Стохастический метод Ньютона с различным семплингом

Денис Швейкин **Рустем Исламов**

Аннотация

Задача минимизации среднего от большого числа гладких сильно выпуклых функций встречается в машинном обучении повсеместно. Стохастические методы первого порядка, такие как стохастический градиентный спуск (SGD), для этой задачи хорошо изучены. В свою очередь, методы второго порядка, такие как метод Ньютона имеют определенные преимущества, поскольку могут адаптироваться к кривизне задачи. Также они известны своей быстрой сходимостью. Однако стохастические варианты методов типа Ньютон изучены не так хорошо, как методы типа SGD, и имеют ограничения на размеры батчей. Ранее был предложен метод, который не требует таких ограничений. Наша работа исследует этот метод с различными стратегиями семплинга, которые ведут практическим улучшениям.

Ключевые слова Стохастический метод Ньютона, стратегии семплинга

1 Вступление

Задача состоит в минимизации функции эмпирического риска, который имеет форму конечной суммы [10]:

$$\min_{x\in\mathbb{R}^d}\left[f(x)\overset{def}{=}\frac{1}{n}\sum_{i=1}^{n}f_i(x)\right], \tag{1}$$

где каждая f_i предполагается имеющей липшицев гессиан.

Как правило, n - общее количество функций - в современных задачах является очень большим числом. Поэтому в виду вычислительное сложности оценки градиента всех f_i на каждом шаге используется стохастический подход. Стохастический градиентный спуск (SGD) [16] вычисляет градиенты некоторых

случайно выбранных f_i, что ведет к более вычислительно дешевым итерациям, по сравнению с традиционным градиентным спуском (GD). Метод SGD и его вариации довольно хорошо изучены. Можно отметить следующие плюсы данного метода. Во-первых, теория методов типа SGD не использует ограничений на размер батчей. Поэтому такие алгоритмы могут приниматься даже с малыми размерами батчей. Известно, что простой SGD дает сходимость только к некоторой окрестности оптимума [14, 4]. Тем не менее, есть техники (также известные как редукция дисперсии), разрешающие эту проблему. Эти техники [17, 7, 2, 6] модифицируют шаг SGD, что позволяет избавиться от вышеупомянутого эффекта, не меняя вычислительной сложности итерации. Однако главный недостаток всех градиентных методов состоит в том, что вычислительная сложность зависит от кривизны задачи, которая определяется как отношение константы Липшица и параметра сильной выпуклости, и называется числом обусловленности.

Здесь в дело вступают методы второго порядка типа Ньютон [13, 8, 3]. Учитывая производные второго порядка, становится возможно адаптировать шаг алгоритма под кривизну задачи [13]. К сожалению, намного меньше работы было проведено в направлении стохастических методов типа Ньютон. Многие методы [9, 1, 22, 19, 18, 21]. нуждаются в использовании батчей больших размеров. В частности, зачастую необходимый размер батча обратно-квадратично пропорционален желаемой точности. Это значит, что нужно считать гессианы от большого количества f_i, которое может иногда даже превышать n. [10] предлагает простой стохастический метод Ньютона, который может работать с батчами любого размера. Алгоритм достигает локальной линейной сходимости, а в некоторых случаях - сверхлинейной.

На практике для алгоритмов типа SGD используются различные стратегии семплинга, улучшающие их качество. Одним из наиболее известных механизмов семплинга является так называемый Importance Sampling (ИС) [5, 20, 11]. Идея состоит в том, чтобы вычислять градиенты от тех функций, которые вносят наибольший вкладв задачу. Другой метод рандомизированной оптимизации - Random Reshuffling, который делает градиентные шаги, проходясь подряд по случайно перемешанным данным [12]. [15] изучает множество других механизмов семплинга. Мы анализируем эти стратегии, но применительно к Алгоритму 1 из [10], чтобы получить теоретический и практические улучшения. Мы исследуем различные стратегии семплинга, подкрепляя их строго поставленными экспериментами.

