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

Практикум 317 группы, ММП ВМК МГУ.

Марьясов Максим Михайлович.

Ноябрь 2024.

Содержание

1 Введение

Данное задание направлено на изучение градиентных методов и их применения в обучении линейных методов. Цель исследования - реализовать двухклассовую линейную модель классификации (логистическую регрессию) и научиться анализировать поведение градиентных методов во время обучения модели. Одними из результатов данного задания - опыт работы с различными техниками NLP для векторизации текстов.

2 Постановка задачи

2.1 Логистическая регрессия

2.1.1 Постановка задачи бинарной логистической регрессии

Задача классификации и принцип максимума правдоподобия
Пусть $X \times Y$ - в. п. с плотностью $p(x, y)$
Пусть X^l - простая выборка: $(x_i, y_i)_{i=1}^l \approx p(x, y)$
Задача: по выборке X^l оценить плотность $p(x, y)$
$p(x, y) = P(y|x, w)p(x)$ - параметризация плотность
$P(y|x, w)$ - модель условной вероятности класса с параметром w
$p(x)$ - неизвестное и непараметризуемое распределение на X
MLE-оценка для w (Maximum Likelihood Estimate):

$$\prod_{i=1}^l p(x_i, y_i) = \prod_{i=1}^l P(y_i|x_i, w)p(x_i) \to \max_w$$

Логарифм правдоподобия (log-likelihood, log-loss):

$$L(w) = \sum_{i=1}^l P(y_i|x_i, w) \to \max_w$$

Двухклассовая логистическая регрессия
Линейная модель классификации для двух классов $Y = \{-1, +1\}$:

$$a(x) = sign < w, x >, x, w \in \mathbb{R}^n$$

Отступ $M =< w, x > y$
Логарифмическая функция потерь: $L(M) = log(1 + \exp^{-M})$
Модель условной вероятности: $P(y|x, w) = \sigma(M) = \frac{1}{1+\exp^{-M}}$, где $\sigma(M)$ - сигмоиндная функция.
Максимизация правдоподобия (logistic loss) с регуляризацией:

$$Q(w) = \sum_{i=1}^{l} log(1 + \exp^{(-<w,x_i>y_i)}) + \frac{\tau}{2}||w||^2 \to min_w$$

2.1.2 Постановка задачи многоклассовой логистической регрессии

Линейный классификатор при произвольном числе классов $|Y|$:

$$a(x) = arg\max_{y \in Y} < w_y, x >, x, w_y \in \mathbb{R}^n$$

Вероятность того, что объект x относится к классу y:

$$P(y|x, w) = \frac{exp < w_y, x >}{\sum_{z \in Y} exp < w_z, x >} = SoftMax_{y \in Y} < w_y, x >$$

Максимизация правдоподобия (log-loss) с регуляризацией:

$$L(w) = \sum_{i=1}^{l} logP(y_i|x_i, w) - \frac{\tau}{2} \sum_{y \in Y} ||w_y||^2 \to \max_w$$

2.2 Градиентные методы

2.2.1 GD

Минимизация эмпирического риска:

$$Q(w) = \sum_{i=1}^{l} L(w, x_i) \to \min_w$$

Метод градиентного спуска:
$w^{(0)}$:= начальное приближение; $w^{(t+1)} := w^{(t)} - h * \nabla Q(w^{(t)})$
$\nabla Q(w) = (\frac{\partial Q(w)}{\partial w_j})_{j=0}^{n}$, где h - градиентный шаг, называемый также темпом обучения.

$$w^{(t+1)} := w^{(t)} - h \sum_{i=1}^{l} \nabla L(w^{(t)}, x_i)$$

2.2.2 SGD (Stochastic Gradient Deep)

$Q = \sum_{i=1}^{l} L(w, x_i) \to \min_w$ - минимизация эмпирического риска
Вход: выборка X^l, темп обучения h, темп забывания δ;
Выход: вектор весов w;

1. инициализировать веса w_j, $j = 0, ..., n$;
2. инициализировать оценку функционала:
   $Q :=$ среднее $L(w, x_i)$ по случайном подмножеству $\{x_i\}$;
3. повторять
4. | выбрать объект x_i из X^l случайным образом;
5. | вычислить потерю $\epsilon_i := L(w, x_i)$;
6. | сделать градиентный шаг $w := w - h\nabla L(w, x_i)$
7. | оценить функционал $Q := \delta\epsilon_i + (1 - \delta)Q$;
8. пока значение Q и/или веса w не сойдутся;

2.2.3 Реализация в практическом задании

В собственной реализации была реализована двухклассовая логистическая регрессия с l2-регуляризацией. Реализован GD и SGD mini-batch версия, в которой выбирается не один объект, а батч, задающийся гиперпараметром batch_size. Градиентный шаг был выбран, как $h_k = \frac{\alpha}{k^\beta}$, где α и β - гиперпараметры, k - номер итерации/эпохи.

2.3 Расчет градиентов функционала логистической регрессии

2.3.1 Вывод градиента функции потерь для задачи бинарной логистической регрессии

Дана функция потерь $Q(w) = \sum_{i=1}^{l} ln(1 + exp(- < w, x_i > y_i)) + \frac{\tau}{2}||w||^2$

$\nabla(\frac{\tau}{2}||w||^2) = \tau w$. Рассмотрим оставшуюся часть.

$\nabla \sum_{i=1}^{l} ln(1 + exp(- < w, x_i > y_i)) = \sum_{i=1}^{l} \nabla ln(1 + exp(- < w, x_i > y_i))$, далее рассмотрим $\nabla ln(1 + exp(- < w, x_i > y_i))$.

$d(ln(1+exp(- < w, x_i > y_i))) = \frac{d(1 + exp(- < w, x_i > y_i))}{1 + exp(- < w, x_i > y_i)} = \frac{exp(- < w, x_i > y_i) d(- < w, x_i > y_i)}{1 + exp(- < w, x_i > y_i)} = \frac{exp(- < w, x_i > y_i)(- < dw, x_i > y_i)}{1 + exp(- < w, x_i > y_i)} = (-y_i \frac{exp(- < w, x_i > y_i)}{1 + exp(- < w, x_i > y_i)} x_i, dw)$, следовательно $\nabla ln(1+exp(- < w, x_i > y_i)) = -y_i \frac{exp(- < w, x_i > y_i)}{1 + exp(- < w, x_i > y_i)} x_i = -y_i \frac{1}{1 + exp(< w, x_i > y_i)} x_i = -y_i \sigma(- < w, x_i > y_i) x_i$.

Собираем полностью формулу.

