Vvedenie_v_optimizatsiyu.

[14] Popkov A. Y. Gradient methods for nonstationary unconstrained optimization problems. — Automation and Remote Control, 2005. —

Access mode: https://link.springer.com/article/10.1007/s10513-005-0132-z.

[15] Robbins H. and Monro S. A stochastic approximation method. — The annals of mathematical statistics, 1951. — Access mode: https://projecteuclid.org/euclid.aoms/1177729586.

[16] Spall J. C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. — IEEE Transactions on Automatic Control, 1992. — Access mode: https://www.researchgate.net/publication/3021008_Multivariate_stochastic_approximation_using_a_simultaneous_perturbation_gradient_approximation.

[17] Granichin O, Vakhitov A., Kosaty D., and Yuchi M. Stochastic fast gradient for tracking // American Control Conference (ACC). — 2019. — Access mode: https://ieeexplore.ieee.org/document/8815070.

[18] Ерофеева В.А., Сергеенко А.Н., and Турсунова М.Б. Ускоренный рандомизированный алгоритм стохастической аппроксимации для задачи трекинга // IEEE. — 2022. — Access mode: http://www.elektropribor.spb.ru/nauchnaya-deyatelnost/konferentsii/1611/.