

# Лабораторная работа 2:

## Продвинутые методы безусловной оптимизации

Выполнила Симакова Елизавета,  
студентка 1 курса магистратуры ВБИБ НИУ ВШЭ Санкт-Петербург

25 мая 2025 г.

### Содержание

1	Эксперимент 2.1. Зависимость числа итераций метода сопряженных градиентов от числа обусловленности и размерности пространства	2
2	Эксперимент 2.2. Выбор размера истории в методе L BFGS	6
3	Эксперимент 2.3. Сравнение методов на реальной задаче логистической регрессии	9
4	Эксперимент 2.4. Сравнение метода сопряженных градиентов и L-BFGS на квадратичной функции	19
5	Эксперимент 2.5. Какая точность оптимизации нужна в реальных задачах?	22

# 1 Эксперимент 2.1. Зависимость числа итераций метода сопряженных градиентов от числа обусловленности и размерности пространства

## Описание эксперимента

Целью данного эксперимента является исследование влияния числа обусловленности  $\kappa$  и размерности пространства  $n$  на число итераций метода сопряженных градиентов, необходимое для достижения заданного уровня точности.

### Основные параметры эксперимента:

- Глобальный seed генератора случайных чисел: `np.random.seed(42)`;
- Количество независимых запусков для каждого  $(n, \kappa)$ :  
20 (массив `seeds = np.random.randint(1, 10000, 20)`);
- Массив значений размерностей:  $n \in \{10, 100, 1000, 10000\}$ ;
- 20 значений числа обусловленности, равномерно распределённых по логарифмической сетке от 1 до 1000. Используется функция `np.logspace(0, 3, num=20)`;
- Максимальное число итераций метода:  $T_{\max} = 2500$ ;
- Начальная точка:  $x_0 = 0 \in \mathbb{R}^n$ ;
- Генерация вектора  $b$ : независимые компоненты  $b_i \sim \mathcal{N}(0, 1)$   
(используется `np.random.randn(n)`, где  $n$  - размерность);
- Генерация диагонали матрицы  $A$ : сначала  $r_j \sim \mathcal{U}(0, 1)$ , затем  $a_j = r_j(\kappa - 1) + 1$ , причём при  $n \geq 2$  дополнительно фиксируются  $a_1 = 1$ ,  $a_2 = \kappa$ . Матрица хранится как разрежённая диагональная через `scipy.sparse.diags`.
- Функция, в которой реализован метод сопряженных градиентов, используется со следующими аргументами: `conjugate_gradients(lambda x: A @ x, b, x0, max_iter=max_iter, trace=True)`.

Для каждой комбинации  $(n, \kappa)$  проводится 20 независимых запусков (разные seed), строятся кривые зависимости числа итераций от  $\kappa$ . Все отдельные кривые отображаются пунктиром полупрозрачным цветом, среднее по 20 запускам — сплошной линией. Цвета соответствуют:

- **красный** для  $n = 10$ ;
- **синий** для  $n = 100$ ;
- **зелёный** для  $n = 1000$ ;
- **коричневый** для  $n = 10000$ .

## Результаты эксперимента

На рисунке 1 представлены зависимости среднего числа итераций метода сопряженных градиентов  $T(n, \kappa)$  от числа обусловленности  $\kappa$  при разных размерностях  $n$ . Каждое семейство кривых соответствует одному значению  $n$  (цвет обозначен в легенде).

