Отчет о практическом задании «Градиентные методы обучения линейных моделей».

Практикум 317 группы, ММП ВМК МГУ.

Морозов Иван Дмитриевич

Ноябрь 2023

Содержание

1 Введение

Данное практическое задание посвящено исследованию градиентного спуска и стохастического градиентного спуска на примере обучения логистической регрессии в задаче распознавания токсичности текста.

Цели исследования:

- Вывод формул градиента функции потерь для различных типов логистической регрессии.
- Исследование способов обработки и преобразований текста.
- Сравнение численного подсчёта градиента функции потерь и подсчёта по аналитической формуле, времени их работы.
- Исследование особенностей поведения градиентного спуска для задачи логистической регрессии в зависимости от параметров размера шага, начального приближения, размера подвыборки (в случае стохастического градиентного спуска).
- Исследование влияния различных способов векторизации текста.
- Выбор лучшего алгоритма модели и анализ объектов, на которых допущены ошибки.

2 Теоретическая часть

2.1 Вывод формулы градиента функции потерь для задачи бинарной логистической регрессии

$$
\begin{aligned}
dQ(w) &= d(\frac{1}{l}\sum_{i=1}^{l}\log(1+\exp\left(-y_i\langle w,x_i\rangle\right)))+\frac{\lambda}{2}\|w\|_2^2) \\
&= \frac{1}{l}\sum_{i=1}^{l} d(\log(1+\exp\left(-y_i\langle w,x_i\rangle\right)))+d(\frac{\lambda}{2}\|w\|_2^2) \\
&= \frac{1}{l}\sum_{i=1}^{l}\frac{1}{1+\exp\left(-y_i\langle w,x_i\rangle\right)}\cdot\exp\left(-y_i\langle w,x_i\rangle\right)\cdot d(-y_i\cdot\langle w,x_i\rangle)+\frac{\lambda}{2}d(\langle w,w\rangle) \\
&= \frac{1}{l}\sum_{i=1}^{l}\frac{1}{1+\exp\left(-y_i\langle w,x_i\rangle\right)}\cdot\exp\left(-y_i\langle w,x_i\rangle\right)\cdot(-y_i)\cdot\langle x_i,dw\rangle+\lambda\langle w,dw\rangle \\
&= \langle\frac{1}{l}\sum_{i=1}^{l}\frac{-y_i\cdot\exp\left(-y_i\langle w,x_i\rangle\right)}{1+\exp\left(-y_i\langle w,x_i\rangle\right)}\cdot x_i+\lambda w,dw\rangle \\
&= \langle-\frac{1}{l}\sum_{i=1}^{l}\frac{y_ix_i}{1+\exp\left(y_i\langle w,x_i\rangle\right)}+\lambda w,dw\rangle.
\end{aligned}
\tag{1}
$$

$$\nabla Q(w) = -\frac{1}{l}\sum_{i=1}^{l}\frac{y_i x_i}{1+\exp\left(y_i\langle w, x_i\rangle\right)} + \lambda w. \tag{2}$$

2.2 Вывод формулы градиента функции потерь для задачи многоклассовой (мультиномиальной) классификации

$$a(x) = \arg\max_{i=1,..,k}\langle w_i, x\rangle, x, w_i \in \mathbb{R}^n. \tag{3}$$

$$P(y_i|(x,w)) = \frac{\exp\langle w_{y_i}, x\rangle}{\sum_{j=1}^{k}\exp\langle w_j, x\rangle}. \tag{4}$$

$$Q(w) = \frac{1}{l}\sum_{i=1}^{l}\log(\frac{\sum_{j=1}^{k}\exp\langle w_j, x_i\rangle}{\exp\langle w_{y_i}, x_i\rangle}) + \frac{\lambda}{2}\sum_{j=1}^{k}\|w_j\|_2^2, \tag{5}$$

$$\begin{aligned} d_j Q(w) &= d_j(\frac{1}{l}\sum_{i=1}^{l}\log(\frac{\sum_{q=1}^{k}\exp\langle w_q, x_i\rangle}{\exp\langle w_{y_i}, x_i\rangle}) + \frac{\lambda}{2}\sum_{i=1}^{k}\|w_i\|_2^2) \\ &= \frac{1}{l}\sum_{i=1}^{l} d_j(\log(\sum_{q=1}^{k}\exp\langle w_q, x_i\rangle) - \langle w_{y_i}, x_i\rangle) + \frac{\lambda}{2}\sum_{i=1}^{k} d_j(\|w_i\|_2^2) \\ &= \frac{1}{l}\sum_{i=1}^{l}\frac{\exp\langle w_j, x_i\rangle}{\sum_{q=1}^{k}\exp\langle w_q, x_i\rangle}\langle x_i, dw_j\rangle - \langle x_i, dw_{y_i}\rangle[y_i = j]) + \lambda\langle w_j, dw_j\rangle \\ &= \langle\frac{1}{l}\sum_{i=1}^{l}(\frac{\exp\langle w_{y_i}, x\rangle}{\sum_{j=1}^{k}\exp\langle w_j, x\rangle} - [y_i = j])x_i + \lambda w_j, dw_j\rangle \end{aligned} \tag{6}$$

$$\nabla_j Q(w) = \frac{1}{l}\sum_{i=1}^{l}(\frac{\exp\langle w_{y_i}, x\rangle}{\sum_{j=1}^{k}\exp\langle w_j, x\rangle} - [y_i = j])x_i + \lambda w_j. \tag{7}$$