$\nabla Q(w) = - \sum_{i=1}^{l} y_i \sigma(- < w, x_i > y_i) x_i + \tau w$

2.3.2 Вывод градиента фукнции потерь для задачи многоклассовой логистической регрессии

$$p_k = P(y = k|x, W) = \frac{exp\,(< w_k, x > + w_{0k})}{\sum\limits_{j=1}^{K} exp\,(< w_j, x_i > + w_{0j})}, \text{ где}$$

$W = (w_1, \ldots, w_K)$ — Матрица весов, а w_k - веса k-ой линейной модели.

$$L(W, X, y) = - \sum_{i=1}^{N} log(p_{y_i})$$

$w_k = (w_{0k}|w_k)$ и к $x_i = 1|x_i$, $i = \overline{1, N}$, следовательно $exp\,(< w_k, x > + w_{0k}) = exp < w_k, x >$

Найдем производную p_k по вектору весов $w_l : l = k$

$$\nabla_{w_k} p_k = \frac{\nabla_{w_k} exp < w_k, x >}{\sum\limits_{j=1}^{K} exp < w_j, x >} - \frac{exp < w_k, x > \nabla_{w_k} exp < w_k, x >}{\left[\sum\limits_{j=1}^{K} exp < w_j, x >\right]^2} =$$

$$= p_k x - p_k^2 x = p_k(1 - p_k)x$$

и по вектору весов $w_l : l \neq k$

$$\nabla_{w_l} p_k = \frac{\nabla_{w_l} exp < w_k, x >}{\sum\limits_{j=1}^{K} exp < w_j, x >} - \frac{exp < w_k, x > \nabla_{w_l} exp < w_l, x >}{\left[\sum\limits_{j=1}^{K} exp < w_j, x >\right]^2} =$$

$$= 0 - p_k p_l x = -p_k p_l x$$

Тогда при $l = k$

$$\nabla_{w_l} log(p_k) = \frac{p_k(1 - p_k)x}{p_k} = (1 - p_k)x$$

при $l \neq k$

$$\nabla_{w_l} log(p_k) = \frac{-p_l p_k x}{p_k} = -p_l x$$

Получается

$$\nabla_{w_k} L(w, X, y) = -\sum_{i=1}^{N} log(p_{y_i}) = \sum_{i=1}^{N} (p_k - 1\,[y_i = k])x_i$$

2.3.3 Сведение многоклассовой логистической регрессии при количестве классов = 2 к бинарной логистической регресии

Так как в Y всего два класса y, z, то уравнение разделяющей плоскости будет только одно $< x, w_y >=< x, w_z >$ следовательно $< x, w_y - w_z >= 0$ - это уравнение разделяющей плоскости для бинарной классификации, если $w := w_y - w_z$. Далее $P(y|x, w) = \frac{exp < w_y, x >}{\sum_{z \in Y} exp < w_z, x >}$, так как классов всего два $P(y|x, w) = \frac{exp < w_y, x >}{exp < w_y, x > + exp < w_z, x >}$, следовательно $P(y|x, w) = \frac{1}{1 + exp < w_z - w_y, x >} = \sigma(< w, x >)$. Ч. т. д. - это задача бинарной логистической регрессии.

3 Эксперименты

3.1 Предварительная обработка данных

Целью данного эксперимента является предобработка текстов датасета с помощью регулярных выражений. В рамках эксперимента все текста были приведены к нижнему регистру, все символы, не являющиеся английскими буквами и цифрами были заменены на пробелы. Данный результат требуется для дальнейшей обработки с меньшей размерностью по частотности текстов из датасета.

3.2 Преобразование данных в матрицу частотности

Целью данного эксперимента является создания матрицы частотности слов из текстов в датасете (предобработанном), чтобы представить тексты, как вектора из числовых признаков. В рамках эксперимента был использован класс CountVectorizer из библиотеки scikit-learn для векторизации каждого документа в разреженную матрицу из библиотеки scipy. Был использован параметр min_df для уменьшения размерности пространства объектов с 89к до 11к (min_df=10), чтобы уменьшить затраты по времени и вероятность переобучения, так как объектов в выборке 51к (<89к). Данный результат требуется для обучения двухклассовой логистической регрессии с помощью градиентных методов.

3.3 Реализация методов GD и SGD и их сравнение аналитического градиента с численным

Целями данного эксперимента является написание собственной реализации метода градиентного спуска и стохастического градиентного спуска на батчах и сравнение собственной реализации аналитического подсчета градиента с численным методом. В рамках эксперимента с использованием библиотек scipy и numpy был написан класс SGDClassifier, GDClassifier и оракул BinaryLogistic для подсчета функционала и его градиента. Также в результате сравнения аналитического подсчета градиента с численным было получено, что собственная реализация получает градиент с точностью 0.00001 от численного подсчета. Данный результат будет применим в дальнейшем для экспериментов с собственными реализациями GD и SGD.

3.4 Исследование поведения метода GD в зависимости от начальных данных и гиперпараметров

Целями данного эксперимента является изучение поведения метода GD в зависимости от начальных данных и гиперпараметров step_beta и step_alpha и определение из выбранных наборов параметров, на котором метод GD сходится . В рамках эксперимента было проведено сравнение поведения на одном переменном параметре и остальном зафиксированном наборе. Выбор начальных параметров, кроме коэффициента l2-регуляризации, был случаен. Начальным является следующий набор параметров $step\_alpha = 1, step\_beta = 0, tolerance = 0.00001, l2\_coef = 0.00001$. Выбор количества итераций равных 50 был обусловлен тем, что в данном эксперименте важно рассмотреть поведения метода и его сходимость. Модель во всех экспериментах была обучена на обучающей выборке, точность модели была подсчитана на каждой итерации также на обучающей выборке. Начальное приближения w было выбрано, как нулевой вектор размерности количества признаков объекта обучающей выборки.

3.4.1 Исследование поведения GD в зависимости от step_alpha

В данном подпункте эксперимента были рассмотрены зависимости функционала и точности метода GD в зависимости от номера итерации/времени и гиперпараметра step_alpha. Выборка гиперпараметров была создана на предположении, что параметр является степенью 10. Экспериментально было проверено, что параметры 10, 100 и т. д. не обеспечивают сходимость GD при заданных начальных параметров. Результаты сравнения на других значения гиперпараметра приведены на Рис. 1.

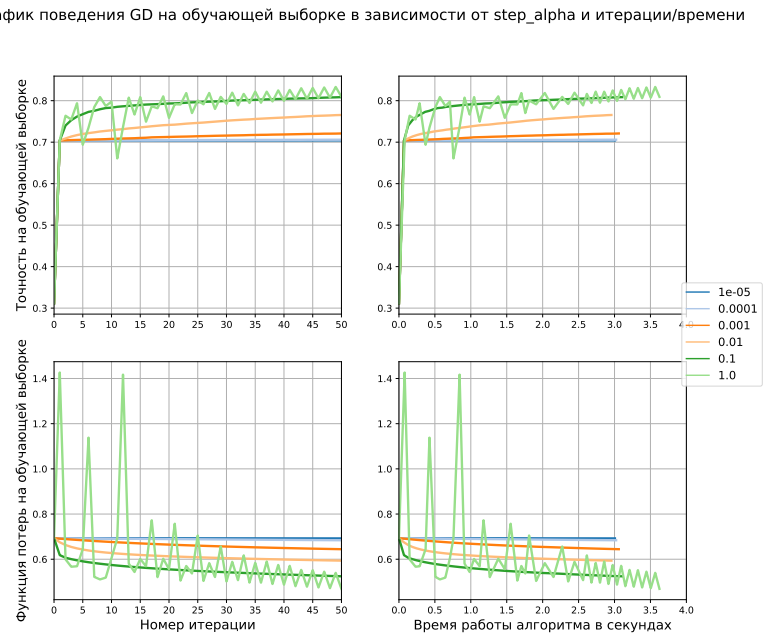

Рис. 1: График поведения GD на обучающей выборке в зависимости от step_alpha и итерации/времени

На графике можно заметить, что при увеличении параметра step_alpha скорость роста точности становится больше. Результатом данного эксперимента является выбор $0.01, 0.1, 1.0$ - величины параметров, на которых будет проведен эксперимент №6 для поиска оптимального набора параметров для получения высокой точности на обучающей выборке.

