Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

Отчёт по лабораторной работе №2
Продвинутые методы безусловной оптимизации.
По дисциплине: Методы оптимизации

Выполнил:
Зозуля А.В.
Проверила:
Головкина А.Г.

Москва 2024 г.

Формулы для градиента и гессиана функции логистической регрессии

1. Зависимость числа итераций метода сопряженных градиентов от числа обусловленности и размерности пространства

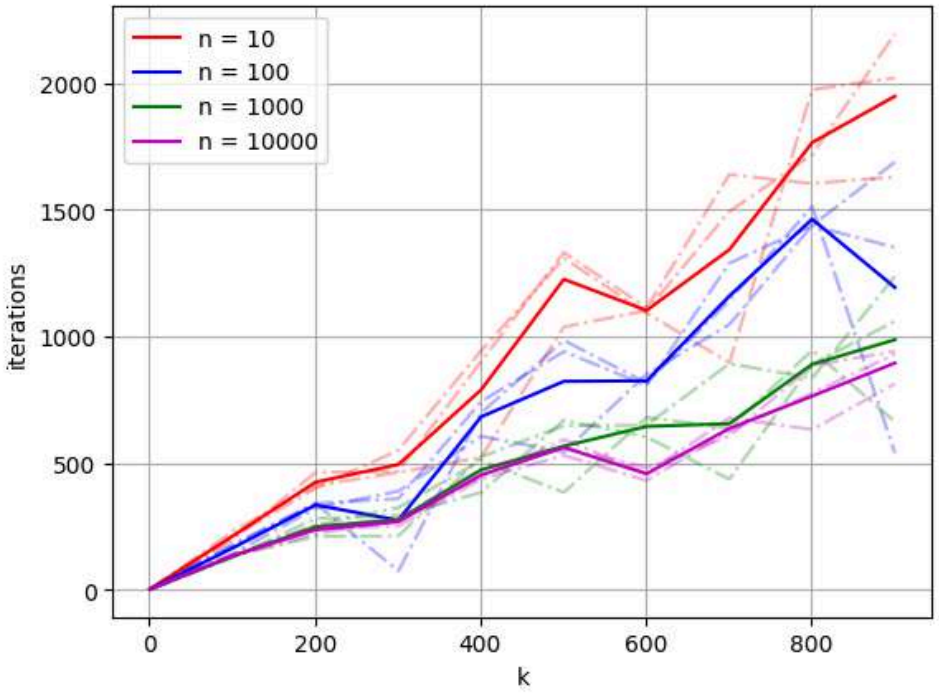

Рис. 1. Число итераций, необходимое градиентному спуску для сходимости, от числа обусловленности и размерности пространства n. Жирными линиями среднее по замерам для данной размерности

Почти линейная зависимость числа итераций от обусловленности функции, и слабое влияние размерности задачи на количество итераций, следовательно желательно преобразовать матрицу к меньшей обусловленности.

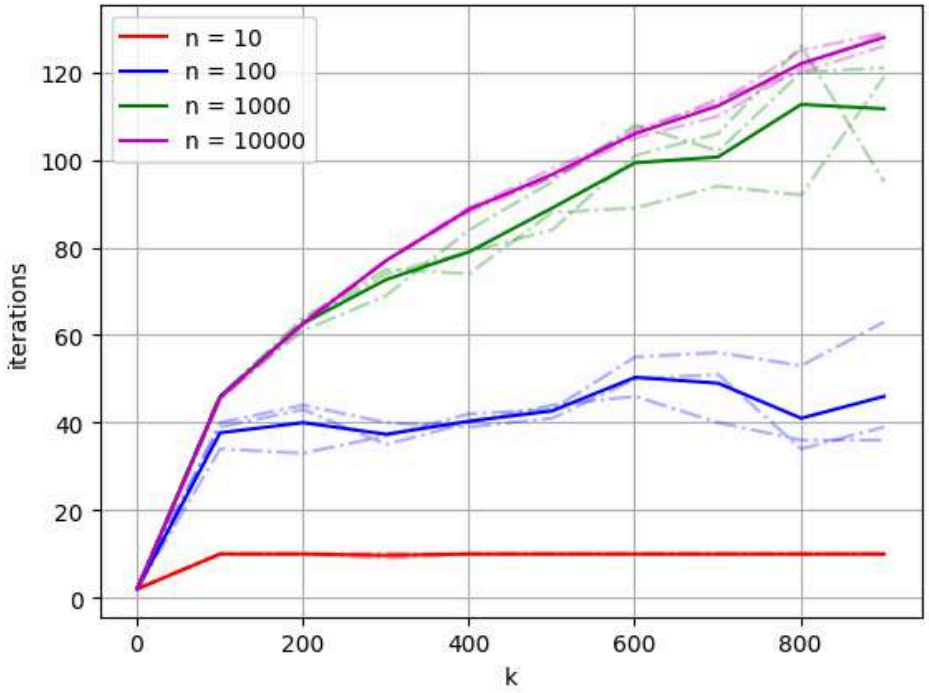

Рис. 2. Число итераций, необходимое сопряженным градиентам для сходимости, от числа обусловленности и размерности пространства n. Жирными линиями среднее по замерам для данной размерности

В методе сопряженных градиентов зависимость итераций от обусловленности похожа на зависимость функции корня. Количество итераций, как и следует из теории, не превосходит размерности пространства функции.

2. Выбор размера истории в методе L-BFGS

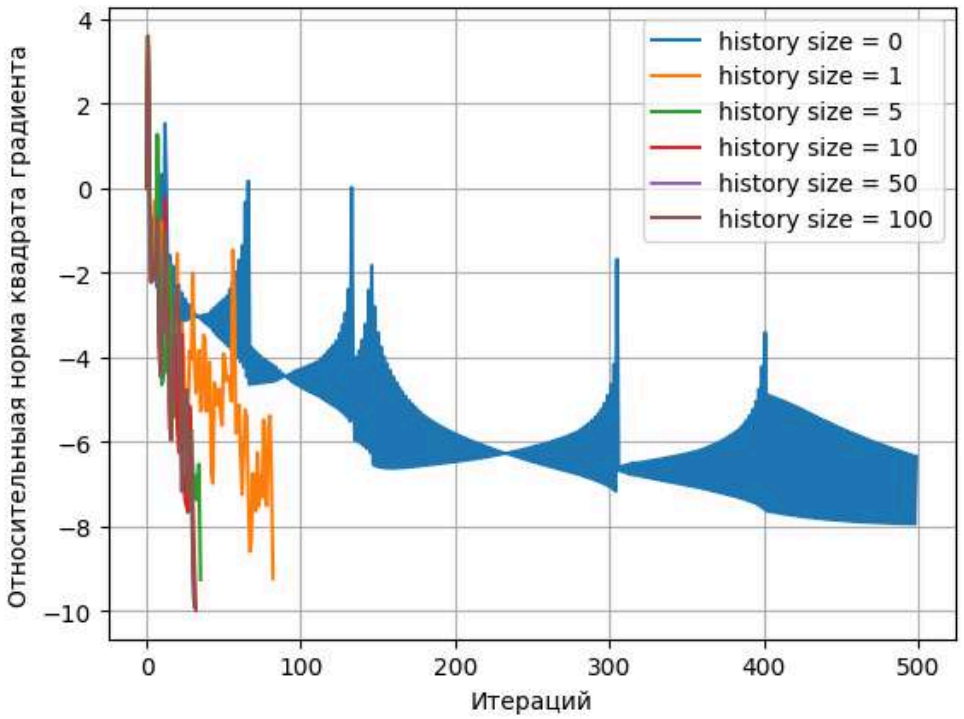

Рис. 3. Выбор размера истории в методе L-BFGS, датасет gisette_scale. Количество итерации до критерия останова