2 Постановка задачи

Мы рассматриваем классическую задачу минимизации функции эмпирического риска (ERM), которая типично возникает во многих задачах машинного обучения в контексте обучения модели. Целевая функция f (1) является средним от большого количества функций f_i, где каждая f_i представляет собой функцию потерь на i-том тренировочном объекте. Например, эта нотация может быть применена к линейной регрессии. В данном случае задача состоит в нахождении оптимального вектора параметров x, который минимизирует среднеквадратическую ошибку (MSE) на обучающей выборке.

2.1 Предположения

Мы делаем стандартные предположения на функции f_i; те же, что были приняты в [10].

Предположение 1 (Сильная выпуклость)**.** *Дифференцируемая функция* $\phi: \mathbb{R}^d \to \mathbb{R}$ *называется* μ*-сильно выпуклой, где* $\mu > 0$*, если* $\forall\ x, y \in \mathbb{R}^d$

$$\phi(x) \geqslant \phi(y) + \langle \nabla\phi, x - y\rangle + \frac{\mu}{2}\|x - y\|^2, \tag{2}$$

где норма $\|\cdot\|$ - евклидова. Для дважды дифференцируемых функций данное предположение равносильно тому, что все собственные значения гессиана $\geqslant \mu$.

Предположение 2 (Липшицевы гессианы)**.** *Функция* $\phi : \mathbb{R}^d \to R$ *обладает* H*-липшицевым гессианом, если* $\forall\ x, y \in \mathbb{R}^d$

$$\|\nabla^2\phi(x) - \nabla^2\phi(y)\| \leqslant H\|x - y\| \tag{3}$$

2.2 Семплинг

Определение 1 (Семплинг)**.** Случайное множественнозначное отображение $\hat{S}: [n] \to 2^{[n]}$ называется семплингом.

Это значит, что каждое $S_k \subseteq [n]$ является реализацией $\hat{S}$. В таком лсучае мы можем назвать любое вероятностное распределение на $2^{[n]}$ стратегией семплинга.

2.3 Алгоритм

Мы применяем различные стратегии семплинга поверх Алгоритма 1 из [10]:

Algorithm 1 Стохастический метод Ньютона (SN)

Инициализация: Выбрать начальные приближения $w_1^0, w_2^0, ...w_n^0 \in \mathbb{R}^d$

for $k = 0, 1, 2, ...$ **do**

$$x^{k+1} = \left[\frac{1}{n}\sum_{i=1}^{n}\nabla^2 f_i(w_i^k)\right]^{-1}\left[\frac{1}{n}\sum_{i=1}^{n}\nabla^2 f_i(w_i^k)w_i^k - \nabla f_i(w_i^k)\right]$$

Выбрать подмножество $S^k \subseteq \{1, 2, ..., n\}$ одной из стратегий семплинга

$$w_i^{k+1} = \begin{cases} x^{k+1} & i \in S^k \\ w_i^k & i \notin S^k \end{cases}$$

end for

2.4 Цель проекта

Мы сравниваем различные стратегии семплинга для Алгоритма 1, доказывая гарантии сходимости и показывая практические улучшения по сравнению с базовым подходом.

3 Теория

3.1 Доказательство Алгоритма

Сходимость Алгоритма 1 была подробно описана и доказана в [10]. Для начала, обратимся к утверждениям трех лемм из данной статьи.

Лемма 1. *Пусть f_i - μ-сильна выпукла и имеет H-липшицев гессиан для всех $i = 1, ..., n$. Рассмотрим следующую функцию Ляпунова*

$$\mathcal{W}^k \overset{def}{=} \frac{1}{n}\sum_{i=1}^{n}\|w_i^k - x^*\|^2, \tag{4}$$