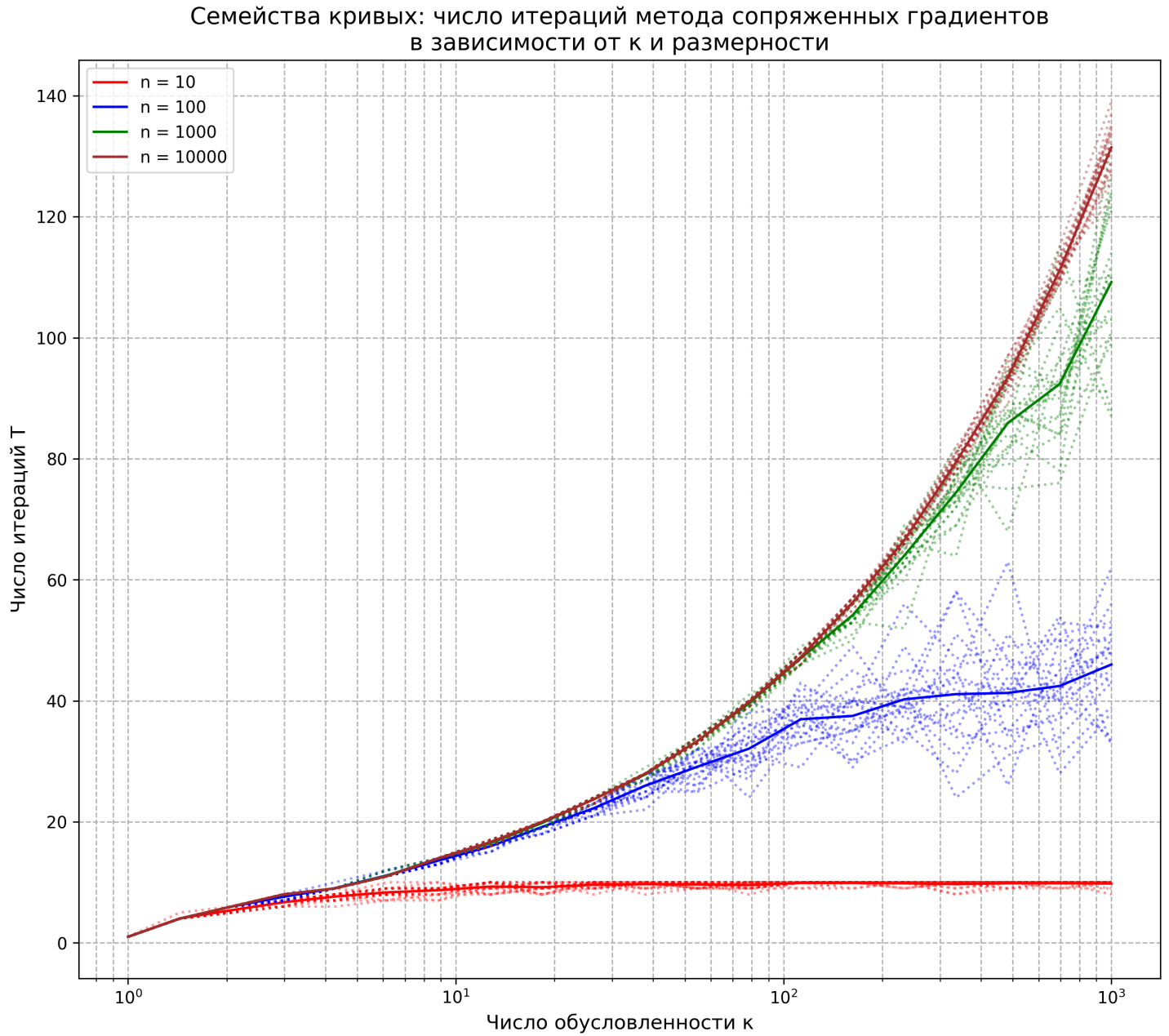


Рис. 1: Зависимость числа итераций метода сопряженных градиентов от числа обусловленности при различных размерностях. Пунктирные кривые — отдельные запуски, сплошные — их среднее.

На рисунке 2 представлены результаты эксперимента 2.3 лабораторной работы №1 - зависимость среднего числа итераций градиентного спуска  $T(n, \kappa)$  от числа обусловленности  $\kappa$  при разных размерностях  $n$ . Каждое семейство кривых соответствует одному значению  $n$  (цвет обозначен в легенде).

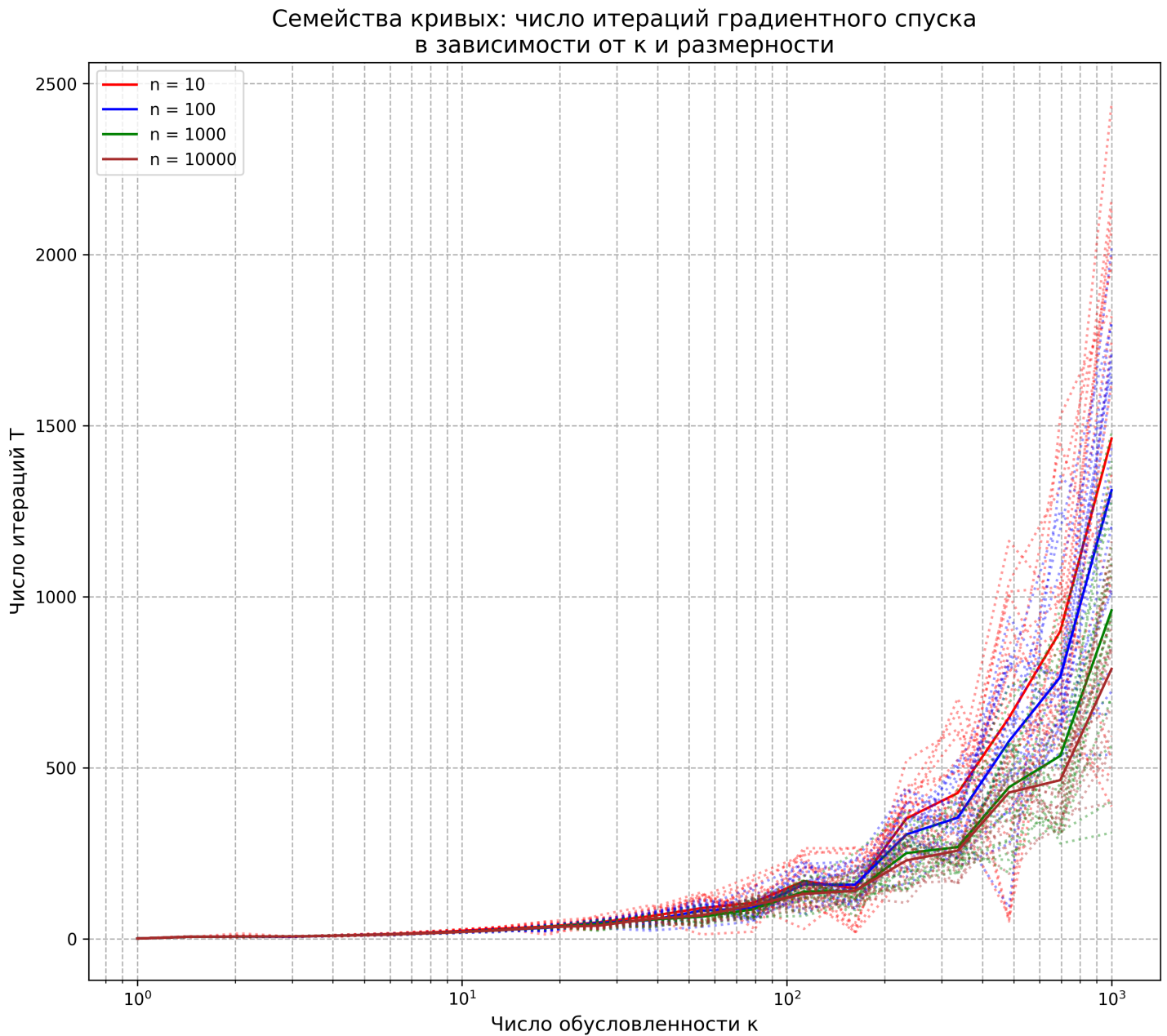


Рис. 2: Зависимость числа итераций градиентного спуска от числа обусловленности при различных размерностях. Пунктирные кривые — отдельные запуски, сплошные — их среднее.