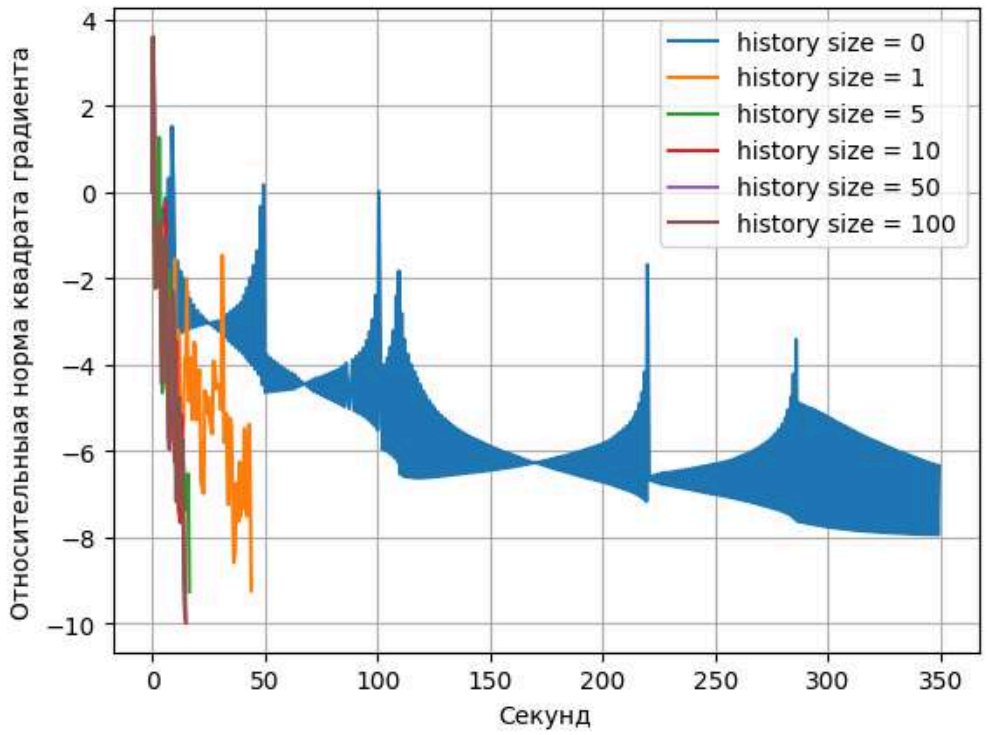

Рис. 4. Выбор размера истории в методе L-BFGS, датасет gisette_scale. Потраченное время до критерия останова

Чем больше история тем быстрее сходится метод по времени и по итерациям, для истории равной 0 метода начинает работать аналогично градиентному спуску. Но стоит учитывать что его преимущество в использовании векторов в количестве *m* вместо матрицы размерностью *nxn,* поэтому слишком большая история тоже не нужна, исходя из графиков оптимальное значение истории – 10, большее количество приводит к незначительному улучшению, и меньшему преимуществу по памяти по сравнению с BFGS.

3. Сравнение методов на реальной задаче логистической регрессии

Датасет w8a

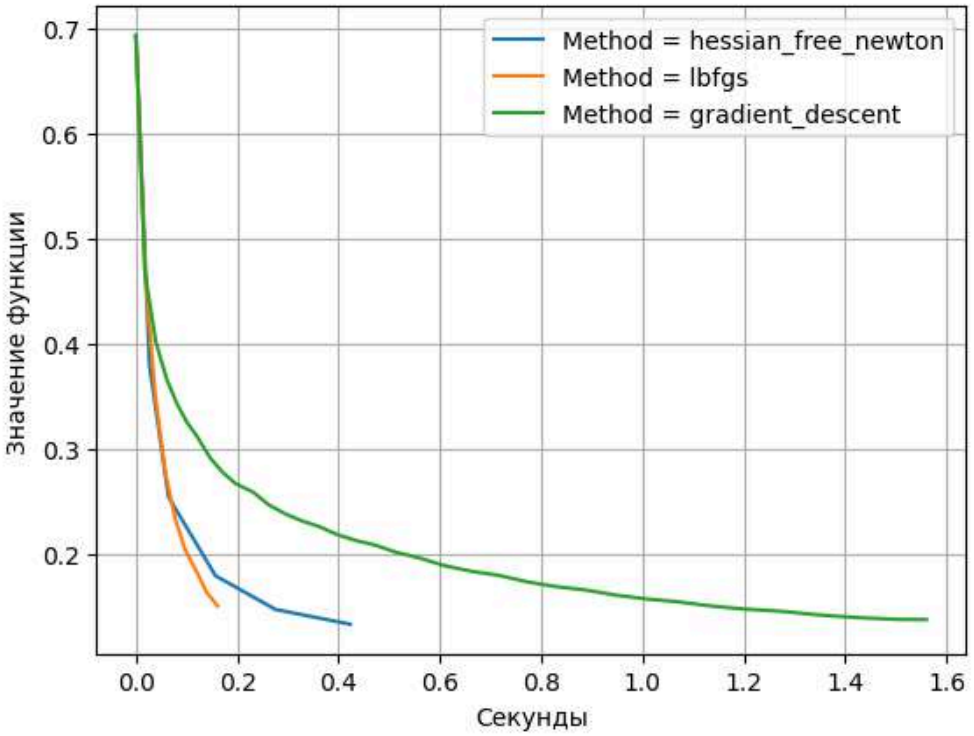

Рис. 5. Сравнение методов. Датасет w8a. Функция от времени.