где x^* - точка оптимума для f (1). Тогда итерации Алгоритма 1 удовлетворяют

$$\|x^{k+1} - x^*\| \leqslant \frac{H}{2\mu}\mathcal{W}^k. \tag{5}$$

Лемма 2. *Предположим, что каждая f_i - μ-сильно выпукла и имеет H-липшицев гессиан. Если $\|w_i^0 - x^*\| \leqslant \frac{\mu}{H}$ для всех $i = 1, ..., n$, тогда для всех k*

$$\mathcal{W}^k \leqslant \frac{\mu^2}{H^2}. \tag{6}$$

Последняя лемма - именно та, которая опирается на природу стратегии семплинга. Авторы Алгоритма 1 используют так называемый τ-nice семплинг,

который выбирает подмножество $S^k \subseteq \{1, 2, ..., n\}$ в точности размера τ равновероятно. В таком случае можно определить математическое ожидание $\mathbb{E}_k[\mathcal{W}^{k+1}] \stackrel{\text{def}}{=} \mathbb{E}[\mathcal{W}^{k+1}|x^k, w_1^k, ..., w_n^k]$.

Лемма 3. *Итерации Алгоритма 1 с τ-nice семплингом удовлетворяют соотношению*

$$\mathbb{E}_k[\mathcal{W}^{k+1}] = \frac{\tau}{n}\|x^{k+1} - x^*\|^2 + \left(1 - \frac{\tau}{n}\right)\mathcal{W}^k \tag{7}$$

Далее следует теорема, дающая оценку сходимости Алгоритма 1 с τ-nice семплингом.

Теорема 1. *Предположим, что каждая f_i - μ-сильно выпукла и имеет H-липшицев гессиан. Тогда для итераций Алгоритма 1 выполнено рекуррентное соотношение*

$$\mathbb{E}_k[\mathcal{W}^{k+1}] \leqslant \left(1 - \frac{\tau}{n} + \frac{\tau}{n}\left(\frac{H}{2\mu}\right)^2 \mathcal{W}^k\right)\mathcal{W}^k. \tag{8}$$

Кроме того, если $\forall i \ \|w_i^0 - x^*\| \leqslant \frac{\mu}{H}$, то

$$\mathbb{E}_k[\mathcal{W}^{k+1}] \leqslant \left(1 - \frac{3\tau}{4n}\right)\mathcal{W}^k. \tag{9}$$

3.2 Стратегии семплинга

3.2.1 Вдвойне однородные сратегии

Для начала рассмотрим стратегии семплинга, представленные в [15] в отношении покоординатных методов. Эти стратегии являются *вдвойне однородными.*

Определение 2 (Вдвойне однородный (Doubly Uniform, DU) семплинг)**.** Это такие стратегии, которые возвращают все множества одинаковой мощности с одинаковой вероятностью. То есть если $|S_1| = |S_2|$, то $\mathbb{P}(S_1) = \mathbb{P}(S_2)$.

Название обусловлено тем, что данное свойство сильнее, чем *однородность*, которая означает, что все $i \in [n]$ имеют одинаковую вероятность быть включенными в выбранное множество $\hat{S}$. Это можно увидеть, если обозначить вероятность $q_j = \mathbb{P}(|\hat{S}| = j)$ для всех $j = 0, 1, ...n$. Тогда для всех $S \subseteq [n]$ имеем $\mathbb{P}(S) = q_j / \binom{n}{j}$. Отсюда

$$p_i = \sum_{j=1}^{n} \sum_{|S|=j, i \in S} \mathbb{P}(S) = \sum_{j=1}^{n} \sum_{|S|=j, i \in S} \frac{q_j}{\binom{n}{j}} = \sum_{j=1}^{n} \frac{\binom{n-1}{j-1} q_j}{\binom{n}{j}} = \frac{1}{n}\sum_{j=1}^{n} j q_j = \frac{\mathbb{E}[|\hat{S}|]}{n} \tag{10}$$

Ясно, что вдвойне однородная стратегия семплинга задается вектором q распределения вероятностей на всевозможных размерах $j = 0, 1, ..., n$ множества $\hat{S}$. Представляют интерес несколько DU-стратегий.