Проанализировав график 8 и результат предыдущего эксперимента график 9, сравнив результат метода сопряженных градиентов с результатом градиентного спуска можно сделать следующие выводы:

1. **Зависимость от  $\kappa$ .** Для квадратичной функции число итераций градиентного спуска растёт почти линейно с увеличением числа обусловленности  $\kappa$ . У метода сопряжённых градиентов рост существенно медленнее. При  $\kappa \approx 1$  оба метода сходятся за один шаг. В диапазоне  $\kappa \in [10, 10^2]$  градиентному спуску требуется от десятков до сотен итераций, тогда как методу сопряжённых градиентов — лишь несколько десятков. При  $\kappa \sim 10^3$  градиентный спуск требует  $T \sim 1000\text{--}1500$  шагов, а метод сопряжённых градиентов — лишь  $T \sim 100\text{--}150$  и никогда не превышает  $n$ .
2. **Зависимость от размерности  $n$ .** При увеличении  $n$  число итераций градиентного спуска слегка уменьшается, однако зависимость от  $\kappa$  остаётся выраженной. У метода сопряжённых градиентов число итераций не превышает  $n$ , что наглядно подтверждается графиками: при  $n = 10$  число шагов

ограничено 10, при  $n = 100$  максимум  $T = 47$ , при  $n = 1000$  и  $n = 10000$  влияние  $\kappa$  становится сильнее, но  $T \leq n$ . Наблюдается более чёткое разбиение по семействам кривых: при больших  $n$  различия между задачами с разной обусловленностью становятся заметнее.

3. **Сходимость и стабильность.** Метод сопряжённых градиентов демонстрирует значительно меньший разброс по числу итераций: семейства кривых для фиксированных параметров более плотные. При одинаковых задачах метод сопряжённых градиентов требует в разы меньше итераций, чем градиентный спуск, особенно при высокой обусловленности.

**Общий вывод:** метод сопряжённых градиентов при решении квадратичных задач демонстрирует существенно более быструю сходимость ( $\mathcal{O}(\sqrt{\kappa})$  против  $\mathcal{O}(\kappa)$ ) и ограниченность числом измерений ( $T \leq n$ ), что делает его предпочтительным по сравнению с градиентным спуском при средних и высоких значениях  $\kappa$  и больших размерностях  $n$ .

## 2 Эксперимент 2.2. Выбор размера истории в методе L BFGS

### Описание эксперимента

Целью данного эксперимента является исследование влияния размера истории в методе L-BFGS на поведение метода.

**Теоретические оценки:** Оценим размер требуемой памяти и сложность итерации метода L-BFGS в зависимости от размера истории  $l$  и размерности пространства  $n$ , не считывая сложность оракула:

- **Требуемая память:** храним историю  $H$  и в каждом элементе пара векторов размера  $n$ , следовательно  $O(n \times l)$ , где  $n$  — размерность пространства,  $l$  — размер истории.
- **Сложность одной итерации:** рекурсия по истории размера  $l$ , следовательно  $O(n \times l)$ .

### Основные параметры эксперимента:

- **Тестовая функция:** логистическая регрессия с  $L_2$ -регуляризацией на датасете `gisette_scale.bz2` с сайта LIBSVM.
- $\lambda = \frac{1}{m}$ , где  $m$  — число примеров в датасете.
- Начальная точка:  $x_0 = 0$ .
- Значения параметра размера истории:

$$l \in \{0, 1, 5, 10, 50, 100, 500, 1000\}.$$

- Для каждого  $l$  метод L-BFGS запускается один раз, при этом на каждой итерации фиксируются:
  - квадрат нормы градиента  $\|\nabla f(x_k)\|^2$ ,
  - время работы  $t_k$ .
- Цвета кривых на графиках назначаются по порядку. Точные обозначения указаны в легенде.

## Результаты эксперимента

Ниже представлены результаты эксперимента в виде двух графиков.