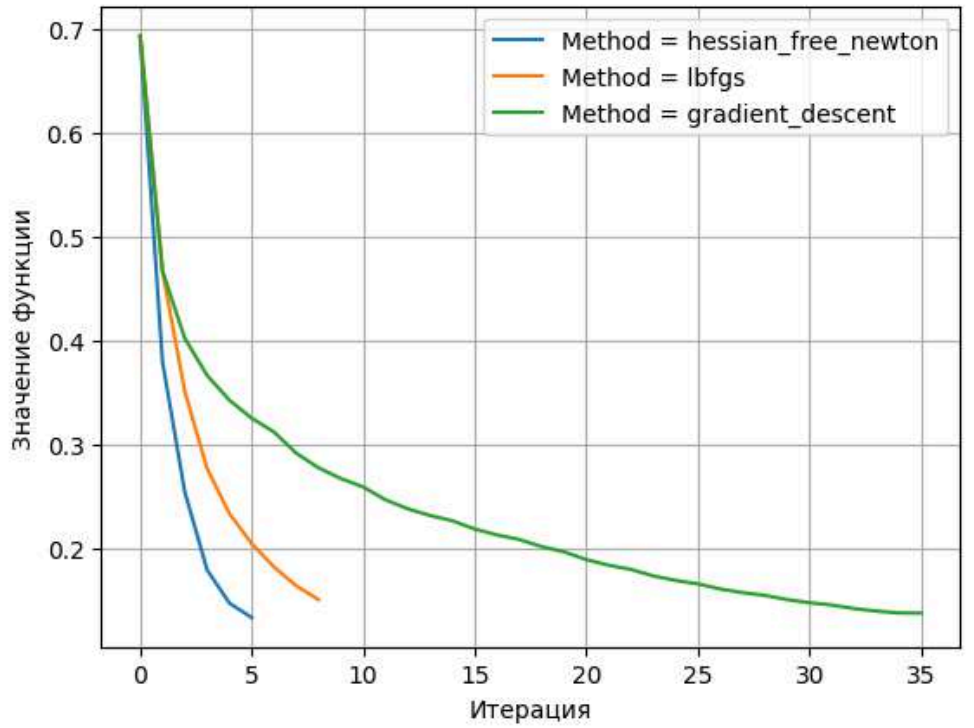

Рис. 6. Сравнение методов. Датасет w8a. Функция от итераций

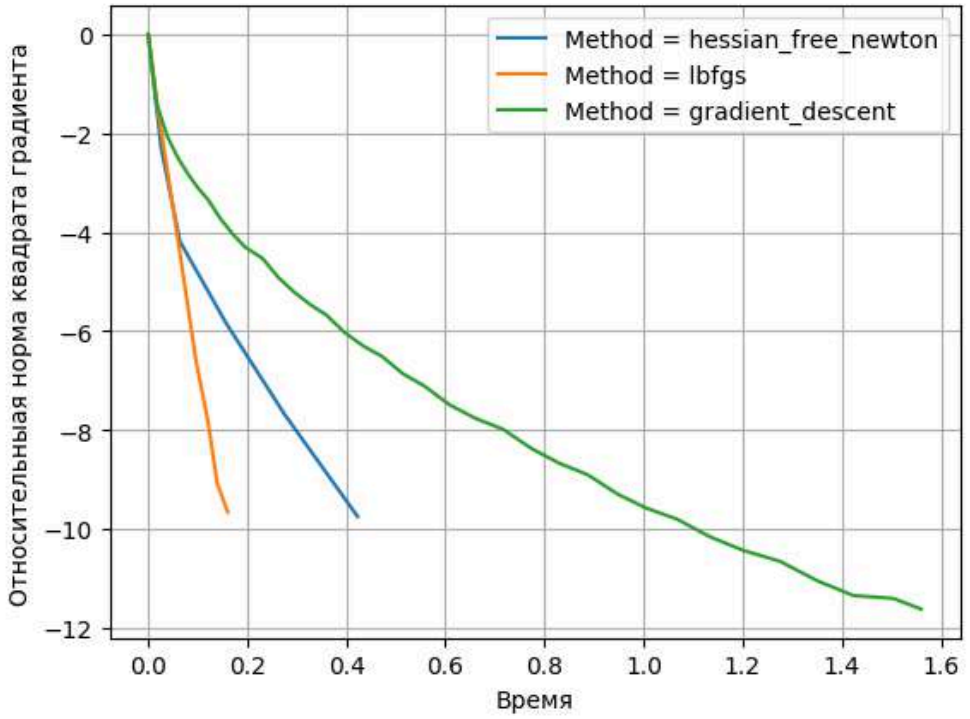

Рис. 7. Сравнение методов. Датасет w8a. Относительная норма квадрата градиента от времени

Датасет gisette_scale

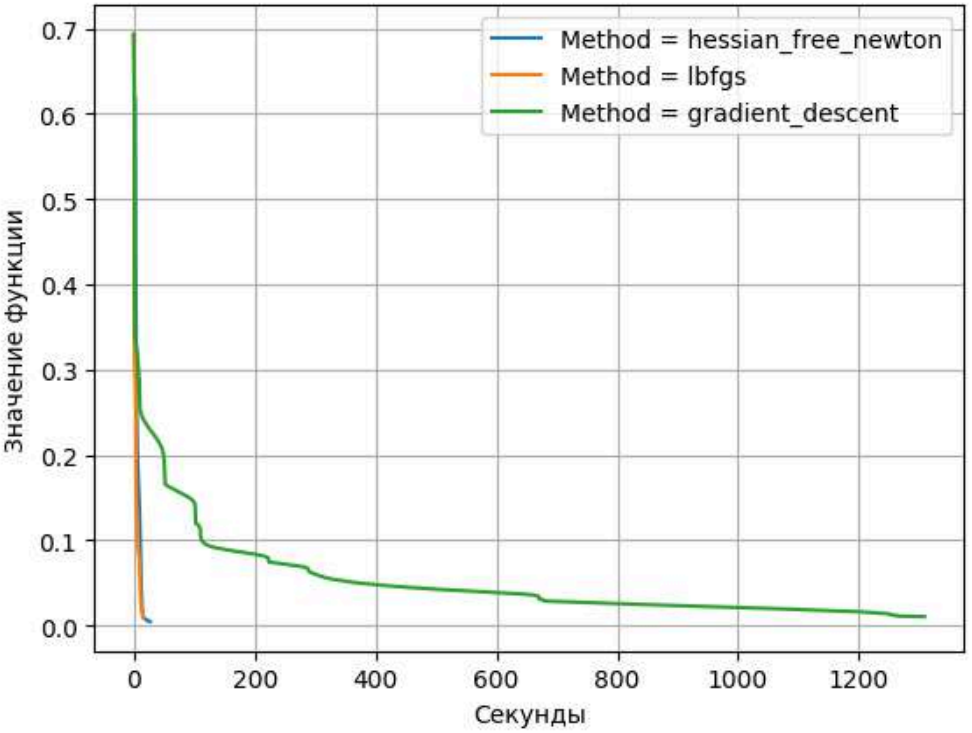

Рис. 8. Сравнение методов. Датасет gisette_scale. Функция от времени.