1. τ-**nice семплинг.** Зафиксируем $1 \leqslant \tau \leqslant n$. Семплинг называется τ-nice, если он является DU с $q_\tau = 1$. Именно эта стратегия используется авторами Алгоритма 1. Эту стратегию можно интерпретировать в терминах параллельных вычислений. Так, пусть есть τ доступных процессоров. В момент семплинга мы выбираем батч размера τ и назначаем каждый объект $i \in \hat{S}$ одному процессору. Таким образом, каждый процессор обновит значение целевой функции, градиенты и гессианы в соответствии с новым w_i^k.
2. τ-**независимый семплинг.** Зафиксируем $1 \leqslant \tau \leqslant n$. τ-независимым семплингом называется DU-семплинг с
$$q_k = \begin{cases} \binom{n}{k} c_k, & k = 1, 2, ..., \tau \\ 0, & k = \tau + 1, ..., n \end{cases} \tag{11}$$
где $c_1 = (1/n)^\tau$ и $c_k = (k/n)^\tau - \sum_{i=1}^{k-1} \binom{k}{i} c_i$ for $k \geqslant 2$.

   Интерпретация состоит в том, что каждый из τ процессоров выбирает один из объектов обучающей выборки $i \in [n]$. Если несколько процессоров выбрали один и тот же объект, то только один из них получает доступ к нему. Эта стратегия легко реализуема в терминах параллельных вычислений. Кроме того, если батчи невелики, мы имеем $\tau \ll n$, и тогда τ-независимый семплинг хорошо приближает τ-nice.
3. (τ, p_b)-**биномиальный семплинг.** Фиксируем $1 \leqslant \tau \leqslant n$ и $0 \leqslant p_b \leqslant 1$. Стратегия является DU-семплингом с
$$q_k = \binom{\tau}{k} p_b^k (1 - p_b)^k, \quad k \leqslant \tau \tag{12}$$
   Это модель ситуации, когда каждый из τ процессоров доступен в момент семплинга с вероятностью p_b, отсюда q_k - вероятность того, что k процессоров оказались доступны.

3.2.2 Независимые стратегии

Кроме вдвойне однородных стратегий, мы рассматриваем такие стратегии, которые принимают решение о каждом тренировочном объекте $i \in [n]$ независимо, со своей вероятностью p_i. Отметим, что в случае $p_1 = p_2 = ... = p_n$ стратегия остается DU. Однако в общем случае появляются некоторые изменения в оценке скорости сходимости. Существует стратегия, называемая

importance sampling, которая присваивает вероятности пропорционально константам Липшица (3) гессианов $\nabla^2 f_i(x)$ (или градиентов $\nabla f_i(x)$).

Importance sampling (ИС). Стратегия задает вероятности

$$p_i = \frac{H_i}{\sum\limits_{i=1}^{n} H_i} \tag{13}$$

3.2.3 Последовательная стратегия

Наконец, мы изучаем стратегию, состоящую в последовательном проходе по перемешанным данным:

Последовательная стратегия. На первой итерации Алгоритма 1 фиксируется случайная перестановка π обучающей выборки $[n]$ и размер батча τ. Далее на итерации номер k берутся объекты с номерами:

$$\pi(k+1 \bmod n), \pi(k+2 \bmod n), ..., \pi(k+\tau \bmod n). \tag{14}$$

А если $(k \bmod n) + \tau > n$, то берутся все объекты с номерами между $k \bmod n$ и n, то есть на данном шаге размер батча может быть меньше τ.

3.3 Скорость сходимости

3.3.1 Оценки для вдвойне однородного семплинга

Чтобы установить скорость сходимости Алгоритма 1 с вдвойне однородным семплингом (2), обратимся к Лемма 3, поскольку это первое место в доказательстве Алгоритма 1, которое зависит от стратегии семплинга. Она обобщается на произвольный DU-семлпинг.