3.4.2 Исследование поведения GD в зависимости от step_beta

В данном подпункте эксперимента был рассмотрен аналогичный прошлому подпункту эксперимент, но в зависимости от step_beta. Аналогично, выборка была создана на предположении, что параметр

является степенью 10. Было экспериментально проверено, что гиперпараметры с степенью 1 и более являются не оптимальными, так как градиентный шаг в методе GD очень быстро стремится к нулю. Результаты сравнения на других значения гиперпараметра приведены на Рис. 2.

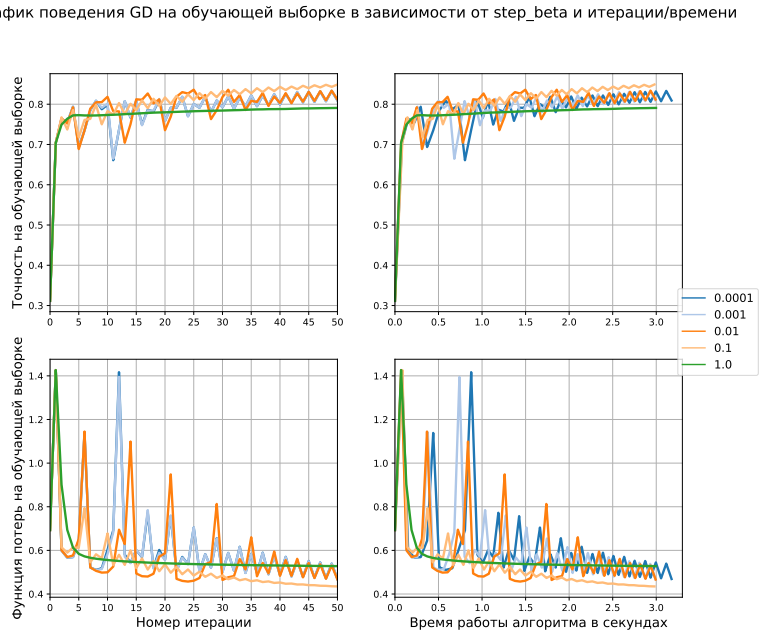

Рис. 2: График поведения GD на обучающей выборке в зависимости от step_beta и итерации/времени

На графике можно заметить, что при уменьшении параметра step_beta скорость уменьшения градиентного шага становится меньше. Результатом данного эксперимента является выбор $0.1, 0.001$ - величины параметров, на которых будет проведен эксперимент №6.

3.4.3 Исследование поведения GD в зависимости от начального приближения w

В данном подпункте эксперимента был рассмотрен эксперимент с изучением поведения метода GD при различных приближениях. Были рассмотрены следующие приближения:

1. Случайный вектор из нормального распределения с мат. ожиданием 0 и дисперсией 1 или 2.
2. Случайный вектор из равномерного распределения из отрезка $[-a, a], a \in \{1, 2, 3\}$.
3. Нулевой вектор.
4. Вектор, обученный на меньшей выборке размером 1k/2k/4k/6k.
5. Вектор, считающийся оптимальной оценкой, полученный следующим образом. $w_i = \frac{< y, f_i >}{< f_i, f_i >}$, где$f_i$-вектор-столбец i-го признака

Данные приближения часто используются, как начальные. График поведения GD при заданных приближениях представлен на Рис. 3.

В рамках эксперимента получены следующий вывод: начальное приближение влияет только на скорость сходимости и начальную точку функционала, с которой начнет обучаться метод. На графике заметно, что некоторые приближения сходятся к одной траектории, но точка начала функционала у них разные. Результатом данного эксперимента является выбор вектора, являющегося оптимальной оценкой, для экспериментов далее.

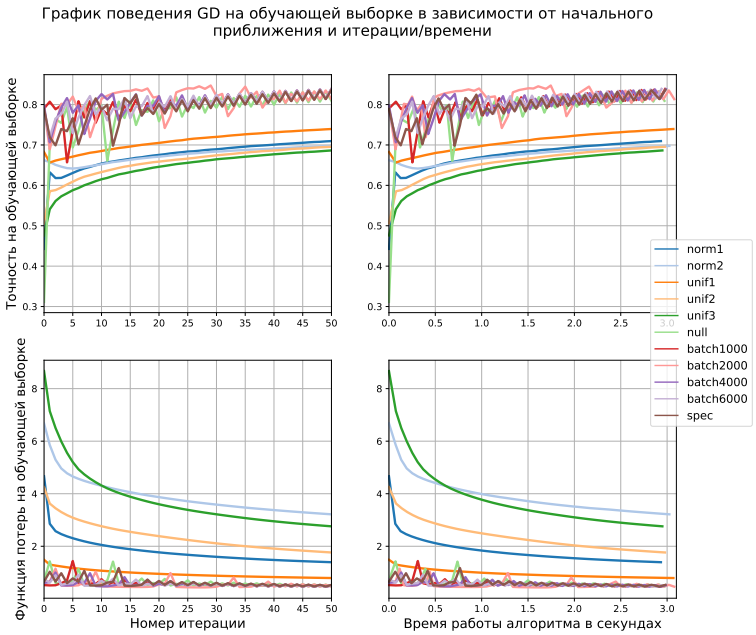

Рис. 3: График поведения GD на обучающей выборке в зависимости от начального приближения и итерации/времени

3.4.4 Общие итоги эксперимента

В рамках данного эксперимента получена выборка параметров step_alpha, step_beta и зафиксировано начальное приближение w_0. Проведен анализ зависимостей функционала и точности от параметров и времени/номера итерации. Заметно, что графики по времени работы имеют часто одинаковый характер зависимостей с графиками по итерациям. Результаты данного эксперимента будут расширены в эксперименте №6, когда будут исследованы сочетания параметров, получившихся оптимальными в данном.

3.5 Исследование поведения метода SGD в зависимости от начальных данных и гиперпараметров

Целями данного эксперимента является изучение поведения метода SGD в зависимости от начальных данных и гиперпараметров step_beta, step_alpha и batch_size и определение из выбранных наборов параметров, на котором метод SGD сходится . В рамках эксперимента было проведено сравнение поведения на одном переменном параметре и остальном зафиксированном наборе. Выбор начальных параметров, кроме коэффициента l2-регуляризации, был случаен. Начальным является следующий набор параметров $step\_alpha = 1, step\_beta = 0, tolerance = 1e^{-5}, l2\_coef = 0.00001, random_seed = 153, batch_size = 1000$. Выбор количества эпох равных 50 был обусловлен тем, что в данном эксперименте важно рассмотреть поведения метода и его сходимость. Модель во всех экспериментах была обучена на обучающей выборке, точность модели была подсчитана на каждой итерации также на обучающей выборке. Начальное приближения w было выбрано, как нулевой вектор размерности количества признаков объекта обучающей выборки. Параметр log_freq равен 0.3 на всех экспериментах.

3.5.1 Исследование поведения SGD в зависимости от step_alpha

В данном подпункте эксперимента изучено поведение метода SGD в зависимости от step_alpha. Аналогично прошлом эксперименту были рассмотрены степени 10 и получены, экспериментально, результаты, что при параметре >1 метод не сходится. Результаты по других параметрам представлены

на Рис. 4.