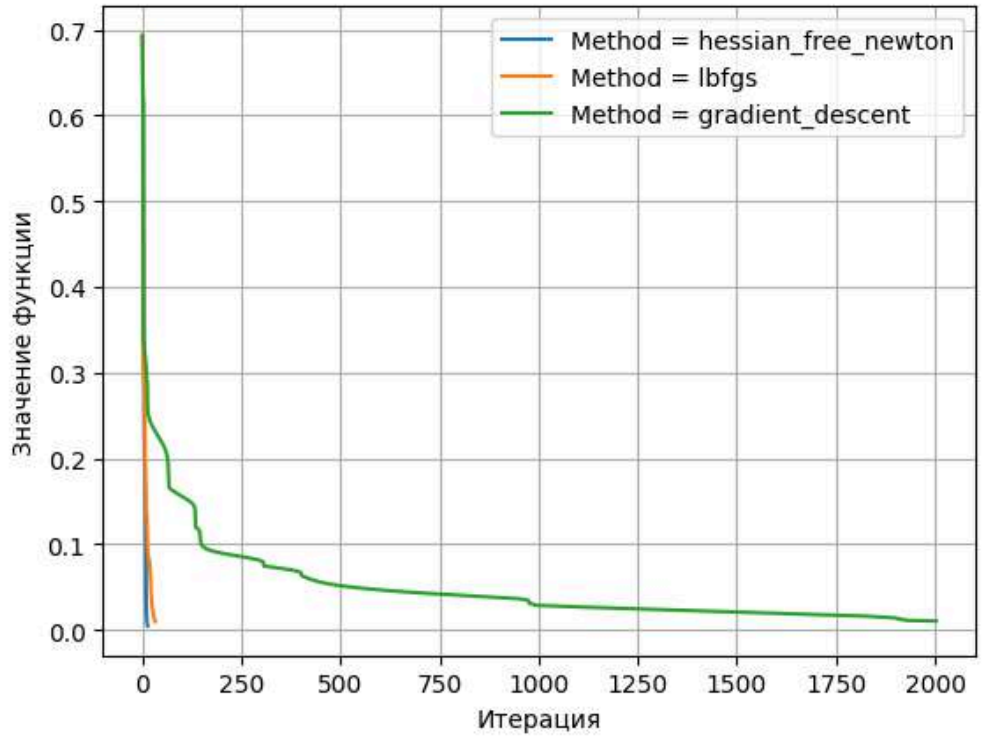

Рис. 9. Сравнение методов. Датасет gisette_scale. Функция от итераций

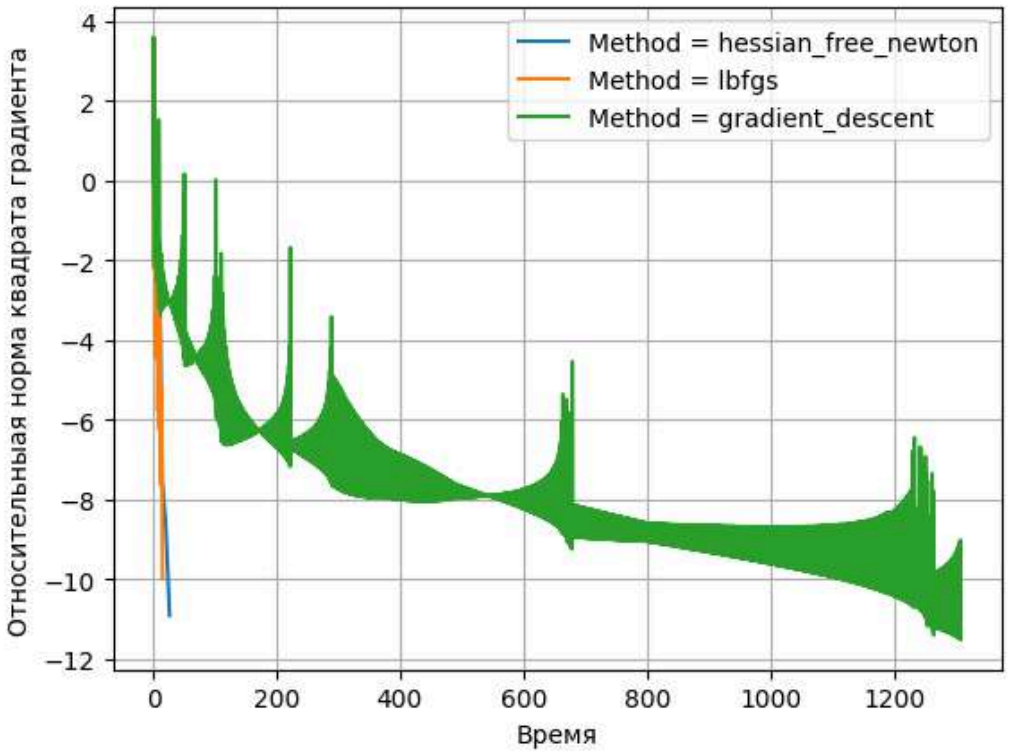

Рис. 10. Сравнение методов. Датасет gisette_scale. Относительная норма квадрата градиента от времени

Датасет real-sim

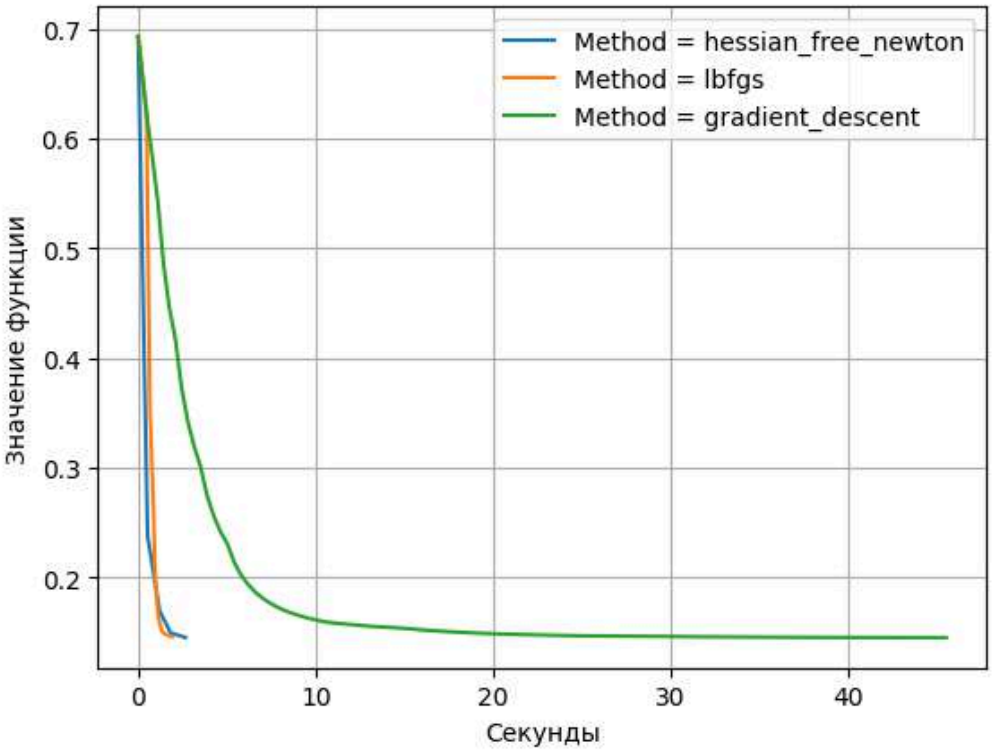

Рис. 11. Сравнение методов. Датасет real-sim. Функция от времени.

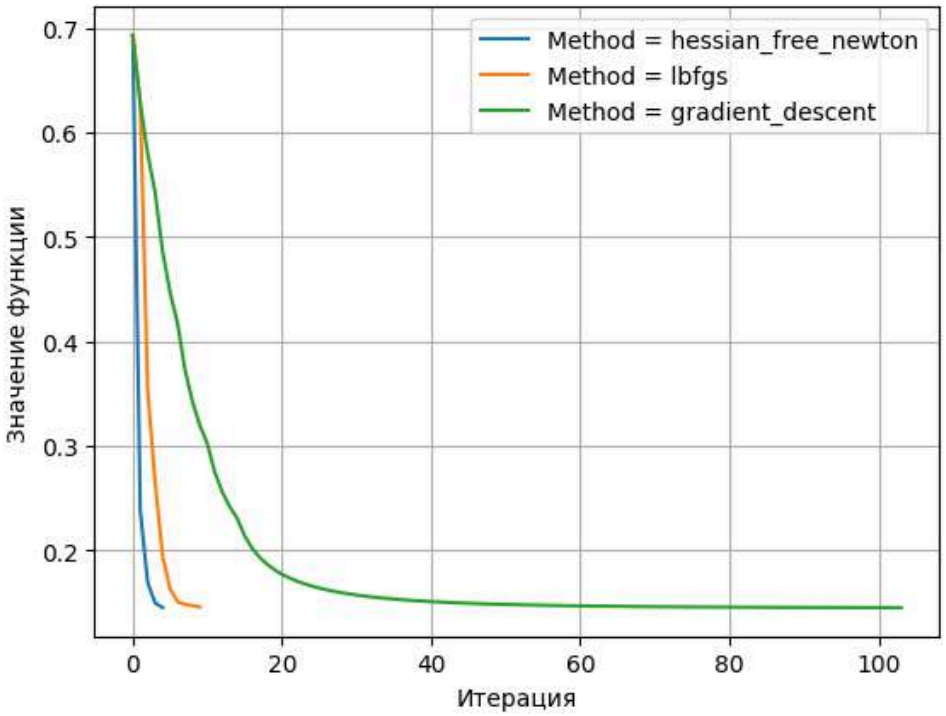

Рис. 12. Сравнение методов. Датасет real-sim. Функция от итераций

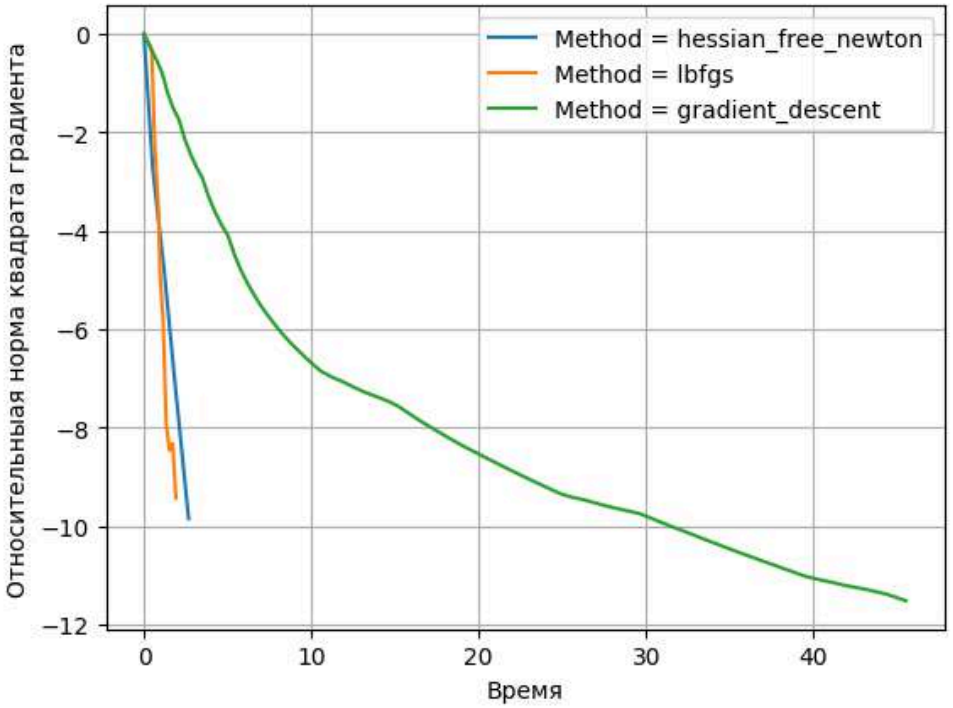

Рис. 13. Сравнение методов. Датасет real-sim. Относительная норма квадрата градиента от времени