2.3 Мультиномиальная и бинарная логистические регрессии

$$|Y| = \{-1, 1\} \tag{8}$$

$$\begin{aligned} Q(w) &= \frac{1}{l}\sum_{i=1}^{l}\log(\frac{\sum_{z\in Y}\exp\langle w_z, x_i\rangle}{\exp\langle w_{y_i}, x_i\rangle}) + \frac{\lambda}{2}\sum_{y\in Y}\|w_y\|_2^2 \\ &= \frac{1}{l}\sum_{i=1}^{l}\log\frac{\exp\langle w_{-1}, x_i\rangle + \exp\langle w_1, x_i\rangle}{\exp\langle w_{y_i}, x_i\rangle} + \frac{\lambda}{2}\sum_{y\in Y}\|w_y\|_2^2 \\ &= \frac{1}{l}\sum_{i=1}^{l}\log(1 + \frac{\exp\langle w_{-y_i}, x_i\rangle}{\exp\langle w_{y_i}, x_i\rangle}) + \frac{\lambda}{2}\sum_{y\in Y}\|w_y\|_2^2 \\ &= \frac{1}{l}\sum_{i=1}^{l}\log(1 + \exp\left(\langle w_{-y_i}, x_i\rangle - \langle w_{y_i}, x_i\rangle\right) + \frac{\lambda}{2}\sum_{y\in Y}\|w_y\|_2^2 \\ &= \frac{1}{l}\sum_{i=1}^{l}\log(1 + \exp\left(-\langle w_{y_i} - w_{-y_i}, x_i\rangle\right) + \frac{\lambda}{2}\sum_{y\in Y}\|w_y\|_2^2 \\ &= \frac{1}{l}\sum_{i=1}^{l}\log(1 + \exp\left(-y_i\langle w, x_i\rangle\right) + \frac{\lambda}{2}\|w\|_2^2 \end{aligned} \tag{9}$$

3 Эксперимент №1. Предварительная обработка текста

Загружен датасет с kaggle-соревнования. Данные являются текстовыми строками, представляющими собой комментарии из раздела обсуждений английской Википедии. Решается задача бинарной классификации: установление факта токсичности комментария. В рамках данного эксперимента все строки были приведены к нижнему регистру, оставлены только буквы и цифры, остальные символы заменены на пробелы.

4 Эксперимент №2. Векторизация текста

Вся выборка была преобразована в разреженную матрицу `scipy.sparse.matrix`. Номер строки отвечает за документ, номер столбца за конкретное слово, на их пересечении записано количество слов в данном документе. Параметры преобразования в матрицу были подобраны так, что не все слова были учтены (наиболее редкие исключены из рассмотрения). Это необходимо для снижения размерности признакового просранства и ускорения проведения экспериментов: достигнуто значение 568 различных слов.

5 Эксперимент №3. Аналитический и численный способы подсчёта градиента функции потерь.

Аналитический способ подсчёта градиента функции заключается в использовании специальной формулы для нахождения градиента, численный способ заключается в приближенном оценивании градиента, для этого достаточно знания только формулы функционала.

Оптимизируемый функционал:

$$Q(w) = \frac{1}{l}\sum_{i=1}^{l}\log(1+\exp\left(-y_i\langle w, x_i\rangle\right)) + \frac{\lambda}{2}\|w\|_2^2, \tag{10}$$

градиент оптимизируемого функционала:

$$\nabla Q(w) = -\frac{1}{l}\sum_{i=1}^{l}\frac{y_i x_i}{1+\exp\left(y_i\langle w, x_i\rangle\right)} + \lambda w. \tag{11}$$

Пусть e_i - базисный вектор, ε - небольшое смещение. Тогда формула численного подсчёта градиента оптимизируемого функционала $f(w)$:

$$[\nabla Q(w)]_i = \frac{Q(x+\varepsilon e_i) - Q(x)}{\varepsilon}, \tag{12}$$

В рамках данного эксперимента градиент функционала вычисляется двумя способами, полученные векторы-значения сравниваются с помощью функций расстояний Ланса-Уильямса:

$$\rho(x,z) = \frac{\sum_{i=1}^{D}|x^i - z^i|}{\sum_{i=1}^{D}|x^i + z^i|}, \tag{13}$$

и Канберра:

$$\rho(x,z) = \frac{1}{D}\sum_{i=1}^{D}\frac{|x^i - z^i|}{|x^i| + |z^i|}, \tag{14}$$

эти функции учитывают относительную разницу векторов, а не абсолютную, также замеряется время, за которое был произведён аналитический и численный подсчёт. $\varepsilon = 10^{-7}$, матрица X взята из предыдущего эксперимента, вектор ответов y из исходного датасета (изменены метки классов: вместо 0 и 1 были подставлены значения -1 и 1 соответственно), w инициализирован нулями.

Получены следующие результаты: подсчёт по аналитической формуле занял **0.05 сек**, численно - **2.63 сек**. Расстояние Ланса-Уильямса для полученных значений составило $1.53 * 10^{-6}$, Канберра $8.93 * 10^{-6}$.

5.1 Выводы

Численный способ подсчёта градиента функции потерь имеет высокую точность, которую можно увеличить, уменьшив ε. Но он сильно уступает аналитическому способу по времени вычислений, более того, при увеличении количества признаков разница во времени, очевидно, будет расти. Таким образом, численный способ позволяет знать только формулу функционала и работает значительно медленнее, чем аналитический метод, требующий знания аналитической формулы для нахождения градиента функционала.

6 Эксперимент №4. Исследование поведения градиентного спуска для задачи логистической регрессии в зависимости от параметров шага и начального приближения

Эксперимент заключается в анализе поведения функции потерь и точности по мере совершения итераций. При этом могут варьироваться параметры α, β и начальное приближение в формуле расчёта весов для итерации k:

$$w^{(k+1)} = w^{(k)} - \eta_k \nabla_w Q(w), \tag{15}$$

где $\eta_k = \frac{\alpha}{k^\beta}$ - *learning rate*. Для более наглядного восприятия результатов на каждом графике фиксируется два из указанных параметров, а третий изменяет значения. В рамках данного эксперимента коэффициент регуляризации взят за 1.

Будут рассмотрены 4 способа начального приближения вектора весов $w^{(0)}$:

- $w^{(0)} \sim \text{Normal}(0, 1)$
- $w^{(0)} \sim \text{Uniform}[-1, 1]$
- $w^{(0)} = 0$
- $w_j^{(0)} = \frac{\langle f_j, y \rangle}{\langle f_j, f_j \rangle}$

6.1 Перебор α при фиксированном $\beta = 0.3$ и начальном приближении $w^{(0)} \sim \text{Normal}(0, 1)$

Начальное приближение $w^{(0)} \sim \text{Normal}(0, 1)$, $\beta = 0.3$. Рассматриваются первые 50 итераций. По рис. 1 видно, что для $\alpha < 1$ функция потерь убывает с увеличением итерации. Для $\alpha >= 1$ на начальных итерациях функция потерь увеличивается, далее, с увеличением номера итерации начинает осциллировать. Это связано с тем, что величина *learning rate* очень большая, поскольку α находится в числителе, а в знаменателе β в степени, меньшей 1. Но заметна тенденцию к уменьшению функции потерь.