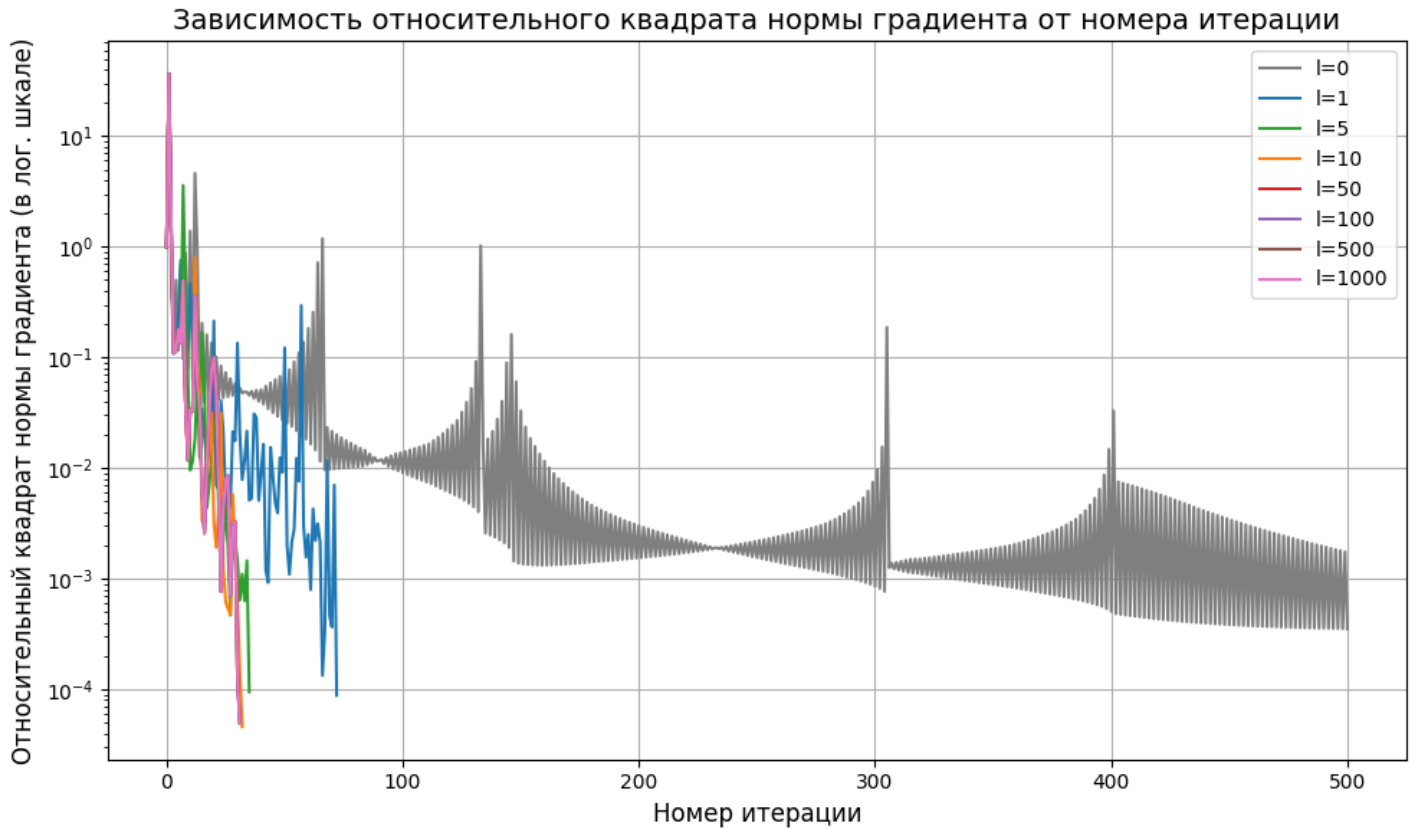


Рис. 3: Зависимость относительного квадрата нормы градиента от номера итерации.

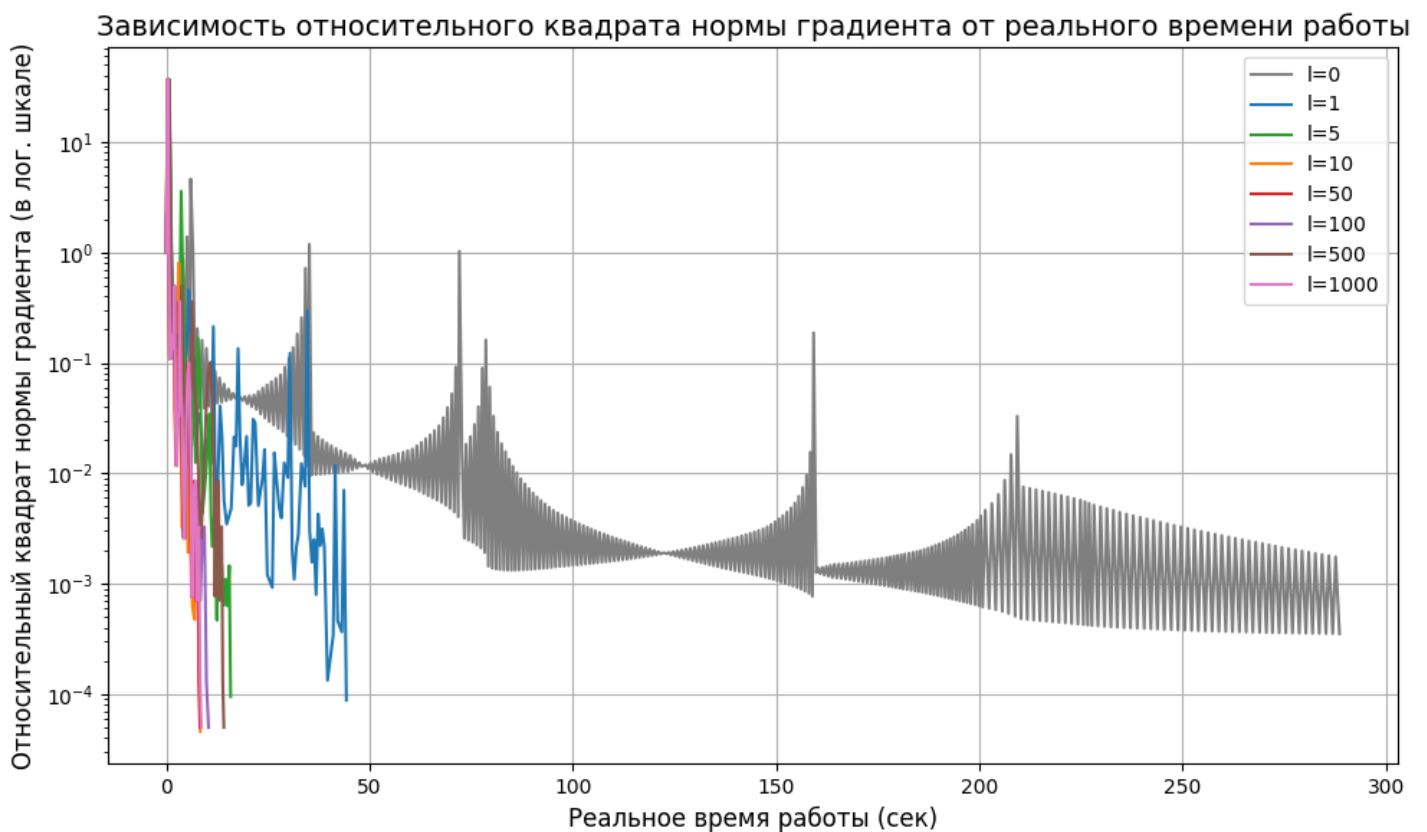


Рис. 4: Зависимость относительного квадрата нормы градиента от реального времени работы