Датасет news20.binary

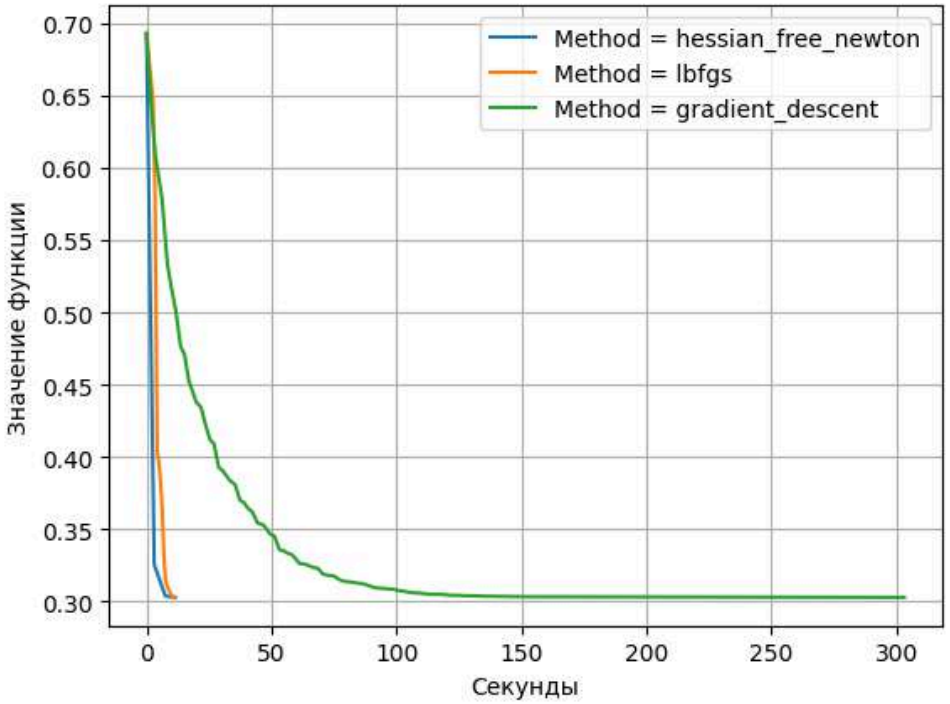

Рис. 14. Сравнение методов. Датасет news20.binary. Функция от времени.

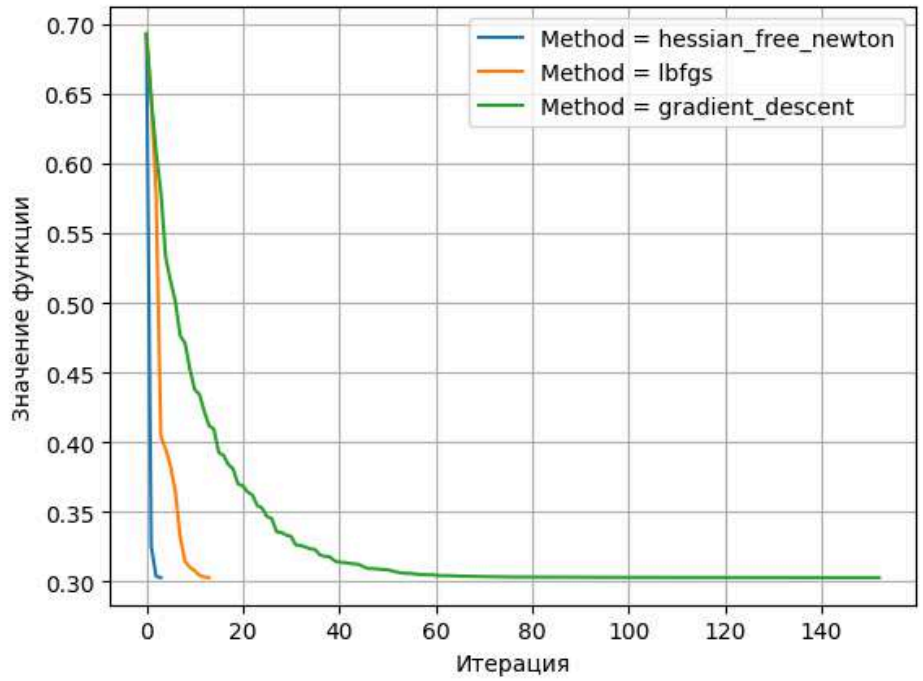

Рис. 15. Сравнение методов. Датасет news20.binary. Функция от итераций

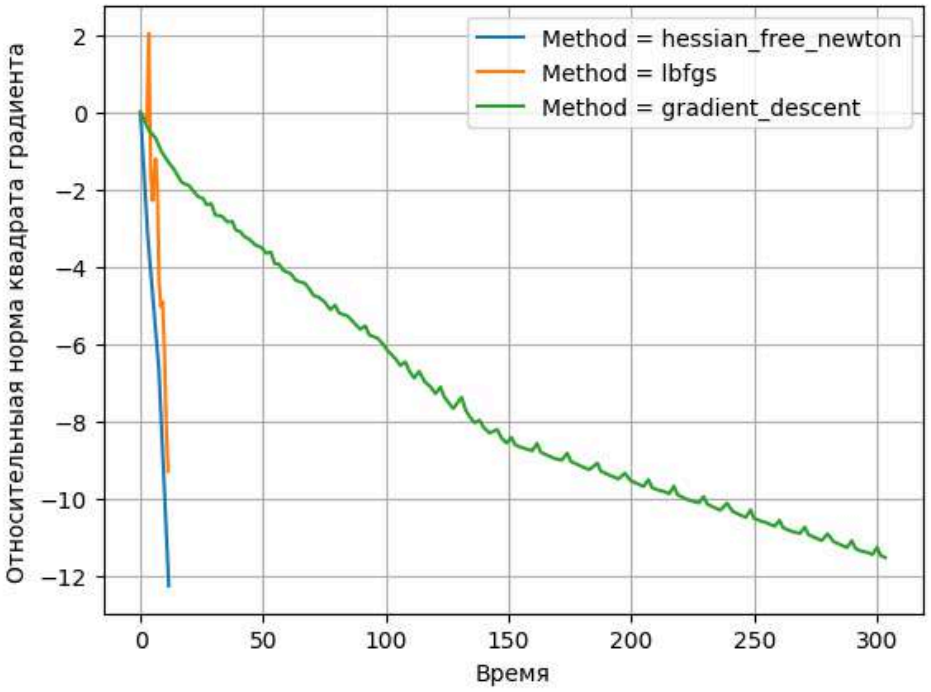

Рис. 16. Сравнение методов. Датасет news20.binary. Относительная норма квадрата градиента от времени