По рис. 2 видно, что для $\alpha >= 1$ точность является осциллирующей функций с тенденцией приближения к значению 0.7 с увеличением номера итерации, что подтверждает сказанное выше. Для $\alpha < 1$ точность гораздо быстрее приближается к асимптоте 0.7, из чего можно сделать вывод, что $\alpha < 1$ - наболее оптимальные значения, они не позволяют *learning rate* быть слишком большим.

6.2 Перебор β при фиксированном $\alpha = 0.3$ и начальном приближении $w^{(0)} \sim \text{Normal}(0, 1)$

Анализируя рис. 3, можно убедиться, что если $\beta >= 1$, то функция потерь уменьшается с увеличением итерации, алгоритм сходится, но когда $\beta < 1$ значение функции потерь остаётся примерно на

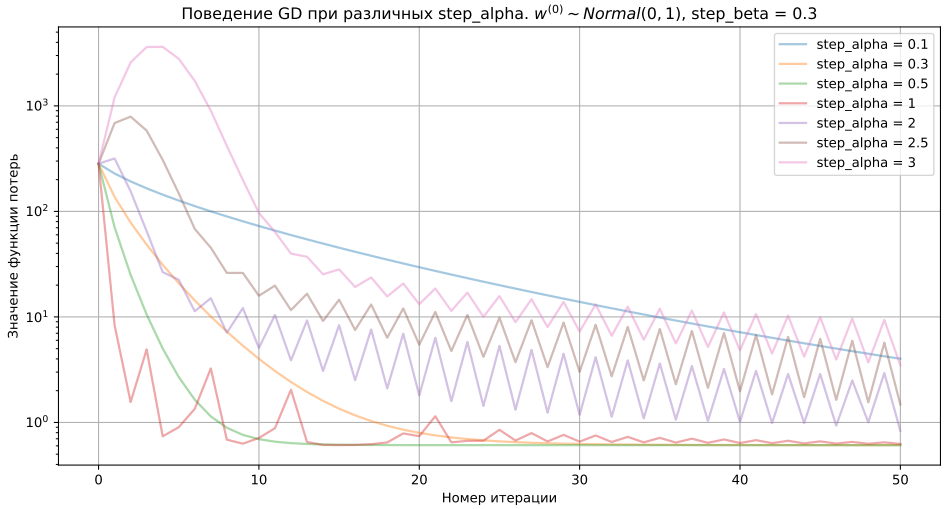

Рис. 1: Анализ поведения функции потерь для различных step_alpha

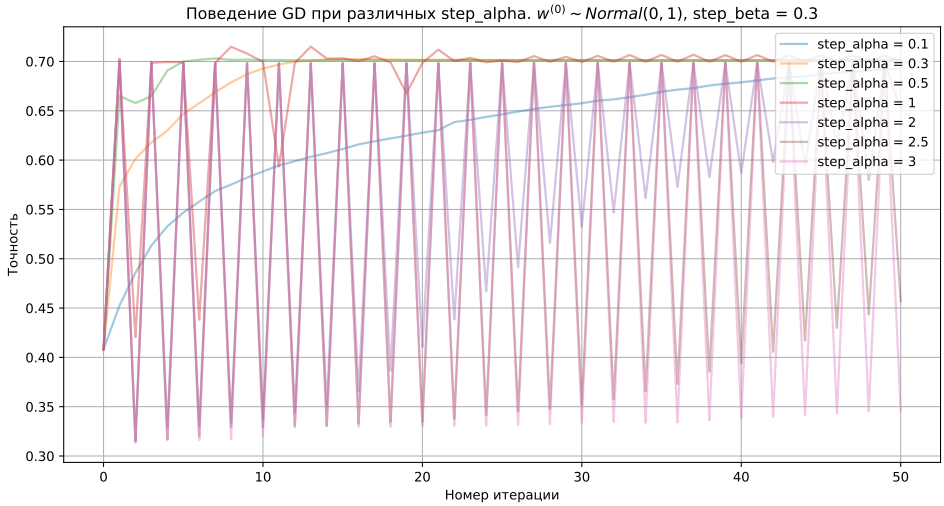

Рис. 2: Анализ поведения точности для различных step_alpha

одном и том же уровне, поскольку *learning rate* очень малый, и алгоритм будет сходиться очень долго или не сойдётся никогда.

Выводы, сделанные из рис. 3, на котором анализируется функция потерь, подтверждаются рис. 4 с отображением поведения точности с увеличением итерации: для $\beta < 1$, задающих малый *learning rate*, точность не может приблизиться к своему потенциальному максимуму, поскольку алгоритм не может сойтись быстро (либо не может сойтись вообще).

6.3 Перебор начальных приближений при фиксированных $\alpha = 0.3$ и $\beta = 0.3$

По рис. 5 и рис. 6 видно, что выбор начального приближения не может испортить сходимость алгоритма, но может повлиять на скорость сходимости. Например, выбор $w^{(0)} = 0$ на первых итерациях даёт малое значение функции потерь и большое значение точности, близкое к предельному (т. е. при больших значениях номеров итераций). По графикам видно, что к 25-30 итерации выбор начального приближения уже не играет существенной роли, поскольку функции потерь и точности приблизительно равны для всех рассматриваемых 4 случаев.