**Проанализировав полученные результаты можно сделать следующие наблюдения и выводы:**

1. При  $l = 0$  траектория метода напоминает траекторию градиентного спуска и демонстрирует наихудшие темпы сходимости — кривая отстаёт от всех остальных.
2. Разница между  $l = 1$  и  $l = 5$  заметна, но уже при  $l \geq 10$  все траектории кривых накладываются друг на друга: дальнейшее увеличение  $l$  (до 1000) не даёт существенного ускорения.
3. По числу итераций с увеличением  $l$  сходимость становится более «резкой», однако каждая итерация требует больше вычислений, поэтому по времени выигрыш при  $l \geq 10$  почти нивелируется.
4. При малых  $l$  кривая зашумлена — норму градиента на отдельных итерациях не удаётся стабильно уменьшать из-за неточного приближения гессиана.
5. Наилучший значения работы, компромисс между скоростью и сходимостью достигается при  $l \approx 10$ .



### 3 Эксперимент 2.3. Сравнение методов на реальной задаче логистической регрессии

#### Описание эксперимента

Целью данного эксперимента является сравнение трёх методов оптимизации — усечённого метода Ньютона, метода L-BFGS и классического градиентного спуска — на задаче логистической регрессии с  $L_2$ -регуляризацией. В качестве тестовых данных используются пять наборов LIBSVM: `w8a`, `gisette_scale.bz2`, `real-sim.bz2`, `news20.binary.bz2` и `rcv1_train.binary.bz2`.

#### Основные параметры эксперимента:

- $\lambda = \frac{1}{m}$ , где  $m$  — число примеров в каждом датасете.
- Начальная точка:  $x_0 = \mathbf{0}$ .
- Все методы запускаются с параметрами по умолчанию.
- Основные характеристики датасетов:
  - `w8a.t`:
    - \* Количество объектов:  $n = 14\,951$
    - \* Число признаков:  $d = 300$
    - \* Количество ненулевых значений:  $\text{nnz} = 174\,276$
  - `rcv1_train.binary.bz2`:
    - \* Количество объектов:  $n = 20\,242$
    - \* Число признаков:  $d = 47\,236$
    - \* Количество ненулевых значений:  $\text{nnz} = 1\,498\,952$
  - `real-sim.bz2`:
    - \* Количество объектов:  $n = 72\,309$
    - \* Число признаков:  $d = 20\,958$
    - \* Количество ненулевых значений:  $\text{nnz} = 3\,709\,083$
  - `news20.binary.bz2`:
    - \* Количество объектов:  $n = 19\,996$
    - \* Число признаков:  $d = 1\,355\,191$
    - \* Количество ненулевых значений:  $\text{nnz} = 9\,097\,916$
  - `gisette_scale.bz2`:
    - \* Количество объектов:  $n = 6\,000$
    - \* Число признаков:  $d = 5\,000$
    - \* Количество ненулевых значений:  $\text{nnz} = 29\,729\,997$

В ходе каждой итерации независимо для каждого метода фиксируется:

- значение функции  $f(x_k)$ ;
- время работы  $t_k$ ;
- квадрат нормы градиента  $\|\nabla f(x_k)\|^2$ .

Цвета кривых соответствуют методам: градиентный спуск — синий, усечённый Ньютон — зелёный, L-BFGS — красный. Все три метода нанесены на одни и те же графики для сравнения.

## Результаты эксперимента

Ниже представлены результаты эксперимента в виде пятнадцати графиков, разделённых на три секции. Каждая секция снабжена пояснением, графики в секции отранжированы в порядке увеличения  $\mu\mu z$ . В завершение приводится общий вывод по эксперименту.