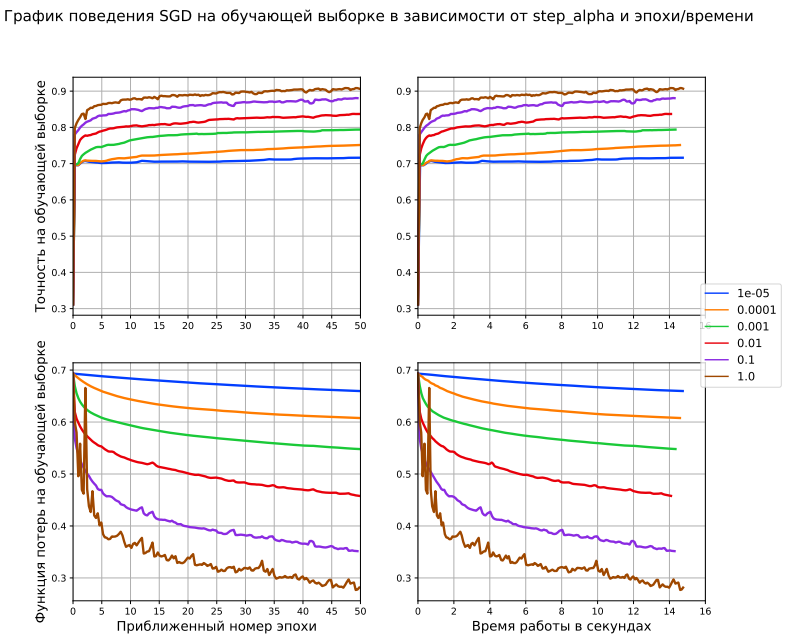

Рис. 4: График поведения SGD на обучающей выборке в зависимости от step_alpha и эпохи/времени

На графике заметно, что с увеличением параметра step_alpha увеличивается точность и скорость сходимости и уменьшения функционала. Результатом данного эксперимента является выбор $0.01, 0.1, 1.0$, как оптимальных параметров для эксперимента №6, в котором будет произведен выбор оптимального сочетания параметров для SGD с максимальной точностью.

3.5.2 Исследование поведения SGD в зависимости от step_beta

В данном подпункте изучено поведение метода SGD в зависимости от step_beta. Аналогично, прошлом подпункту были рассмотрены степени 10 и получены экспериментально, что параметры > 1 быстро уменьшают градиентный шаг. Результаты по другим параметрам представлены на Рис. 5.

На графике заметно, что с уменьшением параметра step_beta точность на обучающей выборке увеличивается, и отрицательные степени 10 имеют близкие характеры зависимости. Результатом данного эксперимента является выбор $0.001, 0.01$, как оптимальных параметров для эксперимента №6.

3.5.3 Исследования поведения SGD в зависимости от batch_size

В данном подпункте изучено поведение метода SGD в зависимости от batch_size. Были рассмотрены параметры, принимающие значения степени 10 и степени 10, умноженной на 3. Графики поведения по параметрам представлены на Рис. 6.

На графике заметно, что с уменьшением размера батча, время потраченное на обучение кратно увеличивается и батчи большие, чем 5 часть от всей обучающей выборки, имеют меньшую точность. На графике заметны две линии, отвечающие за батчи 1000 и 3000, которые сходятся за меньше 50 эпох. В результате данного эксперимента был выбран batch_size размером 1000.

3.5.4 Исследование поведения SGD в зависимости от начального приближения w

В данном подпункте эксперимента изучено поведение метода SGD в зависимости от начальных приближений. Начальные приближения рассмотрены те же, что и в эксперименте 4.3. Графики поведение от начальных приближений представлены на Рис. 7.

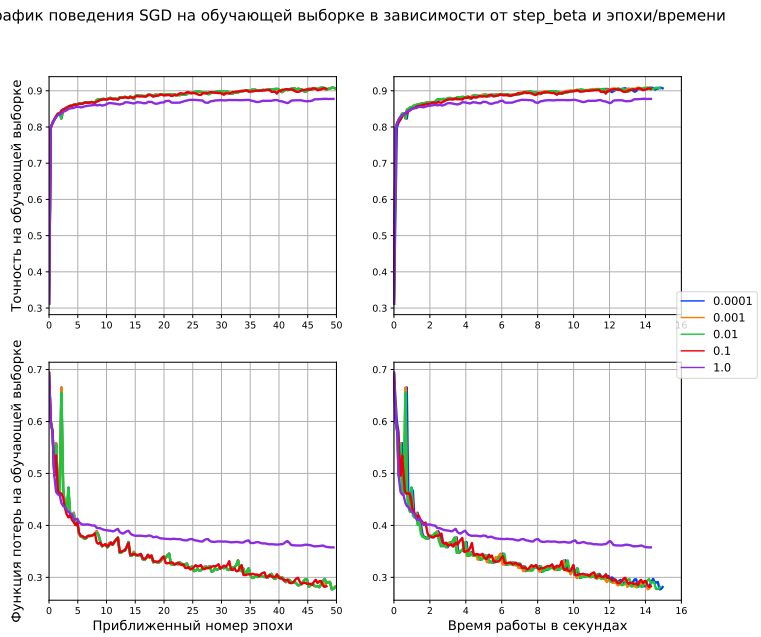

Рис. 5: График поведения SGD на обучающей выборке в зависимости от step_beta и эпохи/времени

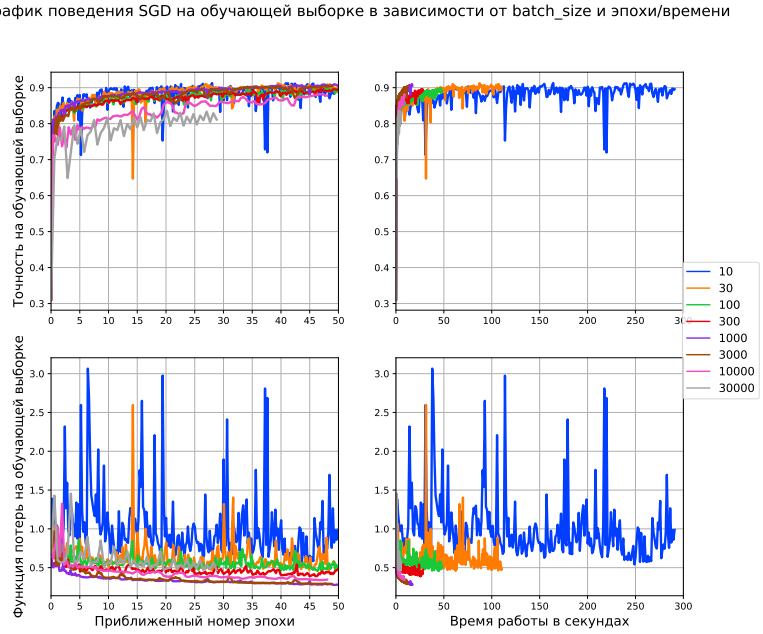

Рис. 6: График поведения SGD на обучающей выборке в зависимости от batch_size и эпохи/времени

Выводы в данном подпункте аналогичны выводам в эксперименте №4.3. Результатом данного эксперимента является выбор начального приближения spec - оптимальной оценки для эксперимента №6.