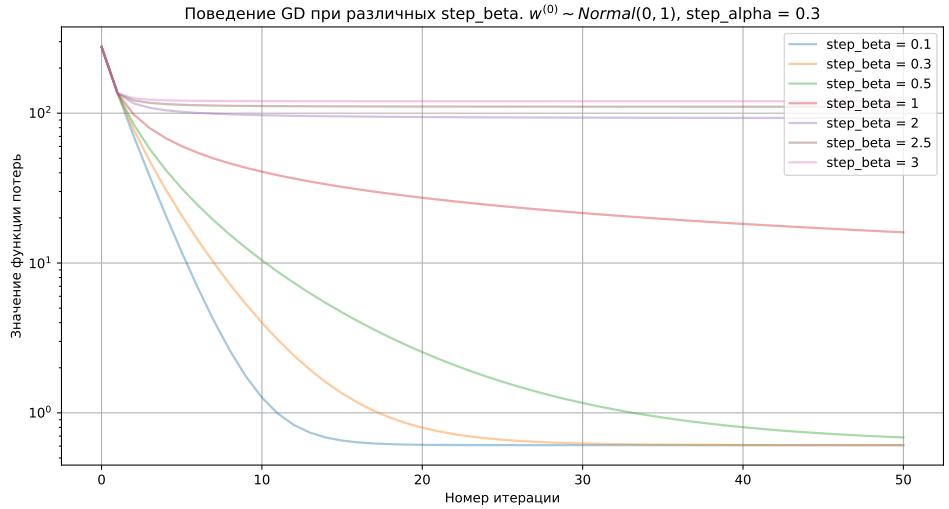

Рис. 3: Анализ поведения функции потерь для различных step_beta

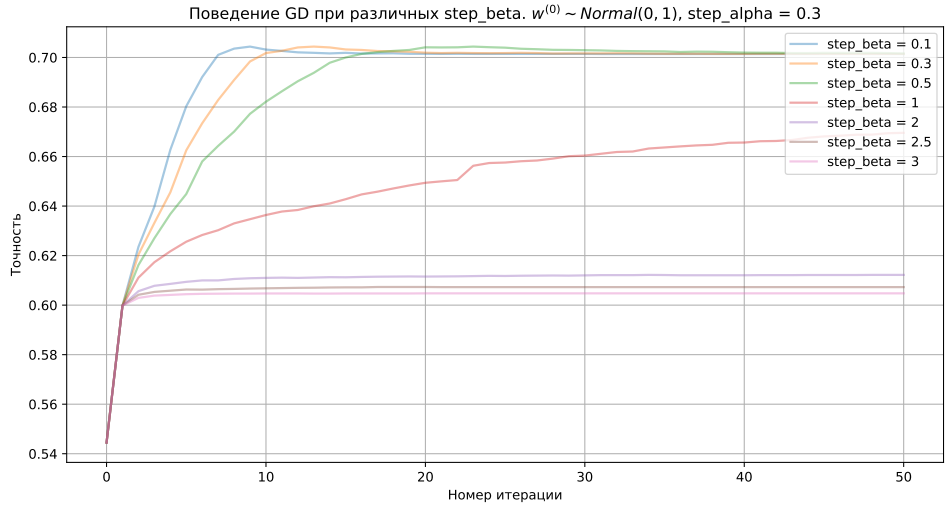

Рис. 4: Анализ поведения точности для различных step_beta

7 Эксперимент №5. Исследование поведения стохастического градиентного спуска для задачи логистической регрессии в зависимости от параметров шага, начального приближения и размера батча

Эксперимент заключается в анализе поведения функции потерь и точности по мере совершения итераций. При этом могут варьироваться параметры α, β и начальное приближение в формуле расчёта весов для итерации k и размер батча:

$$w^{(k+1)} = w^{(k)} - \eta_k \nabla_w Q(w), \tag{16}$$

где $\eta_k = \frac{\alpha}{k^\beta}$ - *learning rate*. Для более наглядного восприятия результатов на каждом графике фиксируется три из указанных параметров, а четвёртый изменяет значения. В рамках данного эксперимента коэффициент регуляризации взят за 1.

Будут рассмотрены 4 способа начального приближения вектора весов $w^{(0)}$ (аналогично предыдущему эксперименту).