### Зависимость значения функции против номера итерации

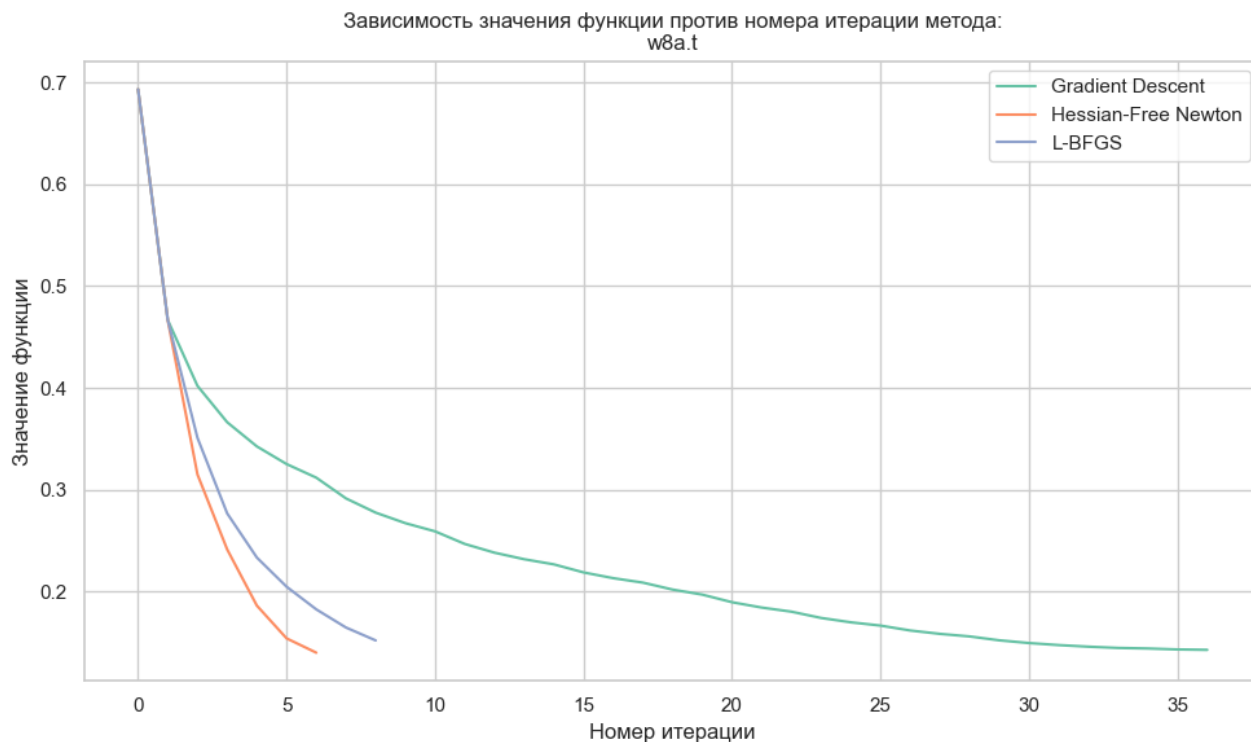


Рис. 5: w8a

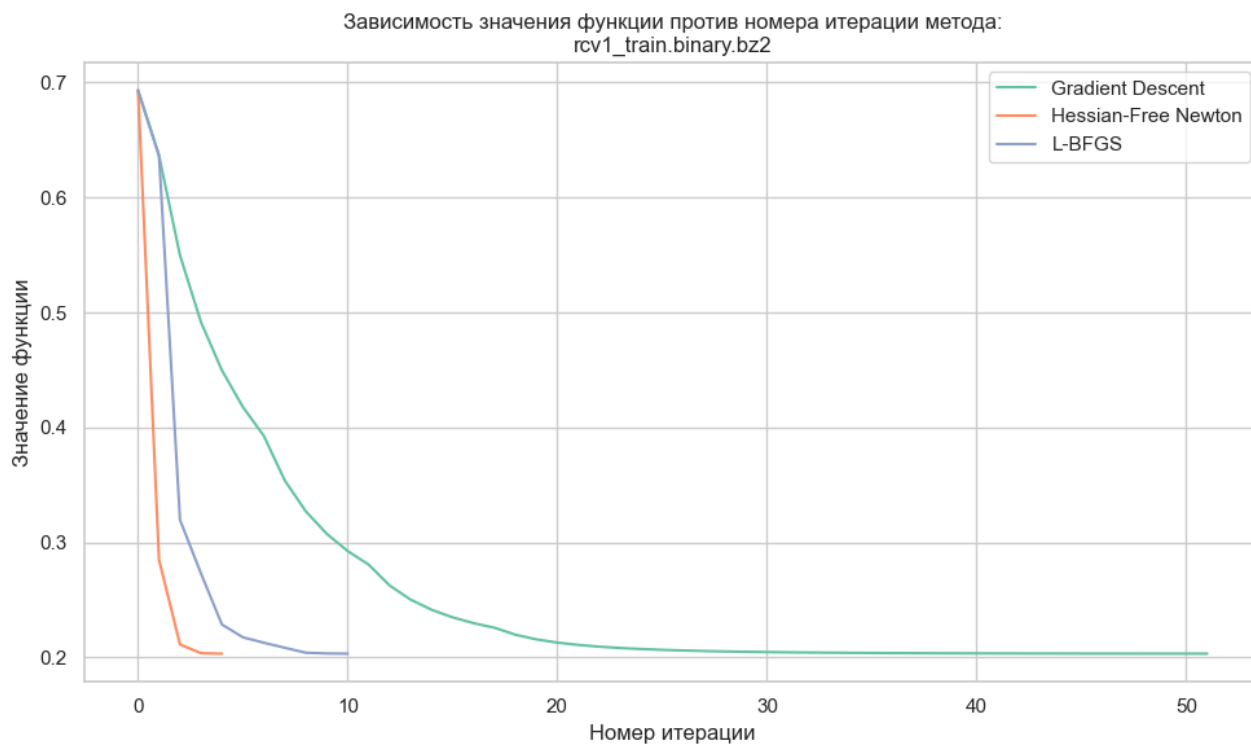


Рис. 6: rcv1\_train.binary

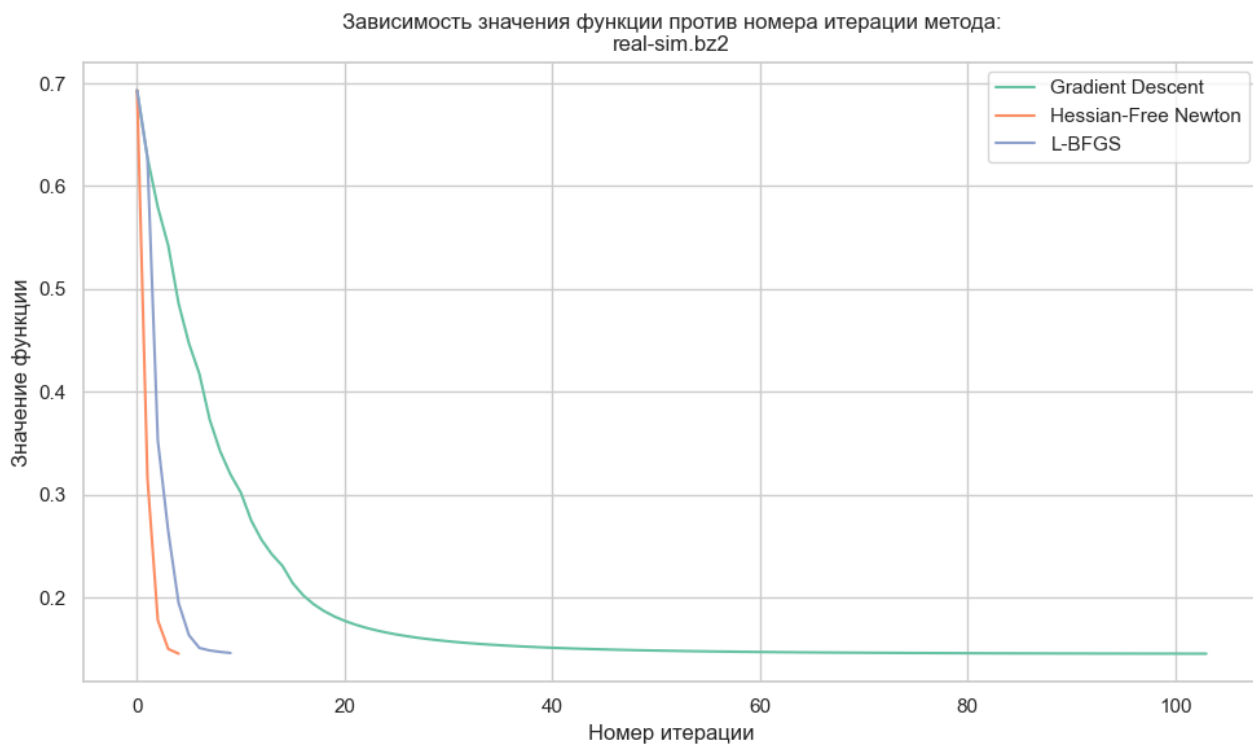


Рис. 7: real-sim.bz2

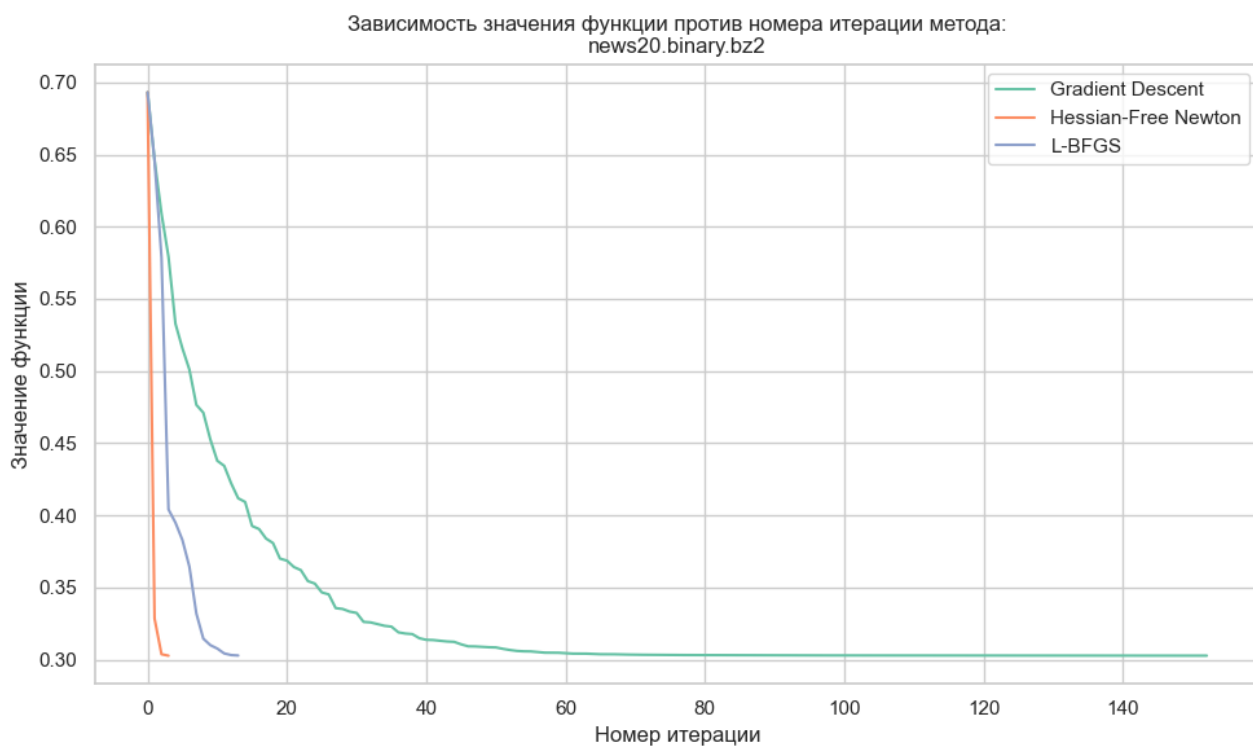