3.5.5 Общие итог эксперимента

В рамках данного эксперимента получена выборка параметров step_alpha, step_beta, batch_size и зафиксировано начальное приближение w_0. Проведен анализ зависимостей функционала и точности от параметров и времени/номера итерации. Заметно, что графики по времени работы имеют часто оди-

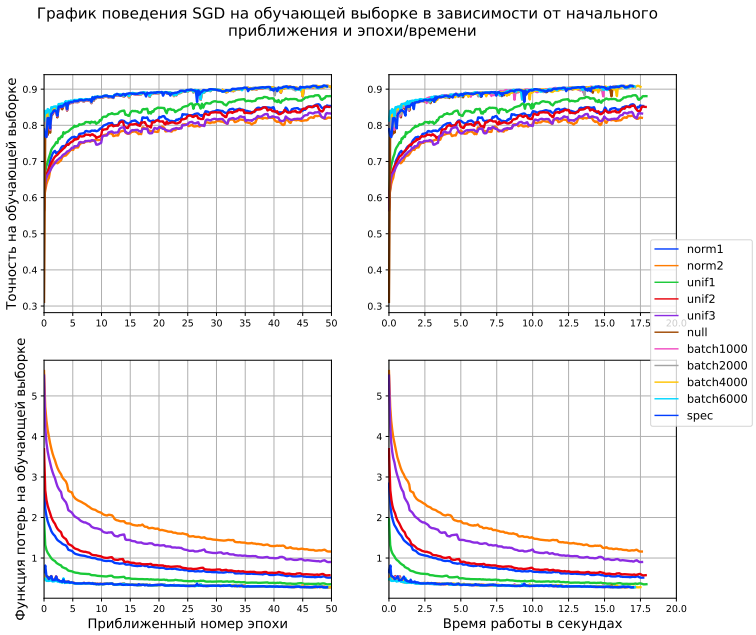

Рис. 7: График поведения SGD на обучающей выборке в зависимости от начального приближения и эпохи/времени

наковый характер зависимостей с графиками зависимости от эпох. Результаты данного эксперимента будут расширены в эксперименте №6, когда будут исследованы сочетания параметров, получившихся оптимальными в данном.

3.6 Сравнение поведения GD и SGD

Цель данного эксперимента сравнить GD и SGD. В рамках эксперимента были найдены параметры GD и SGD, которые реализуют максимальную и стабильную точность спустя 1000 итераций/эпох. Параметры для сравнения были найдены в эксперименте №4 и №5, соответственно, для GD и SGD. После поиска 'оптимальных' наборов параметров было проведено сравнение GD и SGD, результатом которого является кандидат на лучший алгоритм в эксперименте №9 и кандидат для проверки следующих экспериментов.

3.6.1 Поиск оптимальных параметров и начального приближения для GD

Цель данного подпункта - найти 'оптимальный' набор параметров для GD. Следующие параметры являются фиксированными в эксперименте $tolerance = 0.00001, max\_iter = 1000, l2\_coef = 0.00001$. Начальное приближение выбрано, как w_spec. В эксперименте рассмотрены сочетания параметров step_alpha и step_beta из оптимальных выборов в эксперименте №4. Результат обучения на обучающей выборке представлен на Рис. 8.

Результатом данного эксперимента является комбинация параметров step_alpha=1.0 и step_beta=0.1, которые будут использованы в следующем сравнении.

3.6.2 Поиск оптимальных параметров и начального приближение для SGD

Цель данного подпункта - найти 'оптимальный' набор параметров для GD. Следующие параметры являются фиксированными в эксперименте $tolerance = 0.00001, max\_iter = 50, random\_seed = 153, l2\_coef = 0.00001, batch\_size = 1000$. Начальное приближение выбрано, как w_spec. В экспери-

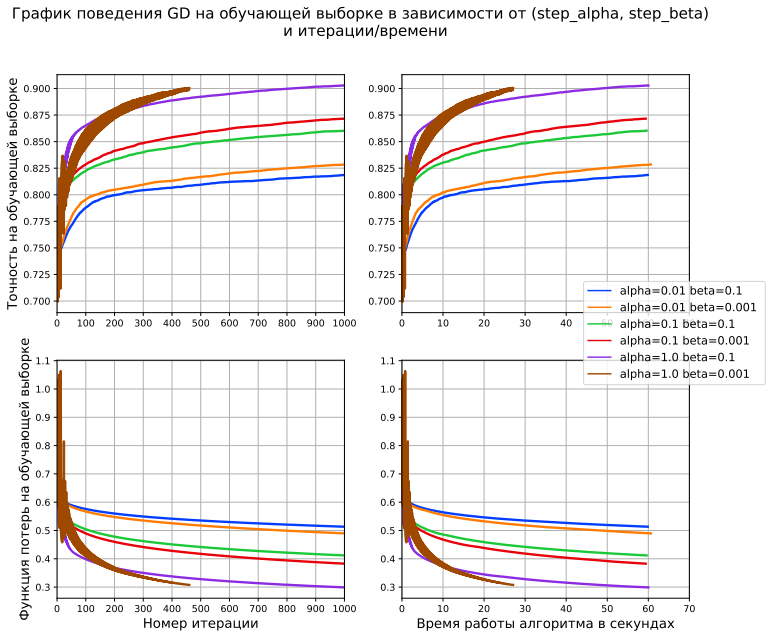

Рис. 8: График поведения GD на обучающей выборке в зависимости от (step_alpha, step_beta) и итерации/времени

менте рассмотрены сочетания step_alpha и step_beta из оптимальных выборов в эксперименте №5. Результат обучения на обучающей выборке представлен на Рис. 9.

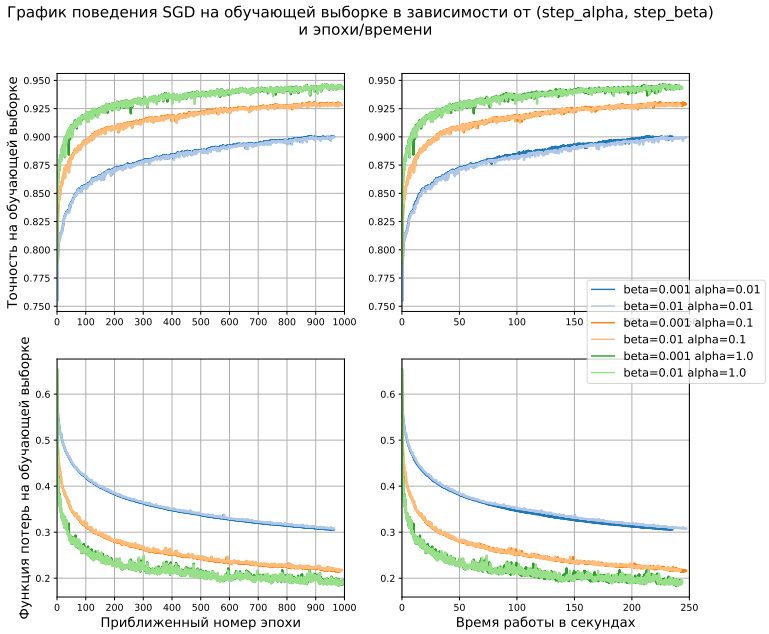

Рис. 9: График поведения SGD на обучающей выборке в зависимости от (step_alpha, step_beta) и эпохи/времени

Результатом данного эксперимента является выбор параметров step_alpha=1.0 и step_beta=0.001.