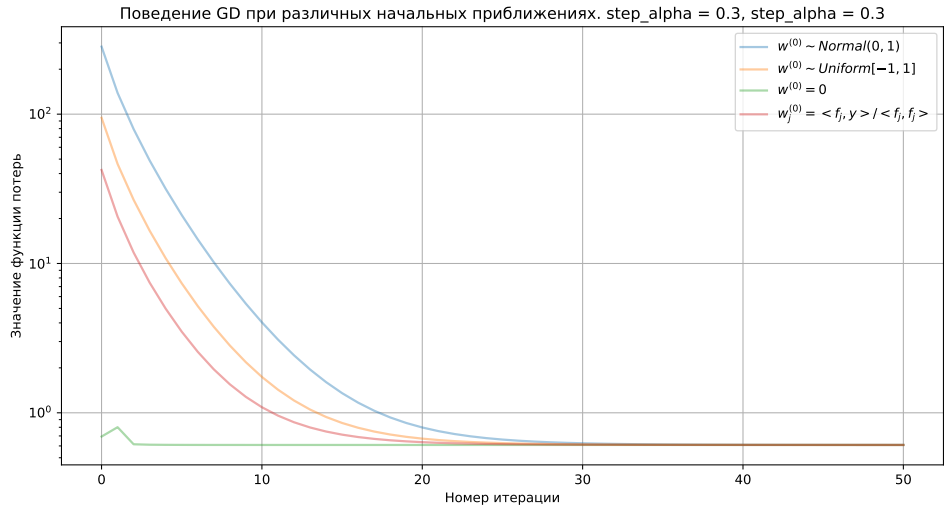

Рис. 5: Анализ поведения функции потерь для различных начальных приближений

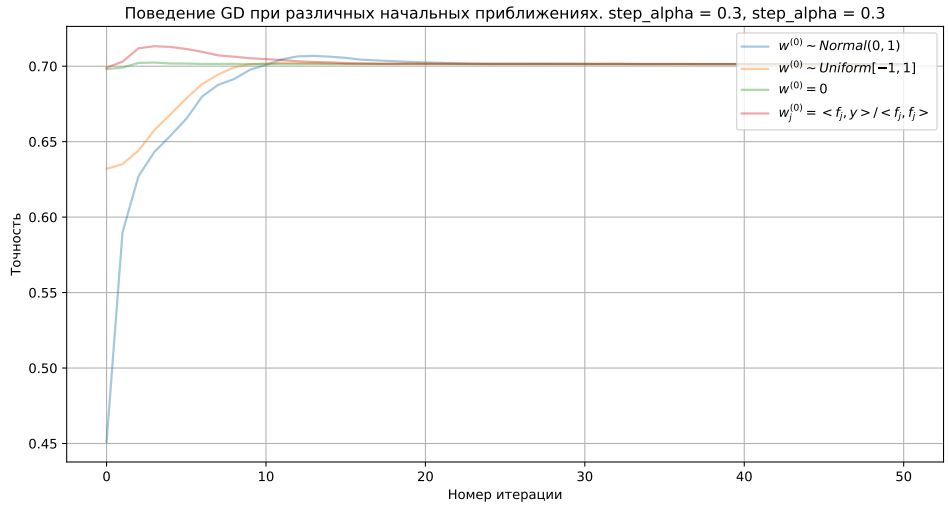

Рис. 6: Анализ поведения точности для различных начальных приближений

7.1 Перебор α при фиксированном $\beta = 0.3$, начальном приближении $w^{(0)} \sim \text{Normal}(0, 1)$ и размере батча 500

Рассматриваются рис. 7 и рис. 8. Для значений α, меньших единицы, заметна лучшая сходимость алгоритма по сравнению с $\alpha >= 1$: значения функции потерь при увеличении номера эпохи совершают меньше скачков, аналогично, точность "проседает" реже и на меньшие величины, чего нельзя сказать об $\alpha >= 1$. Однако, заметно, что даже при рассмотрении 80 эпох точность может "проседать" так же сильно, как она снижалась на 10, 39, 50 эпохах. Можно сделать вывод, что $\alpha < 1$ - наиболее оптимальные значения, они задают меньший *learning rate*.

7.2 Перебор β при фиксированном $\alpha = 0.3$, начальном приближении $w^{(0)} \sim \text{Normal}(0, 1)$ и размере батча 500

Рассматриваются рис. 9 и рис. 10, и одновременно анализируется точность и значения функции потерь. По графикам видно, что для любых β на начальных эпохах функция потерь очень большая, соответсвенно, точность низкая. Так же, как и в случае с α есть тенденция к самопроизвольному резкому снижению точности (повышению значений функции потерь для отдельных эпох), однако этот

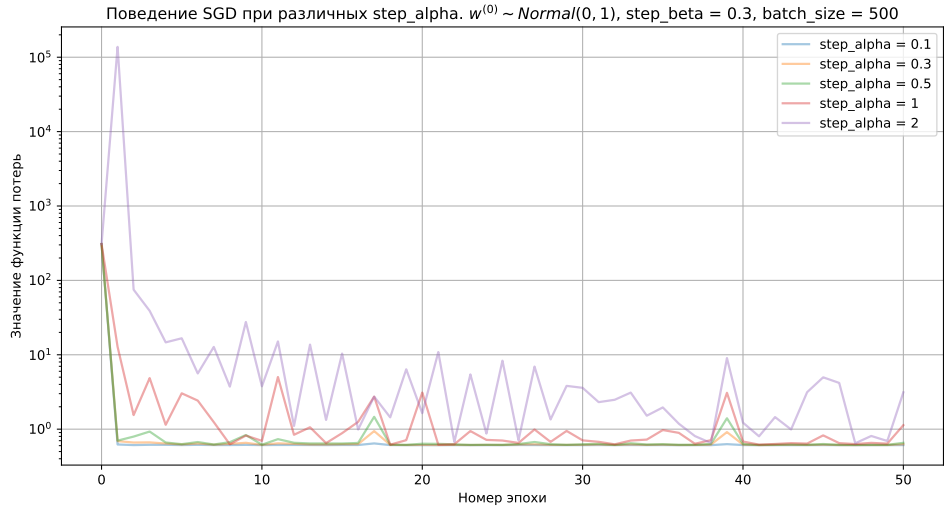

Рис. 7: Анализ поведения функции потерь для различных step_alpha

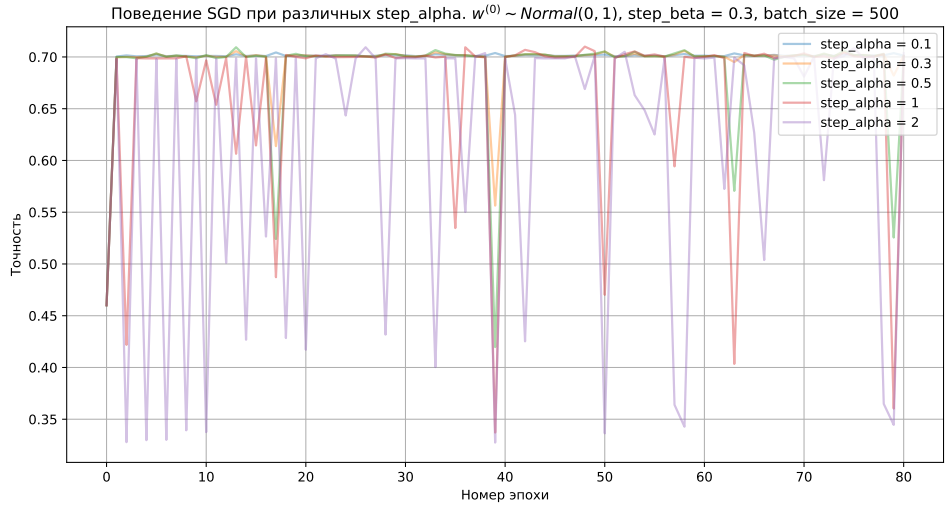

Рис. 8: Анализ поведения точности для различных step_alpha

эффект можно ослабить, выбрав β достаточно большими. Таким образом, можно снизить эффект непредвиденного поведения стохастического градиентного спуска выбором больших β, при этом на сходимость алгоритма это не повлияет.

7.3 Перебор начальных приближений при фиксированных $\alpha = 0.3$, $\beta = 0.3$ и размере батча 500

Рассматриваются рис. 11 и рис. 12, и одновременно анализируется точность и значения функции потерь. По графикам видно, что для всех 4 видов начальных приближений поведение точности одинаково, так же как и для значений функций потерь. Точно так же, как в предыдущем пункте. возникают непредвиденные понижения точности для отдельных эпох (повышения значений функций потерь). Этот эффект никак нельзя исправить выбором какого-либо из рассмотренных начальных приближений.