Рис. 8: news20.binary.bz2

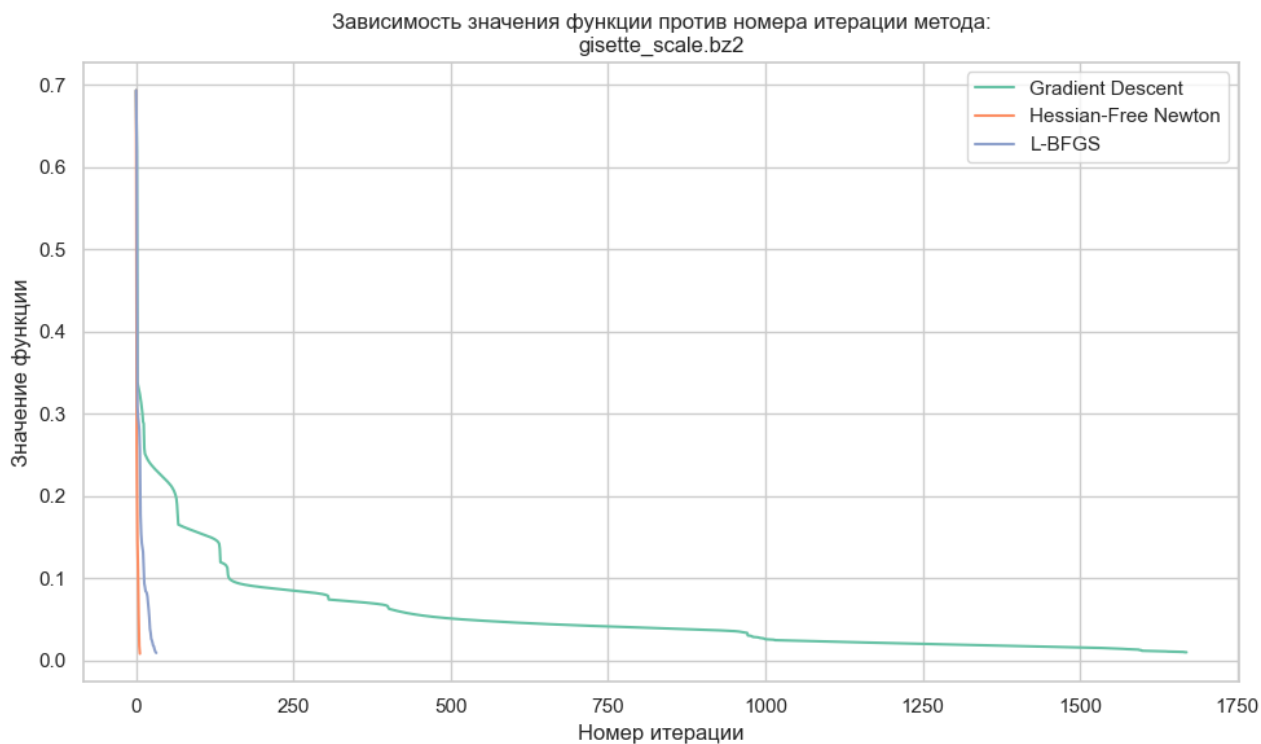


Рис. 9: gisette\_scale.bz2

**Комментарий.** На всех пяти графиках видно, что усечённый метод Ньютона (HFN) и L-BFGS сходятся значительно быстрее по числу итераций, чем градиентный спуск. HFN требует меньше всего итераций на