Лемма 4. *Итерации Алгоритма 1 с вдвойне однородным семплингом удовлетворяют*

$$\mathbb{E}_k[\mathcal{W}^{k+1}] = p\mathbb{E}_k[\|x^{k+1} - x^*\|^2] + (1-p)\,\mathcal{W}^k, \tag{15}$$

где $p = \frac{\mathbb{E}[|\hat{S}|]}{n}$.

Доказательство. Свойство вдвойне однородности влечет, что все объекты имеют одинаковую вероятность $p = \frac{\mathbb{E}[|\hat{S}|]}{n}$ быть включенными в выбранное множество $\hat{S}$. Таким образом, в силу линейности математического ожидания, получаем требуемое. □

Тогда итоговая оценка имеет ту же форму, что и представлено в [10]:

$$\mathbb{E}_k[\mathcal{W}^{k+1}] \leqslant \left(1 - p + p\left(\frac{H}{2\mu}\right)^2 \mathcal{W}^k\right)\mathcal{W}^k \tag{16}$$

в общем случае, а для $\forall i \ \|w_i^0 - x^*\| \leqslant \frac{\mu}{H}$:

$$\mathbb{E}_k[\mathcal{W}^{k+1}] \leqslant \left(1 - \frac{3}{4}p\right)\mathcal{W}^k. \tag{17}$$

Значит, если зафиксировать ожидаемый размер батча $\mathbb{E}[|\hat{S}|] = \tau$, то скорость сходимости такая же, как и у τ-nice семплинга.

3.3.2 Независимые и последовательная стратегии

Теперь рассмотрим общий случай выбора каждого объекта независимо с вероятностью p_i.

Предложение 1. *На каждой конкретной итерации Алгоритма 1 при фиксированном матожидании размера батча $\mathbb{E}_k[|\hat{S}^k|] = \tau$ минимальное значение в оценке $\mathbb{E}_k[\mathcal{W}^{k+1}]$ достигается, если задать $p_i = 1$ для некоторых $i \in [n]$, возможно $p_i \in (0, 1)$ для одного i, и $p_i = 0$ для остальных.*

Доказательство. Мы можем обобщить оценку из Леммы 4 на наш случай.

$$n \cdot \mathbb{E}_k[\mathcal{W}^{k+1}] = \sum_{i=1}^{n} p_i \|x^{k+1} - x^*\|^2 + \sum_{i=1}^{n} (1 - p_i) \|w_i^k - x^*\|^2. \tag{18}$$

Учитывая $\tau = \mathbb{E}_k[|\hat{S}^k|] = \sum\limits_{i=1}^{n} p_i$, имеем

$$n \cdot \mathbb{E}_k[\mathcal{W}^{k+1}] = \left(\tau\|x^{k+1} - x^*\|^2 + n \cdot \mathcal{W}^k\right) - \sum_{i=1}^{n} p_i \|w_i^k - x^*\|^2. \tag{19}$$

То есть задача состоит в максимизации $\sum\limits_{i=1}^{n} p_i \|w_i^k - x^*\|^2$ при следующем условии:

$$\begin{cases} \sum\limits_{i=1}^{n} p_i = \tau \\ 0 \leqslant p_i \leqslant 1 \ \forall i \end{cases} \tag{20}$$

Решением данной задачи линейного программирования является вершина многогранника, задаваемого ограничениями (20). Оптимальная вершина имеет максимальную проекцию своего координатного вектора $(p_1, ..., p_n)^\top$ на вектор $(\|w_i - x^*\|^2, ..., \|w_n - x^*\|^2)$. Это та вершина, коэффициенты которой, отвечающие наибольшим $\|w_i - x^*\|^2$ установлены равными 1, отвечающие наименьшим $\|w_i - x^*\|^2$ установлены равными 0, и одна координата имеет промежуточное значение, если τ не целое. □

В таком случае мы видим, что имеет смысл обновлять те векторы параметров w_i^k, которые не обновлялись дольше других. Это значит, что оптимальная независимая стратегия может быть "приближена"последовательной стратегией. Данное предложение иллюстрируется в секции экспериментов (4.3).