3.6.3 Сравнение 'оптимальных' GD и SGD

Целью данного эксперимента - определение кандидата, которые дает большую точность на отложенной выборке. Для создания отложенной выборки была использована перетасованная обучающая выборка, разделенная на 4:1, соответственно, на обучающую и отложенную.

Два метода не сравнимы в понятиях эпох и итераций, поэтому основное сравнение будет рассмотрено на времени работы при обучении. Результаты сравнения оптимальных методов на обучающей выборке представлены на Рис. 10.

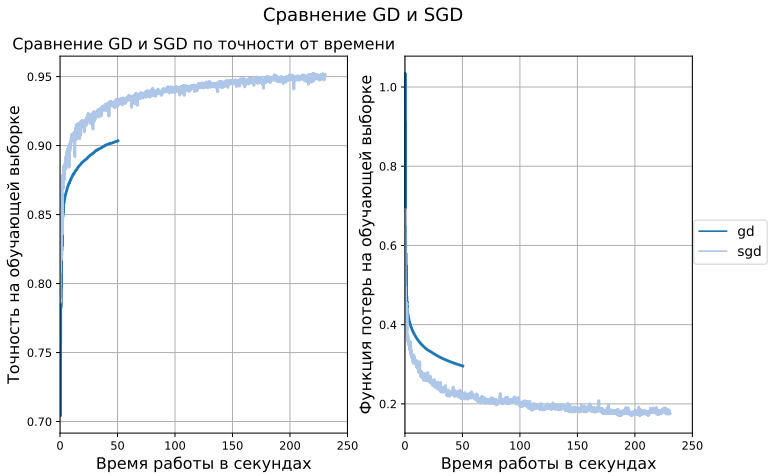

Рис. 10: Сравнение GD и SGD

Метод GD быстрее сходится, но не качественнее, в отличие от метода SGD. Результаты сравнения на отложенной выборке представлены на Рис. 11.

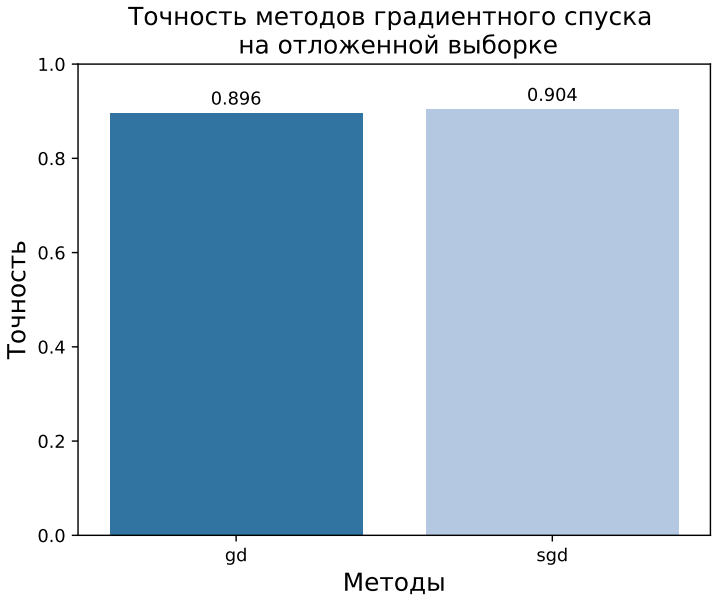

Рис. 11: Сравнение GD и SGD по точности на отложенной выборке

Данный эксперимент предоставляет важный результат для дальнейших экспериментов - SGD дает лучшее качество на отложенной выборке, чем GD.

3.7 Исследование влияния лемматизации текста на точность, время работы и размерность пространства

Целью данного эксперимента является анализ влияния лемматизации слов в тексте на точность, время работы и размерность признакового пространства. В рамках эксперимента используется библиотека nltk для работы с словами: удаление стоп-слов и их лемматизация.

Результаты сравнения признакового пространства, полученного на обучающей выборке, представлены на Рис. 12.

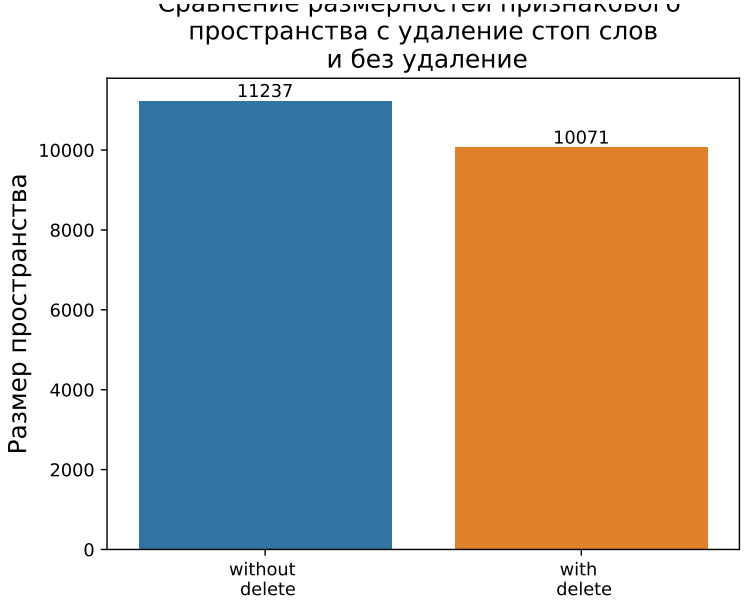

Рис. 12: Размерность признакового пространства до лемматизации и после

Уменьшение пространства происходит, потому что удаляются стоп-слова и некоторые слова становятся равными после лемматизации. Результат сравнения времени работы на обучающей выборке и точность на отложенной представлены на Рис. 13.

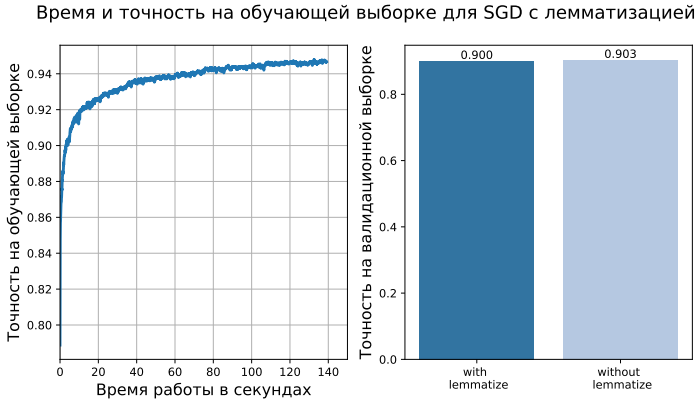

Рис. 13: График зависимость точности от времени на обуч. выборке и график точности на отложенной выборке

Результат эксперимента показывает меньшую точность на отложенной выборке, но это явление не является следствием таких же зависимостей на тестовой. Следовательно, лемматизация остается, потому что она меньше по времени учится и обладает меньшим признаковым пространством.

3.8 Исследование качества, время работы алгоритма и размер признакового пространства в зависимости от представления и параметров min_df и max_df

Целями данного эксперимента является подбор параметров min_df и max_df, которые обеспечивают точность выше на обучающей выборке, и выбор качественного представления между BagOfWords и TF-IDF. В рамках эксперимента происходит перебор параметров min_df и max_df и сравнения каждого сочетания по времени работы, качеству и размеру признакового пространства, аналогично, и для представлений.