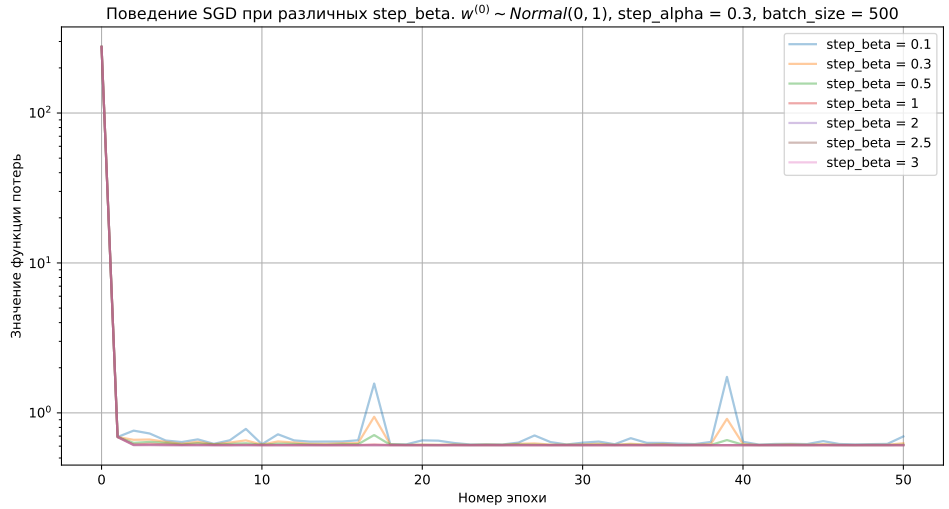

Рис. 9: Анализ поведения функции потерь для различных step_beta

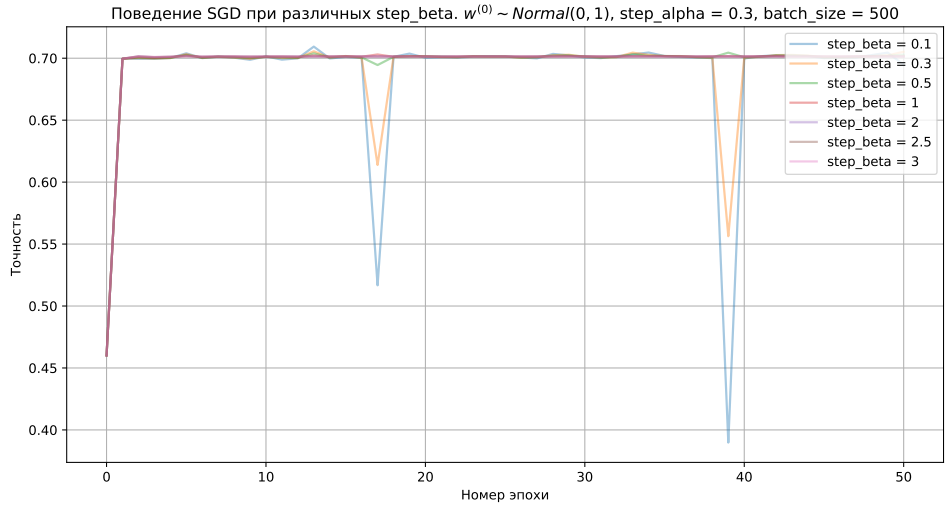

Рис. 10: Анализ поведения точности для различных step_beta

7.4 Перебор размеров батчей при фиксированных $\alpha = 0.3$, $\beta = 0.3$, $w^{(0)} \sim \text{Normal}(0, 1)$

Из рис. 13 можно сделать вывод, что он не особо полезен для анализа, поскольку функция потерь ведет себя примерно одинаково для любого размера батча, чего нельзя сказать о рис. 14, на котором видно, что если брать батчи большими, то будут возникать непредвиденные "просадки" точности для отдельных номеров эпох, чего не будет случаться в случае малых батчей.

8 Эксперимент №6. Сравнение поведения методов градиентного спуска и стохастического градиентного спуска, выводы

Метод стохастического градиентного спуска гораздо быстрее сходится, чем метод градиентного спуска, т. е. сходится за меньшее количество эпох, чем количество итераций, необходимых градиентному спуску, однако, стоит учесть факт, что одна эпоха зачастую дольше одной итерации, но даже учитывая это, можно утверждать, что метод стохастического градиентного спуска быстрее. Анализируя поведение оптимизируемого функционала, можно заметить, что его изменение более стабильно и предсказуемо для метода градиентного спуска, чем для метода стохастического градиентного спуска.