3.3.3 Замечания о Importance Sampling

В некоторых методах, в частности, в традиционном SGD, преимущество применения ИС состоит в том, чтобы заменить оценки, основанные на H_{max} - максимуме из констант Липшица H_i каждой f_i - оценками, обусловленными $\overline{H}$ - среднему значению всех H_i. Данное улучшение достигается благодаря рассмотрению задачи, похожей на оригинальную (1), но где каждая f_i масштабирована обратно пропорционально H_i для того, чтобы сохранить математическое ожидание $f(x)$ и $\nabla f(x)$ равными истинным значениям f и $\nabla f(x)$.

Однако Алгоритм 1 имеет немного другую природу. Это значит, что направление шага оценивается по информации не только лишь от тех объектов, которые включены в батч на текущей итерации. Оно вычисляется, учитывая всю информацию с предыдущего шага, с небольшими изменениями на текущем шаге. Следовательно, вышеупомянутая техника не может быть применена напрямую. В таком случае оказывается очень трудно установить какие-либо конкретные оценки для ИС применительно к Алгоритму 1.

4 Эксперименты

Мы проводим эксперименты на данных, полученных с помощью функции $\mathrm{make\_classification}$ из библиотеки $\mathrm{sklearn}$ с 500 объектами и 5 признаками. Код эксперимента доступен в репозитории проекта (ссылка). Оптимизируем логистическую функцию с l_2-регуляризацией: $f_i(x) = \log(1 + \exp(-b_i \cdot a_i^T x)) + \frac{\lambda}{2}\|x\|_2^2$, где $x, a_1, ..., a_n \in \mathbb{R}^d$, $b_i = \pm 1$ и константа регуляризации λ равна 0.01. На каждой картинке представлено два графика. Первый отражает траекторию величины $\mathcal{W}^k = \frac{1}{n}\sum\limits_{i=1}^{n} \|w_i^k - x^*\|^2$, а второй - историю невязки по функции. Ось x - количество полных проходов по датасету.

4.1 Вдвойне однородные стратегии

Мы видим, что все вдвойне однородные стратегии с одинаковым математическим ожиданием размера батча демонстрируют одинаковую скорость сходимости.