3.8.1 Исследование в зависимости от представления

В рамках данного подпункта исследуется два представления: TF-IDF и BOW и их сравнительный анализ по времени работы, качеству на отложенной выборке и размере полученного векторного пространства (на отложенной выборке).

Точность на отложенной выборке представлена на Рис. 14.

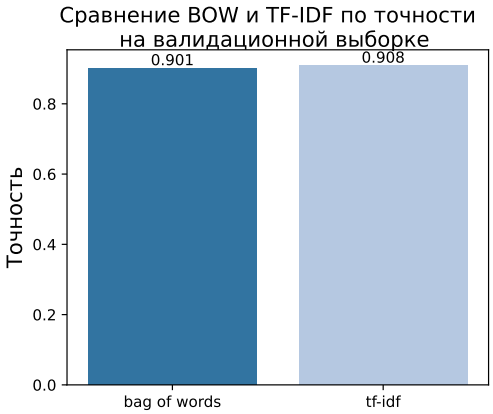

Рис. 14: Сравнение BOW и TF-IDF на отложенной выборке по точности

Заметно, что TF-IDF качественнее работает на отложенной выборке. На следующем Рис. 15 представлены результаты сравнения времени работы и качеству на обучающей выборке.

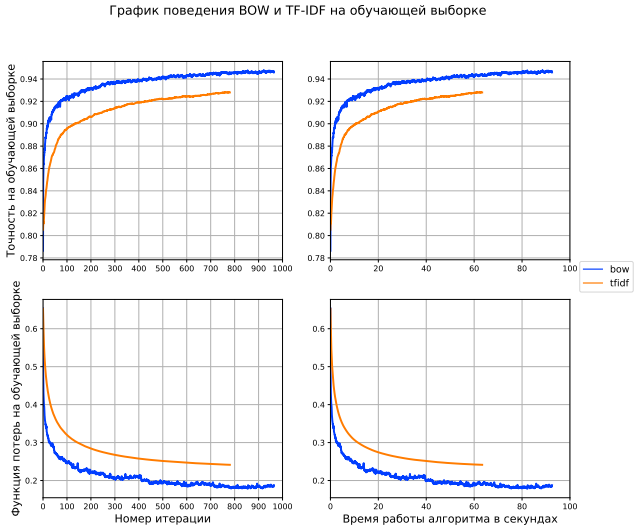

Рис. 15: Сравнение поведения BOW и TF-IDF на обучающей выборке

Заметно на графиках, что TF-IDF также показывает качественный результат. Данный результат будет использован в следующем эксперименте №9.

Размер признаков объектов не меняется в независимости от представлений, следовательно не требует отдельного рассмотрения.

3.8.2 Исследование в зависимости от параметров min_df и max_df

В данном эксперименте рассмотрены различные комбинации min_df и max_df. Экспериментально проверено, что max_df должен иметь значения заметно ниже 0.3 относительно размера выборки. min_df выбирается относительно представления, что нам нужно убрать самые редкие слова. Результаты данного эксперимента относительно размерности признакового пространства представлены на Рис. 16.

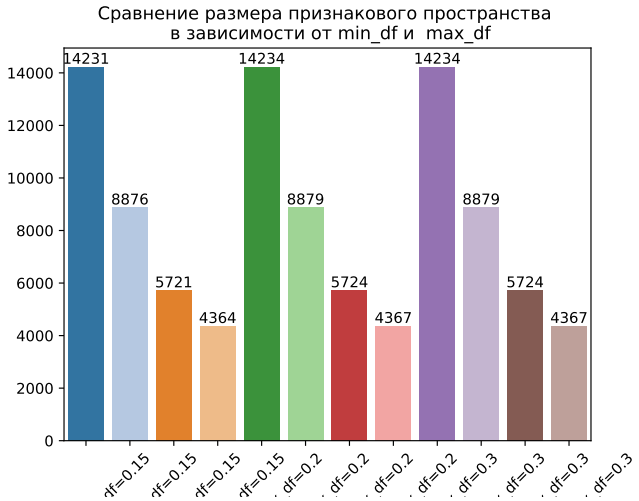

Рис. 16: Сравнение количества признаков для разных min_df и max_df на обучающей выборке

Заметно, что от изменения max_df размер пространства почти не меняется. В min_df наблюдается обратная картина. Далее рассмотрены результаты поведения SGD на обучающей выборке на Рис. 17 и на Рис. 18 представлена точность на отложенной выборке.

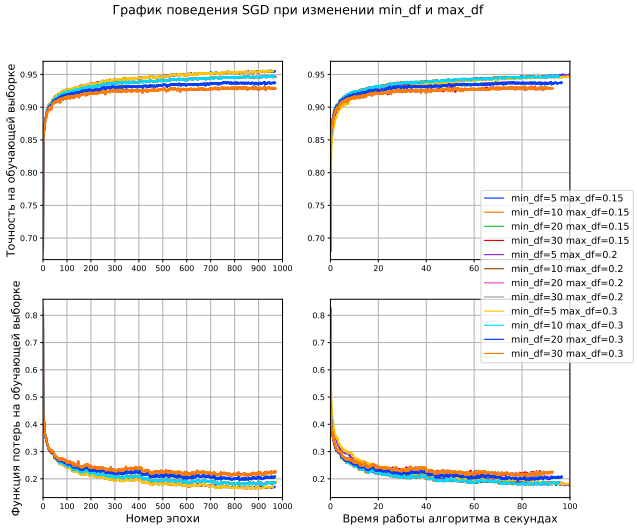

Рис. 17: График поведения SGD на обучающей выборке от эпохи/времени и min_df и max_df

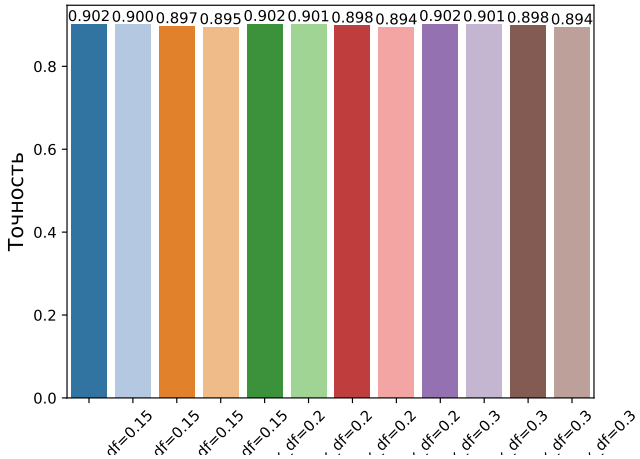

Рис. 18: Сравнение количества признаков для разных min_df и max_df на обучающей выборке

Заметно, что с увеличением min_df время обучения и точность уменьшается. Относительно max_df, вероятно, качество будет также снижаться по мере уменьшения параметра. Результатом эксперимента является набор параметров $min\_df = 5, max\_df = 0.2$, который реализует хорошую точность на отложенной выборке.

3.9 Выбор лучшего алгоритма и анализ объектов, на которых он допускает ошибку

В данном эксперименте рассмотрены лучшие параметры, которые были получены в прошлых результатах, на тестовой выборке с подсчетом точности на ней. Также проведен анализ объектов, для которых модель выдает ошибочные предсказания.