Датасет rcv1_train.binary

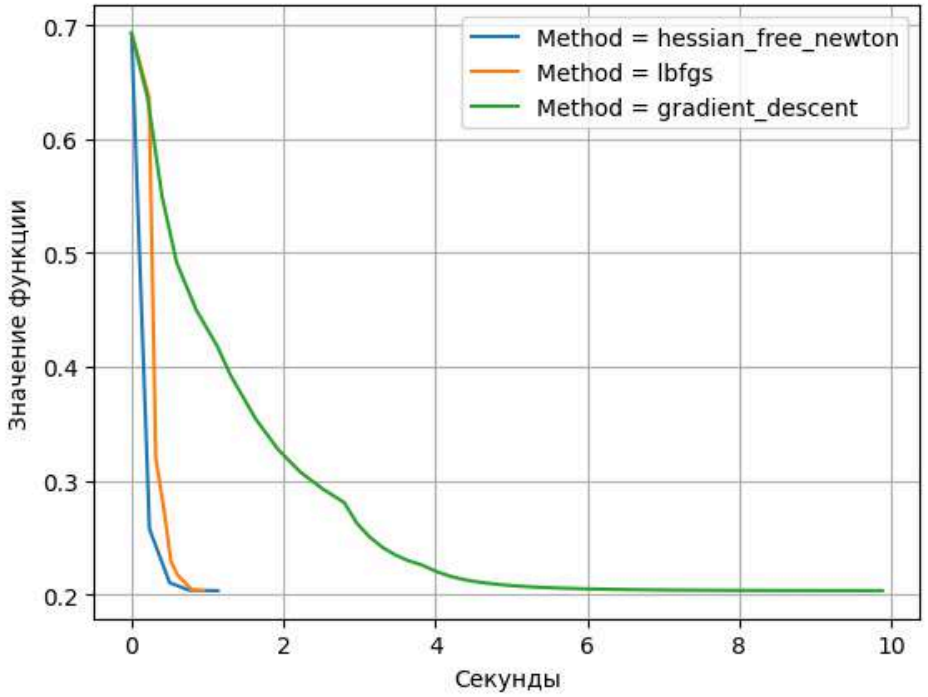

Рис. 17. Сравнение методов. Датасет rcv1_train.binary. Функция от времени.

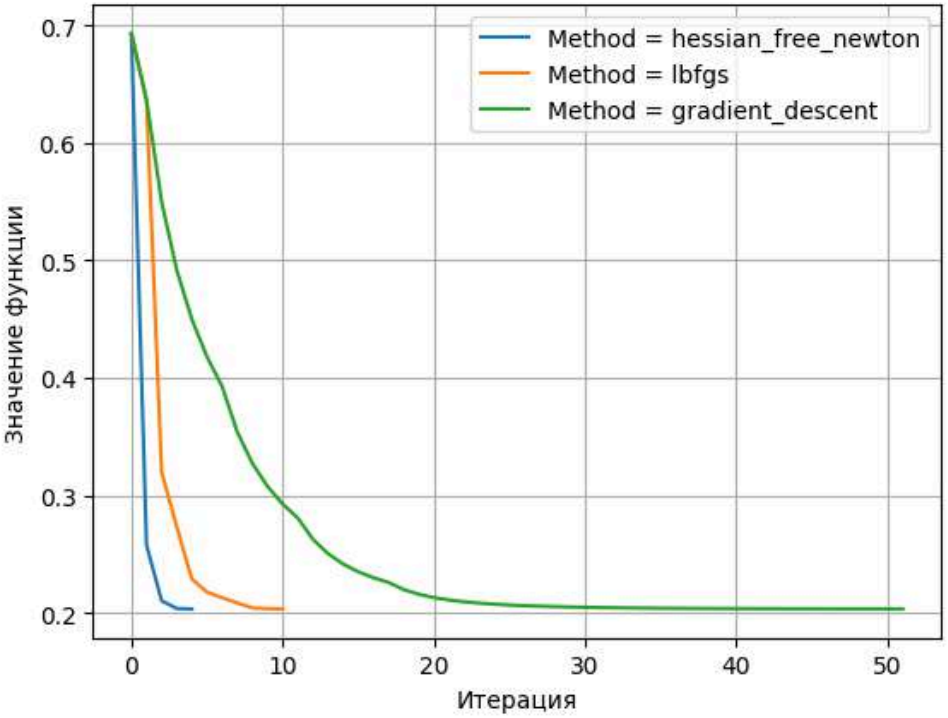

Рис. 18. Сравнение методов. Датасет rcv1_train.binary. Функция от итераций

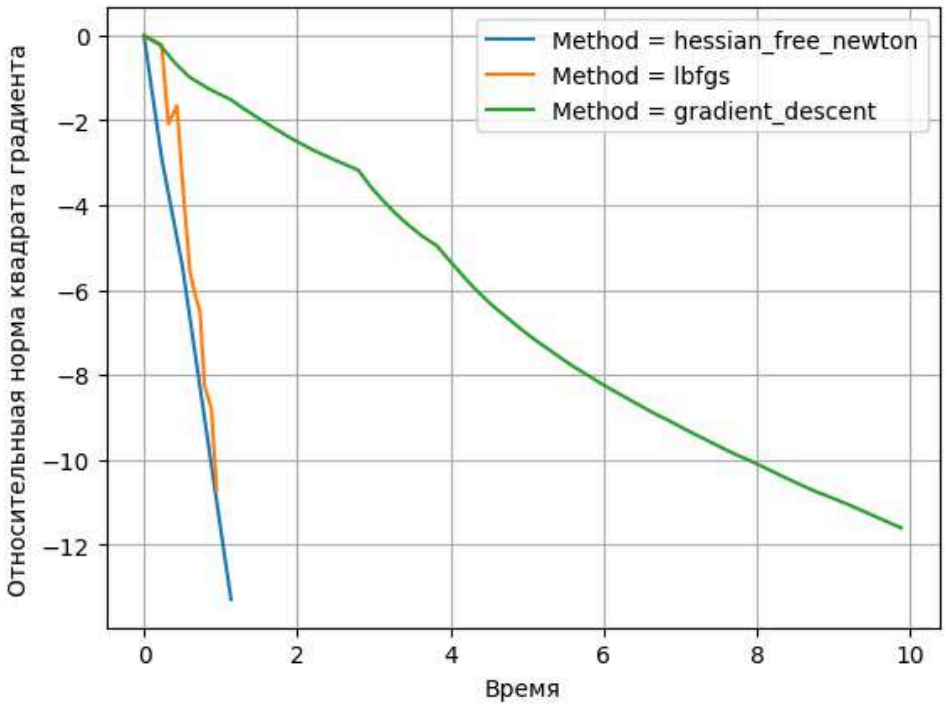

Рис. 19. Сравнение методов. Датасет rcv1_train.binary. Относительная норма квадрата градиента от времени

Исходя из результата эксперимента L-BFGS и усеченный метод Ньютона (УМН) показывают примерно одинаковую результативность, метод градиентного спуска намного медленнее справляется со своей задачей. В каких то местах L-BFGS справляется чуть медленнее со своей задачей по сравнению с УМН. Но также стоит учитывать что для УМН в этом эксперименте используется аналитическая форма записи умноженного Гессиана на вектор, в конкретном случае для лог регрессии требуется o(n^2) операций, в противном бы случае пришлось считать конечную разность для произведения гессиана на вектор, тут уже несомненно бы выигрывал по производительности L-BFGS так как использует только информацию о градиентах и координатах *x* на предыдущих шагах.