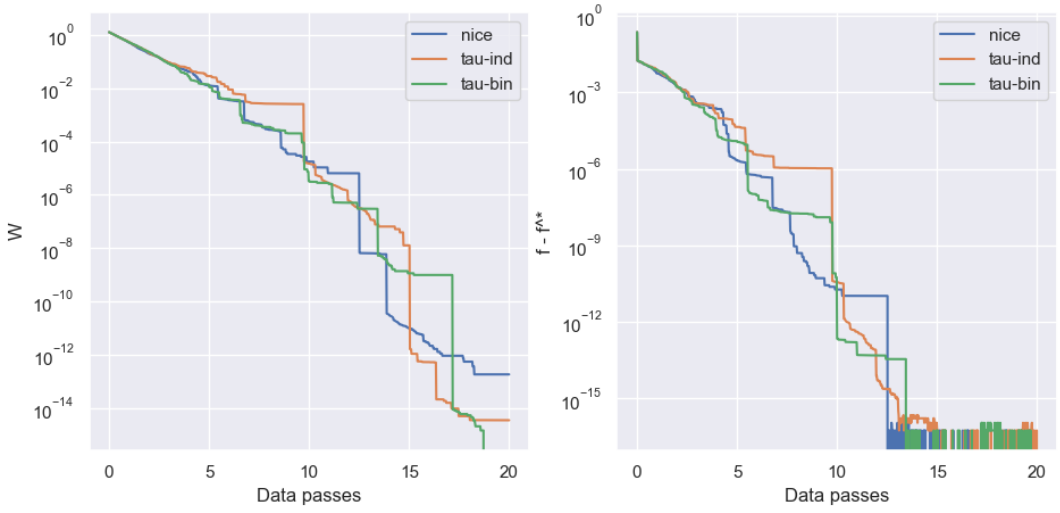

Рис. 1: Сравнение стратегий τ-nice, τ-независимой и τ-биномиальной

4.2 Importance Sampling

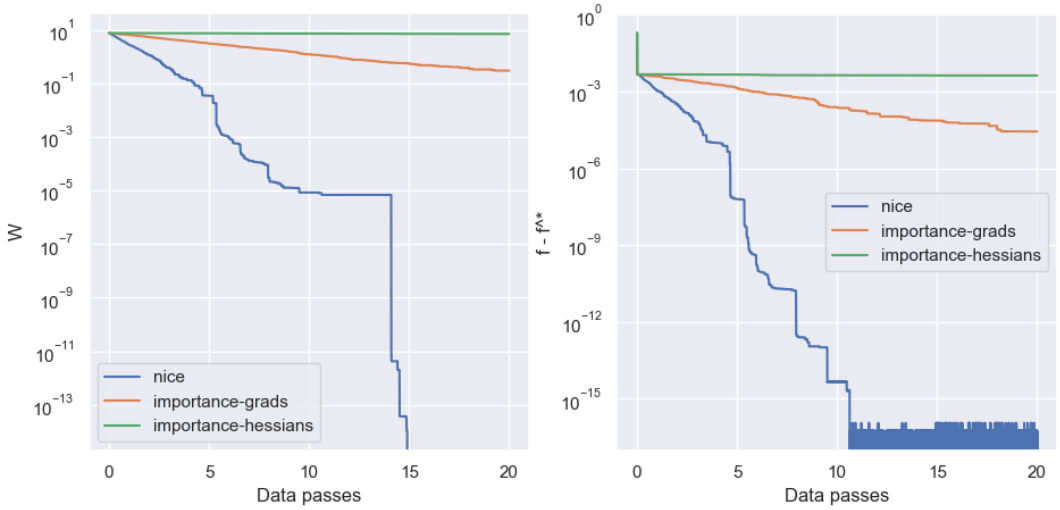

Рис. 2: Сравнение τ-nice семплинга и Importance Sampling

Мы применяем ИС, основанный как на липшицевости гессианов, так и градиентов. Для задачи логистической регрессии константы Липшица равны $H_i = \frac{1}{10}\|a_i\|^3$ и $L_i = \frac{1}{4}\|a_i\|^2$ соответственно. Видно, что для рассматриваемой задачи ИС, примененный к Алгоритму 1, не дает практических улучшений.

4.3 Последовательная стратегия

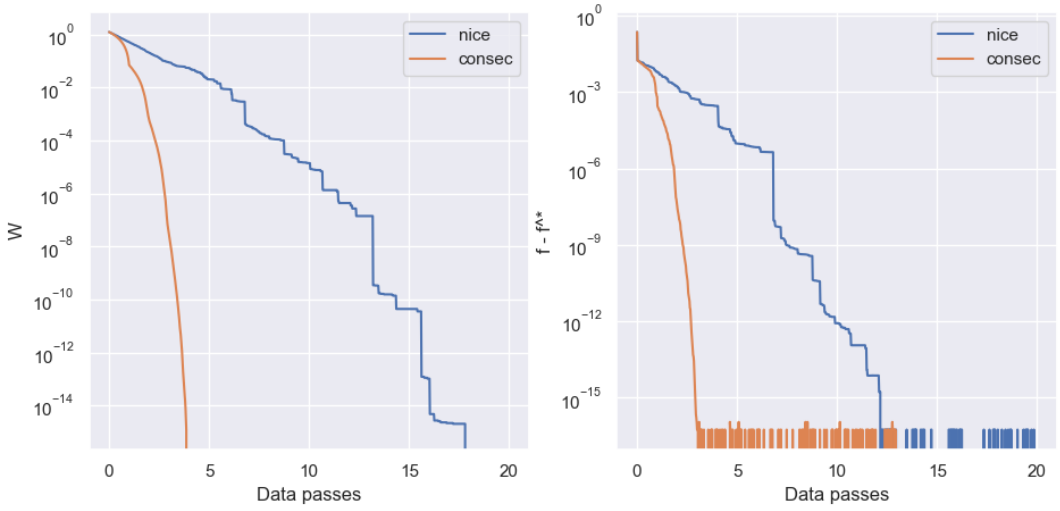

Рис. 3: Сравнение τ-nice и последовательной стратегии

В данном случае имеется практическое улучшение. Последовательная стратегия превосходит базовую τ-nice. Тем не менее, теоретическое обоснование этого факта остается неясным.