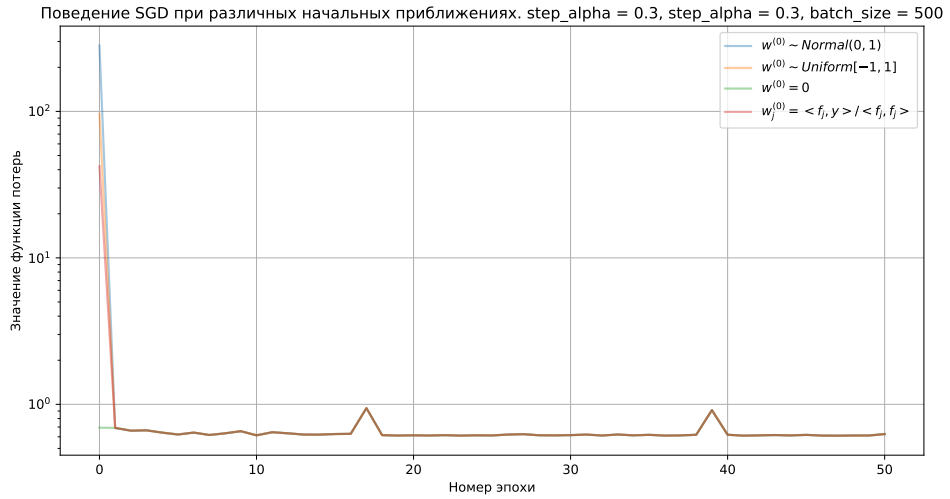

Рис. 11: Анализ поведения функции потерь для различных начальных приближений

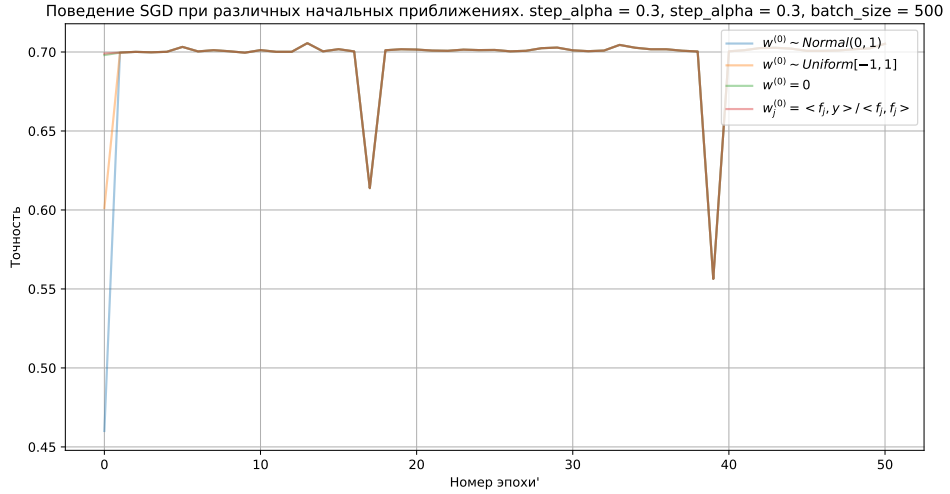

Рис. 12: Анализ поведения точности для различных начальных приближений

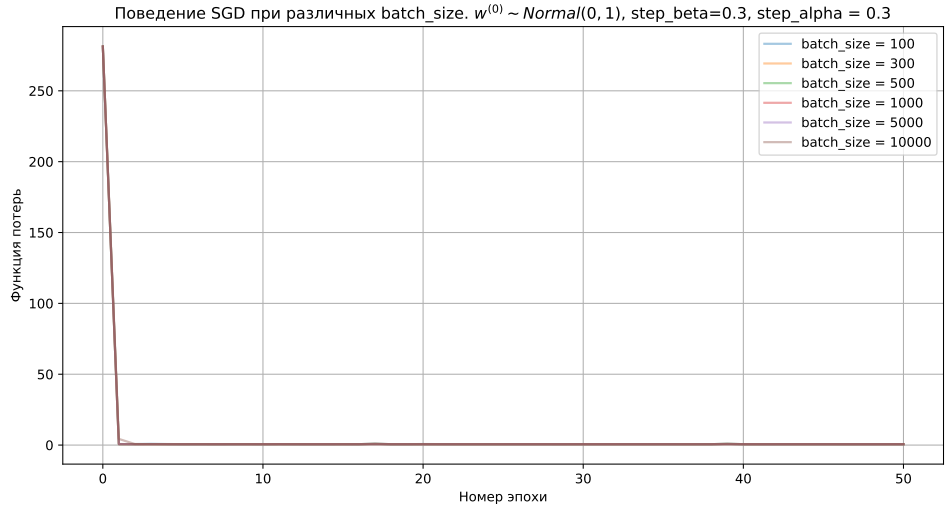

Рис. 13: Анализ поведения функции потерь для различных размеров батчей

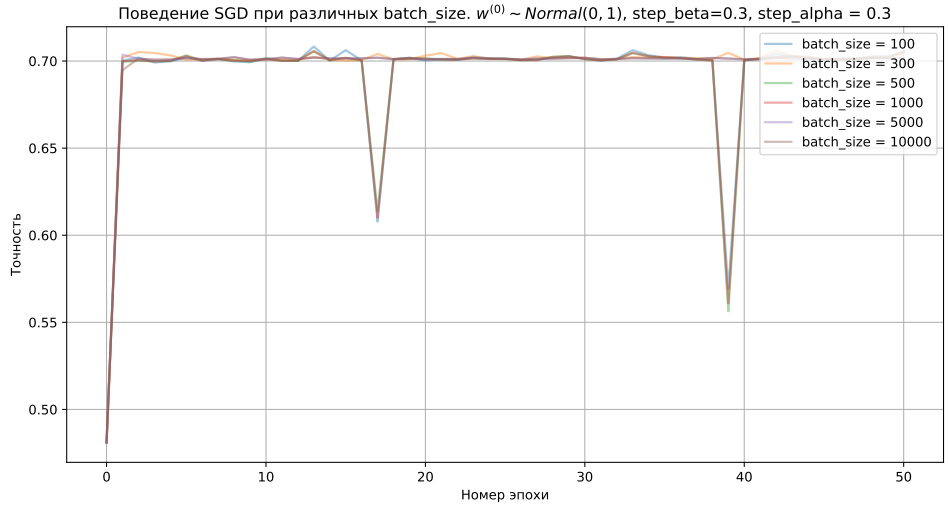

Рис. 14: Анализ поведения точности для различных размеров батчей

Характер изменения точности и оптимизируемого функционала для метода градиентного спуска зависит от начального приближения, чего нельзя сказать о стохастическом. Увеличение параметра β в случае градиентного спуска ведёт к уменьшению его сходящейся способности, для стохастического градиентного спуска это эффективнй способ стабилизировать функционал. Увеличение параметра α приводит к схожим негативным последствиям для обоих методов (дестабилизация функционала). Для метода стохастического градиентного спуска ни разу не наблюдалось ситуации несходимости метода при увеличении номеров эпох (встречались осцилляции, но их можно ликвидировать выбором большого значения параметра β).

Можно сделать вывод, что метод стохастического градиентного спуска более эффективный и надежный, чем метод градиентного спуска.