4. Сравнение метода сопряженных градиентов и L-BFGS на квадратичной функции

Аналитическое вычисление наилучшей длины шага $\alpha_j = argmin_{\alpha \geq 0} f(x_j + \alpha d_j)$:

$Ax_{\#} = b$; $x_0 -$ начальная точка; $\Rightarrow$

$x_{\#} - x_0 = \sum_{i=0}^{n-1} \alpha_i d_i$ | домножим обе части на $d_j^T A$ $\Rightarrow$

$$d_j^T A(x_\# - x_0) = \sum_{i=0}^{n-1} \alpha_i d_j^T A d_i = \alpha_j d_j^T A d_j \quad \Rightarrow$$

$$\alpha_j = \frac{d_j^T A(x_\# - x_0)}{d_j^T A d_j} = \frac{d_j^T (b - A x_0)}{d_j^T A d_j} = - \frac{g_0^T d_j}{d_j^T A d_j}$$

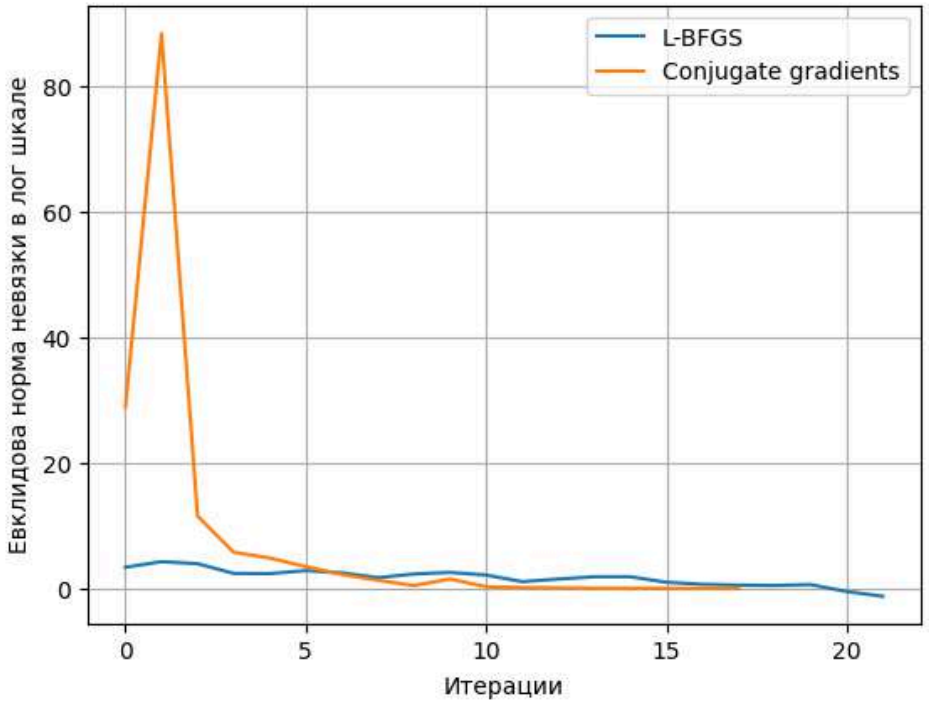

Рис. 20. Сравнение метода сопряженных градиентов и L-BFGS на квадратичной функции. Без использование оптимального шага α

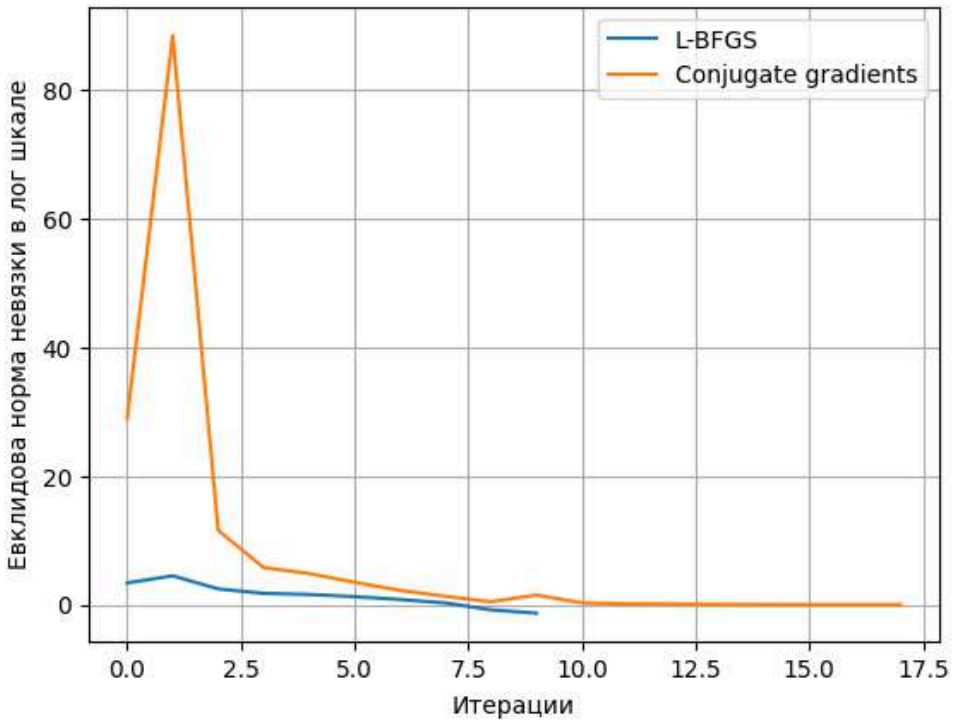

Рис. 21. Сравнение метода сопряженных градиентов и L-BFGS на квадратичной функции. Используется стандартный размер памяти – 10

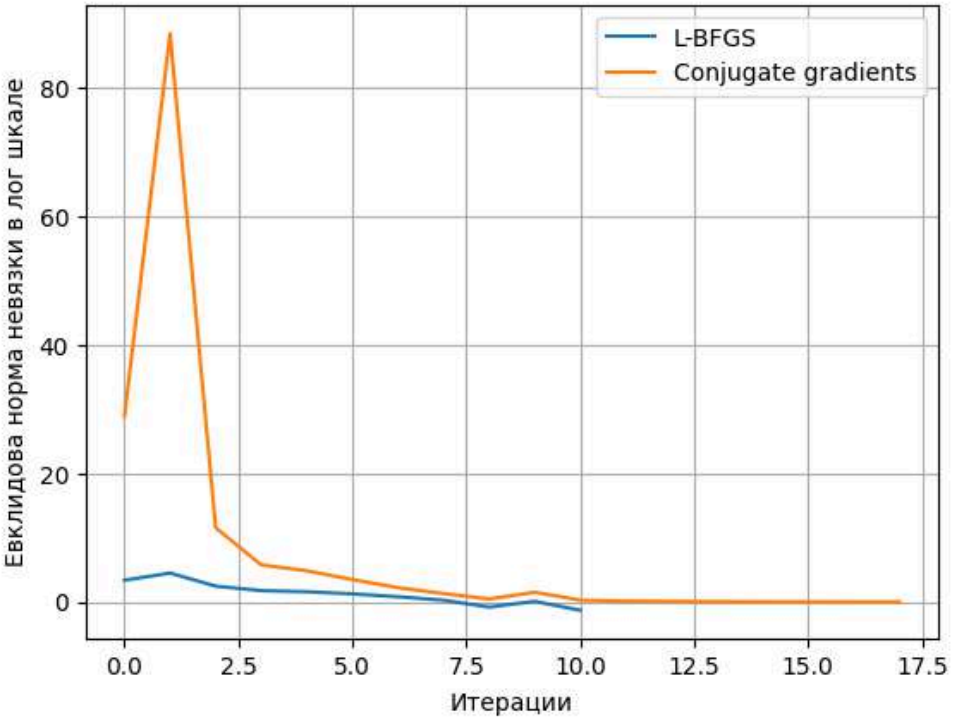

Рис. 22. Сравнение метода сопряженных градиентов и L-BFGS на квадратичной функции. Используется размер памяти – 1