5 Заключение

Таким образом, мы приводим сравнение различных стратегий семплинга, применяемых к Стохастическому методу Ньютона (1). Во-первых, мы обощаем линейную оценку скорости сходимости на все вдвойне однородные стратегии и устанавливаем практическую эквивалентность наиболее популярных представителей данного класса (Лемма 4, Уравнение 16, Рис. 4.1). Во вторых, мы получаем, что Importance Sampling маловероятно будет превосходить стандартную стратегию τ-nice, а на практике сильно уступает (Рис. 4.2). Наконец, улучшения появляются при использовании последовательной стратегии (Рис. 4.3). Это, безусловно, положительный эффект, и он требует дальнейшего изучения.

References

[1] Raghu Bollapragada, Richard H Byrd, and Jorge Nocedal. Exact and inexact subsampled Newton methods for optimization. *IMA Journal of Numerical Analysis*, 39(2):545–578, 04 2018.

[2] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent, 2019.

[3] Robert Gower, Dmitry Kovalev, Felix Lieder, and Peter Richtarik. Rsn: Randomized subspace newton. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[4] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtarik. Sgd: General analysis and improved rates. 2019.

[5] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. Sgd: General analysis and improved rates. In *International conference on machine learning*, pages 5200–5209. PMLR, 2019.

[6] Filip Hanzely and Peter Richtárik. One method to rule them all: Variance reduction for data, parameters and many new methods, 2019.

[7] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[8] Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Global linear convergence of newton's method without strong-convexity or lipschitz gradients, 2018.

[9] Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1895–1904. PMLR, 06–11 Aug 2017.

[10] Dmitry Kovalev, Konstantin Mishchenko, and Peter Richtárik. Stochastic newton and cubic newton methods with simple local linear-quadratic rates. *arXiv preprint arXiv:1912.01597*, 2019.

[11] Huikang Liu, Xiaolu Wang, Jiajin Li, and Anthony Man-Cho So. Low-cost lipschitz-independent adaptive importance sampling of stochastic gradients. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2150–2157, 2021.

[12] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik. Random reshuffling: Simple analysis with vast improvements. *Advances in Neural Information Processing Systems*, 33:17309–17320, 2020.

[13] Yurii Nesterov. *Introductory lectures on convex optimization: a basic course/ by Yurii Nesterov*. Applied optimization, 87. Kluwer Academic Publishers, Boston, 2004.

[14] Lam Nguyen, PHUONG HA NGUYEN, Marten van Dijk, Peter Richtarik, Katya Scheinberg, and Martin Takac. SGD and hogwild! Convergence without the bounded gradients assumption. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3750–3758. PMLR, 10–15 Jul 2018.

[15] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156:433–484, 2016.

[16] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951.

[17] Nicolas Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[18] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. Stochastic cubic regularization for fast nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

[19] Junyu Zhang, Lin Xiao, and Shuzhong Zhang. Adaptive stochastic variance reduction for subsampled newton method with cubic regularization. *INFORMS Journal on Optimization*, 4(1):45–64, 2022.

[20] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling, 2014.

[21] Dongruo Zhou and Quanquan Gu. Stochastic recursive variance-reduced cubic regularization methods. In *International Conference on Artificial Intelligence and Statistics*, pages 3980–3990. PMLR, 2020.

[22] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic variance-reduced cubic regularized newton methods. In *International Conference on Machine Learning*, pages 5990–5999. PMLR, 2018.