В рамках эксперимента рассмотрены результаты двух представлений - BOW и TF-IDF, чтобы решить какое представление дает лучший результат на тестовой выборке. Параметры для векторизатора равны следующим значениям $min_df = 5, max_df = 0.2$. Данный эксперимент проводится на 10к эпох, чтобы сойтись к еще более качественном результату, чем на 1000 эпох.

Результаты сравнения по точности на тестовой выборке представлены на Рис. 19.

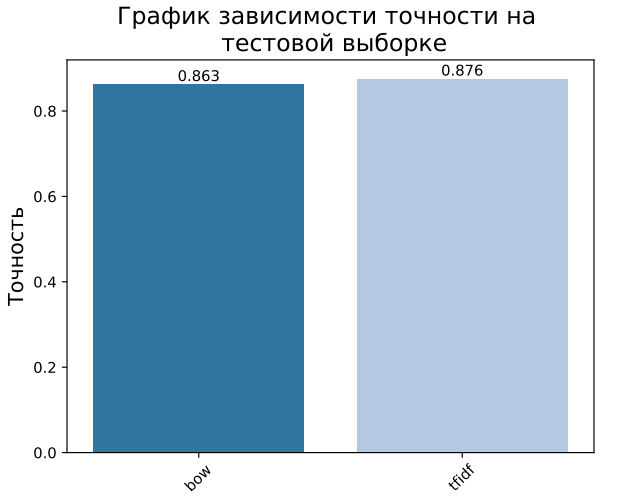

Рис. 19: Сравнение точности BOW и TF-IDF на тестовой выборке

Заметно, что TF-IDF качественнее BOW на тестовой выборке. Данный результат будет использован

в следующих экспериментах.

Были рассмотрены объекты, на которых модель совершает ошибки.

0.029 - ошибка FN **0.095** - ошибка FP

Следовательно, ошибка в определении токсичного комментария возникает в 3 раза чаще, чем ошибка определения токсичного комментария в не токсичном.

В объектах, которая модель не определяет, как негативные есть шумовые слова, полученные из токсичных слов, но с заменой нескольких букв, или добавлением. Встречается неправильная грамматика. Для того, чтобы решить эти проблемы, можно научиться подмешивать такие слова во время обучения.

3.10 Исследование влияния качества и времени работы алгоритма от размера максимальных n-gramm

Цель данного эксперимента - изучить влияние n-грам на точность и время работы метода. В рамках эксперимента будет меняться только максимальный размер n-грамы (n_max). Была установлена граница, ограничивая словарь триграмами и биграмами, в дополнение униграмам.

Увеличение до n-грам размером больше 3 только усложнить модель и время вычисления, но никак не увеличит качество, потому что зачастую в жизни не используют словосочетания из четырех слов, тем более в токсичных комментариях.

Обучено было проведено на 10к эпох.

Результат времени работы во время обучения представлено на Рис. 20.

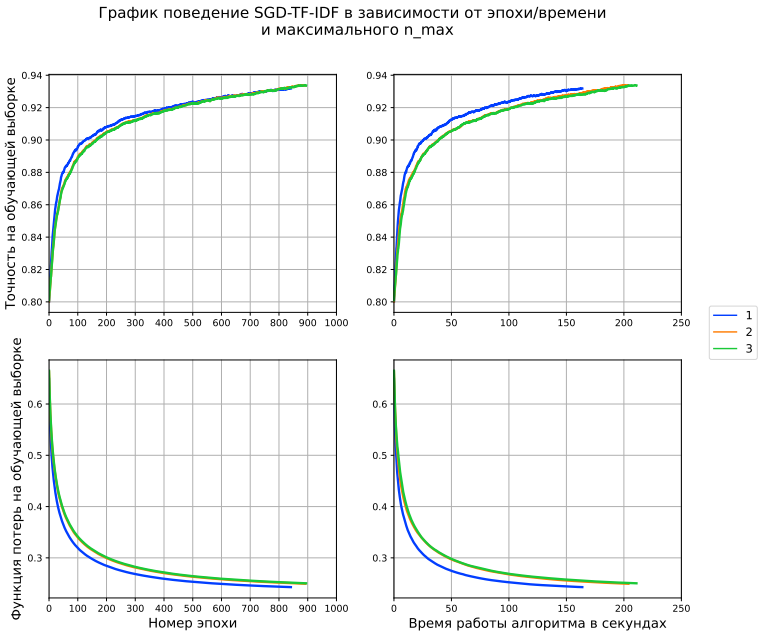

Рис. 20: График поведения SGD-TF-IDF в зависимости от n_max и эпохи/времени

Заметно, что n_max=2 и n_max=3 почти не отличимы и обычный алгоритм быстрее сходится. Для более информативного сравнения была рассмотрена точность на тестовой выборке, представленная на Рис. 21.

Данный результат можно объяснить тем, что в классификации текстов на токсичность роль n-грамм почти отсутствует, так как зачастую токсичность выражена в одном слове, но не в словосочетании. К примеру, n-грамы будут полезны для текстов подверженной наличию словосочетаний, как научные текста.

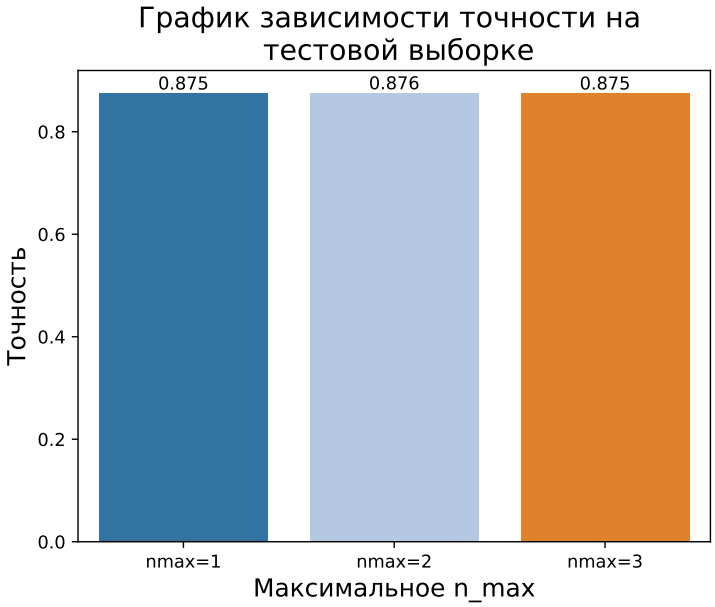

Рис. 21: Точность на тестовой выборке в зависимости от n_max

4 Заключение

Результатом данной работы являются навыки работы с градиентными методами и NLP, навыки подбора гиперпараметров. Были изучены методы векторизации текстов и способы улучшения их качества.

Есть несколько проблем у данного подхода, реализованного через 'мешок слов', с помощью него не передается контекст предложения, а в задаче классификации на токсичность порядок слов может многое менять. Был разобран эксперимент с n-грамами, направленный на решения данной проблемы, но ситуацию они не изменили.

Так же важно учитывать сходимость алгоритма SGD при выборе гиперпараметров.

5 Библиография

1. Документация библиотеки scipy https://docs.scipy.org/doc/scipy/
2. Документация библиотеки nltk https://www.nltk.org/
3. Лекции по ММО К. В. Воронцова по темам "Вероятностное порождение данных"и "Линейная классификация".
4. Семинарский конспект по логистической регрессии для классификации текстов.