9 Эксперимент №7. Лемматизация и удаление стоп-слов

Эксперимент направлен на исследование возможности повышения точности модели и скорости её работы последством лемматизации и удаления стоп-слов. Сначала подбирается коэффициент регуляризации по логарифмической сетке для имеющихся данных (алгоритм применяется к данным, полученным в результате 1 и 2 эксперимента). Получены следующие значения: коэффициент регуляризации 0.00075, точность 0.86, причем количество признаков равно 568. Далее применяется алгоритм лемматизации и удаления стоп-слов. Снова подбирается коэффициент регуляризации по логарифмической сетке и обнаруживается, что лучшим коэффициентом регуляризации является 5.18e-05 0.819. Значит, для модели лучшим коэффициентом регуляризации является коэффициент, близкий к 0, т. е. модель стала устойчива к переобучению и повысила свою обобщающую способность. Однако точность после лемматизации и удаления стоп-слов снизилась: для коэффициента регуляризации 5.18e-05 точность равна 0.82. Количество признаков снизилось до 437 после лемматизации и удаления стоп-слов, что говорит о повышении скорости работы алгоритма.

Таким образом, лемматизация и удаление стоп-слов снижает размерность признакового пространства, увеличивает скорость работы, повышает обобщающую способность, но снижает точность, возможно, точность снизилась на конкретных данных, а в общем случае повысилась.

10 Эксперимент №8. Сравнение представлений `BagOfWords` и `Tfidf`

`BagOfWords` учитывает количество вхождений слов в текст, в том числе и тех слов, которые не несут дискриминативной информации, поэтому вводится представление `Tfidf`, учитывающее важность при-

знаков, т. е. насколько данное слово редкое. `idf(w)` - мера редкости слова в датасете. `min_df` и `max_df` определяют порог для отбора слов, встречающихся не реже и не чаще данных значений соответсвенно. В рамках эксперимента для каждого из представлений перебираются пары значений `min_df` и `max_df` и подсчитывается точность, размерность признакового пространства, время обучения. Алгоритм применяется к данным, полученным в результате 1 и 2 эксперимента. Получены следующие результаты:

- `BagOfWords`, `min_df` = 0.01, `max_df` = 0.99, точность 0.81, время 4.23 сек, размерность 437
- `BagOfWords`, `min_df` = 0.03, `max_df` = 0.97, точность 0.78, время 2.89 сек, размерность 158
- `BagOfWords`, `min_df` = 0.2, `max_df` = 0.8, точность 0.7, время 1.22 сек, размерность 13
- `Tfidf`, `min_df` = 0.01, `max_df` = 0.99, точность 0.79, время 11.86 сек, размерность 437
- `Tfidf`, `min_df` = 0.03, `max_df` = 0.97, точность 0.78, время 1.76 сек, размерность 158
- `Tfidf`, `min_df` = 0.2, `max_df` = 0.8, точность 0.7, время 1.89 сек, размерность 13

`BagOfWords` выдал более высокую точность на данном датасете, в тексте действительно много редких и частых слов, которые не представляют собой ценную информацию (вывод сделан по изменению размерности признакового пространства). Стоит отметить, что при понижении размерности точность падала не очень сильно, что говорит о наличии в обучающей выборке большого количества неинформативных объектов. С уменьшением количества признаков время обучения моделей уменьшалось, но время обучения `Tfidf` больше, чем для `BagOfWords`, поскольку `Tfidf` использует более сложные формулы.

11 Эксперимент №9. Выбор лучшего алгоритма и анализ ошибок.

Используется лучший алгоритм, полученный в 7 эксперименте, на тестовой выборке его точность равна 0.84. Анализ объектов, на которых этот алгоритм ошибся:

- you guys are sick. Нетоксичный комментарий, который был неверно классифицирован, вероятно, из-за привязки к слову sick.
- bacteria is sicko. Нетоксичный комментарий, который был неверно классифицирован, вероятно, из-за привязки к слову sicko.
- what a shit country Токсичный комментарий, который был неверно классифицирован.
- i like sex Нетоксичный комментарий, который был неверно классифицирован, вероятно, из-за привязки к слову sex.
- sam is a fag Токсичный комментарий, который был неверно классифицирован.
- anaheim ducks Нетоксичный комментарий, который был неверно классифицирован, вероятно, из-за привязки к слову ducks.

Эти ошибки являются слествием того, что не рассматривается контекст, алгоритм распознаёт оскорбительные слова, но эти слова могут иметь и другие значения. Недостаток `BagOfWords`, что не учитывается контекст и порядок слов в предложении.

12 Эксперимент дополнительный. Добавление в признаковое пространство n-грамм.

В результате добавления n-грамм получены следующие результаты:

- n = 1 точность 0.875, время 7.38 сек, размерность 3680
- n = 2 точность 0.876, время 18.7 сек, размерность 9056
- n = 3 точность 0.875, время 18 сек, размерность 10866
- n = 4 точность 0.875, время 21 сек, размерность 11610
- n = 5 точность 0.875, время 22.7 сек, размерность 12221
- n = 6 точность 0.875, время 22.4 сек, размерность 12775
- n = 7 точность 0.875, время 27.7 сек, размерность 13286
- n = 8 точность 0.875, время 22.9 сек, размерность 13756
- n = 9 точность 0.875, время 24.4 сек, размерность 14198

При n = 2 удаётся добиться точности 0.876, но размерность признакового пространства и время обучения возросли: 9056 и 18.7 сек соответственно. n-граммы полезны, поскольку учитывают контекст.

13 Заключение

Отчёт посвящён логистической регрессии и градиентным методам обучения линейный моделей. В результате проведения экспериментов были сделаны следующие ключевые выводы:

- Аналитический способ подсёта градиента гораздо быстрее численного.
- Метод стохастического градиентного спуска устойчивее и быстрее обычного метода градиентного спуска.
- Применение методов лемматизации и удаления стоп-слов повышает размерность признакового пространства и уменьшает время обучения.
- Представления `BagOfWords` и `Tfidf` позволяют эффективно преобразовать текст.

Список литературы

[1] Воронцов К.В., «Линейные методы классификации и регрессии», 2022, Лекции по курсу «Методы машинного обучения», https://github.com/MSU-ML-COURSE/ML-COURSE-21-22/blob/main/slides/2_stream/msu21-lin-sg.03.pdf