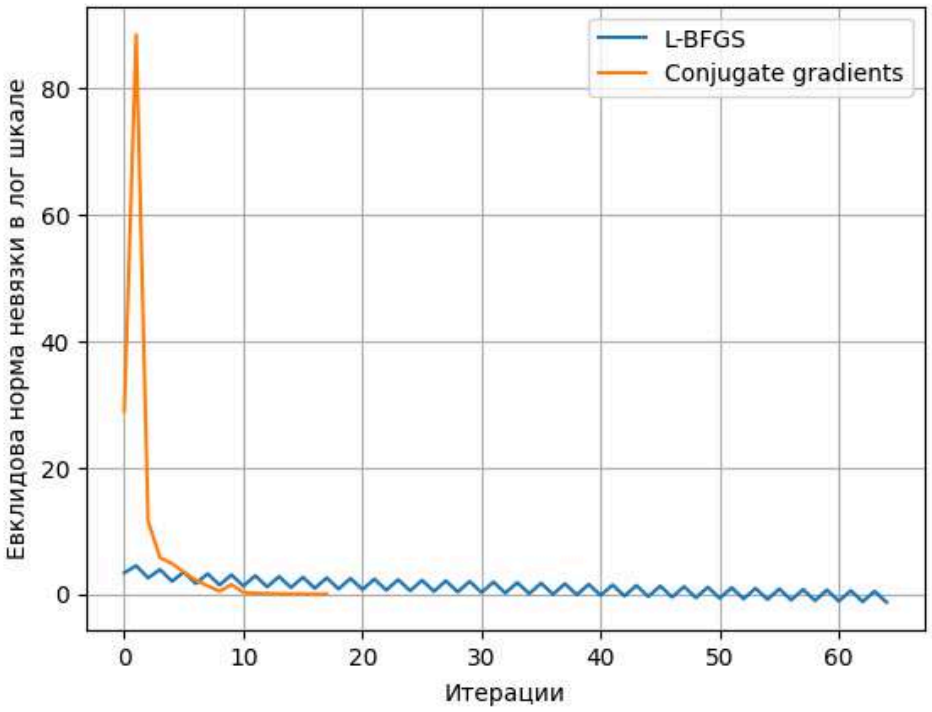

Рис. 23. Сравнение метода сопряженных градиентов и L-BFGS на квадратичной функции. Используется размер памяти – 0

При использовании информации об оптимальной длине шага быстрее по итерациям справляется метод L-BFGS (не считая случая с историей 0). При

использовании длины истории равной 1 L-BFGS чуть хуже справляется с задачей чем с 10, так как ухудшается точность оптимизации, и требуется больше итераций для удовлетворения критерия останова. При истории равной 0 метод работает как градиентный спуск, который в свою очередь имеет линейную скорость сходимости (усеченный метод ньютона сверхлинейную).

5. Какая точность оптимизации нужна в реальных задачах?

Датасет w8a

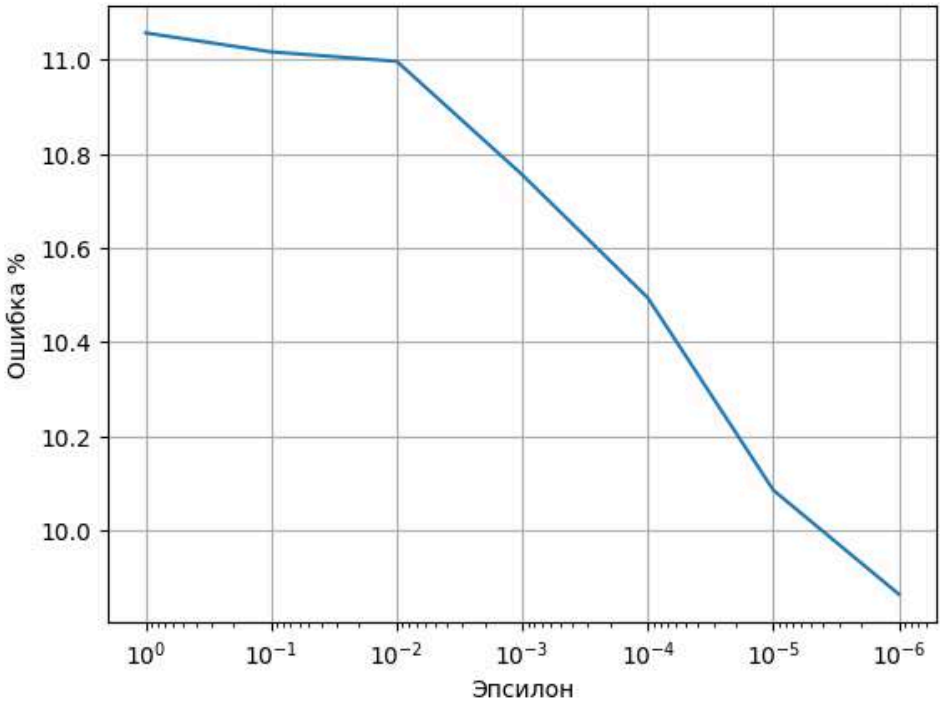

Рис. 24. Зависимость процента ошибки от параметра ε точности оптимизации.

Датасет gisette_scale

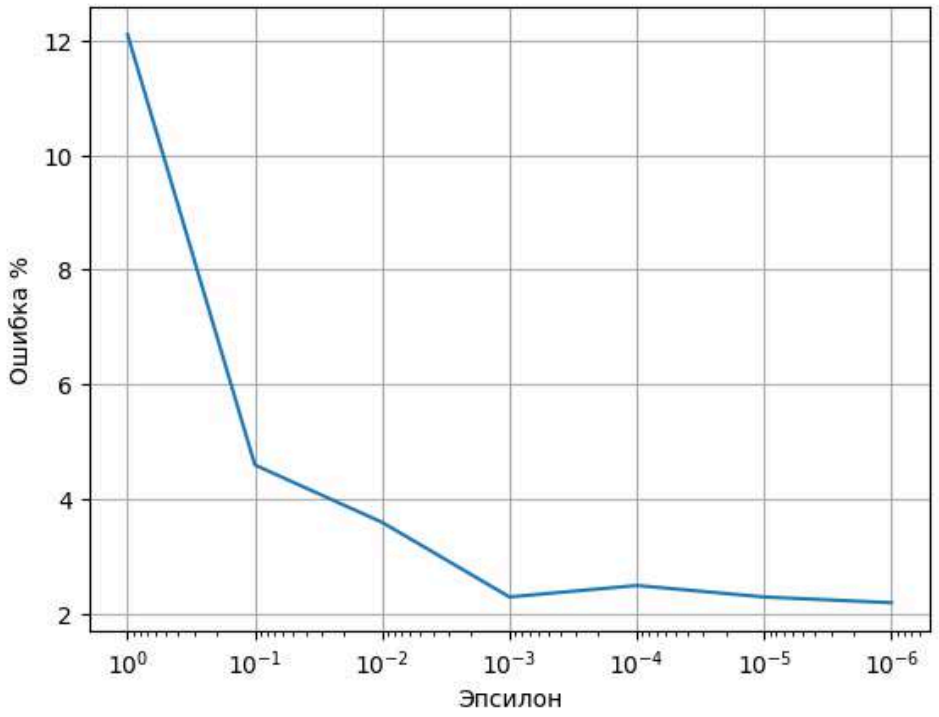

Рис. 25. Зависимость процента ошибки от параметра ε точности оптимизации.

Процент ошибки вычислялся как количество не совпадающих значений $\widehat{b}_{test}$ и b_{test} по всем направлением пространства деленное на размер пространства b, т.е. $100 * count(\widehat{b}_{test} != b_{test}) / size(b)$. Как видно из графиков с увеличением точности оптимизационной задачи лог регрессии ошибка уменьшается. В случае w8a уже изначальная точка x_0 была близка к оптимуму, в случае gillette scale ошибка значительно уменьшается с уменьшением ε. Учитывая что решается задача логистической регрессии, то есть создания такого алгоритма который бы по настроенному параметру модели $\widehat{x}$ бинарно классифицировал, принимая на вход матрицу признаков A, наблюдения к одному или другому классу, т.е. $\widehat{b} = sign(A\ \widehat{x})$. Следовательно чем лучше решена задача оптимизации (что достигается в том числе за счет маленького значения ε) тем лучше наблюдения классифицируются т.е. меньше ошибка классификации.