всех датасетах. Замечается также, что с ростом числа ненулевых значений возрастает и число итераций, особенно у градиентного спуска. Например, на датасете gisette\_scale градиентный спуск требует в сотни раз больше итераций, чем на w8a.

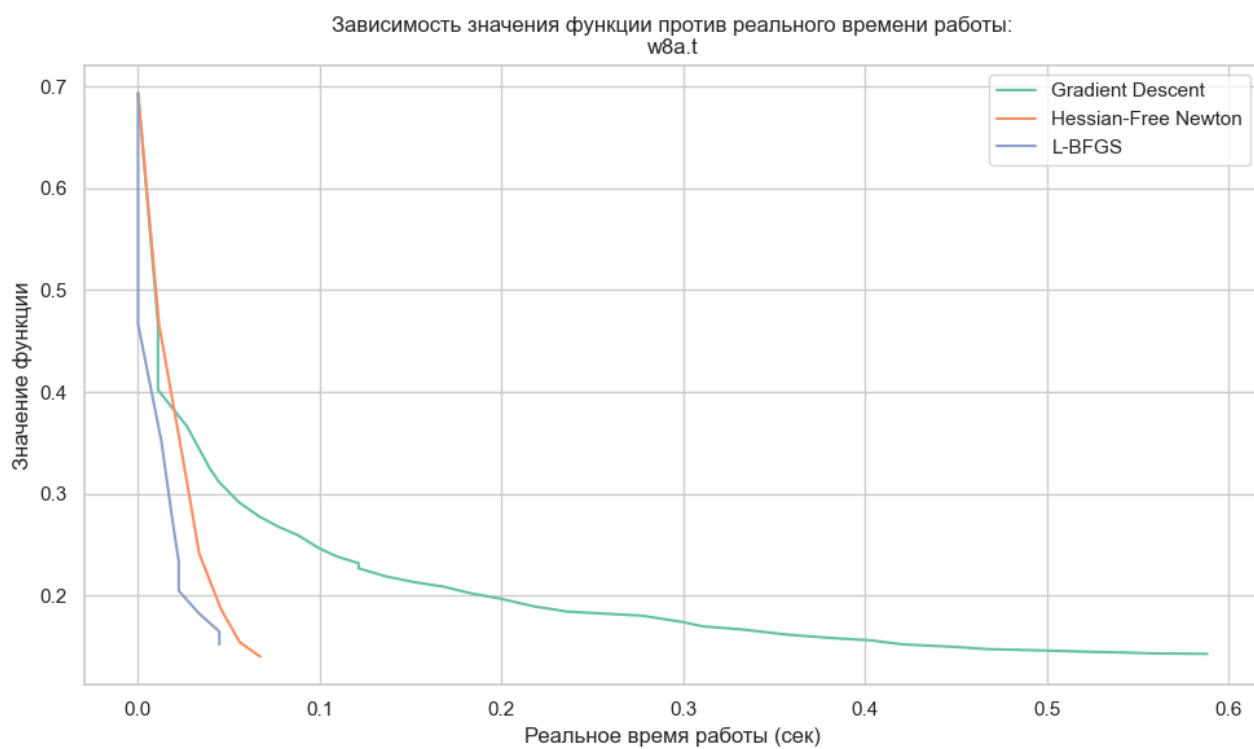


Рис. 10: w8a

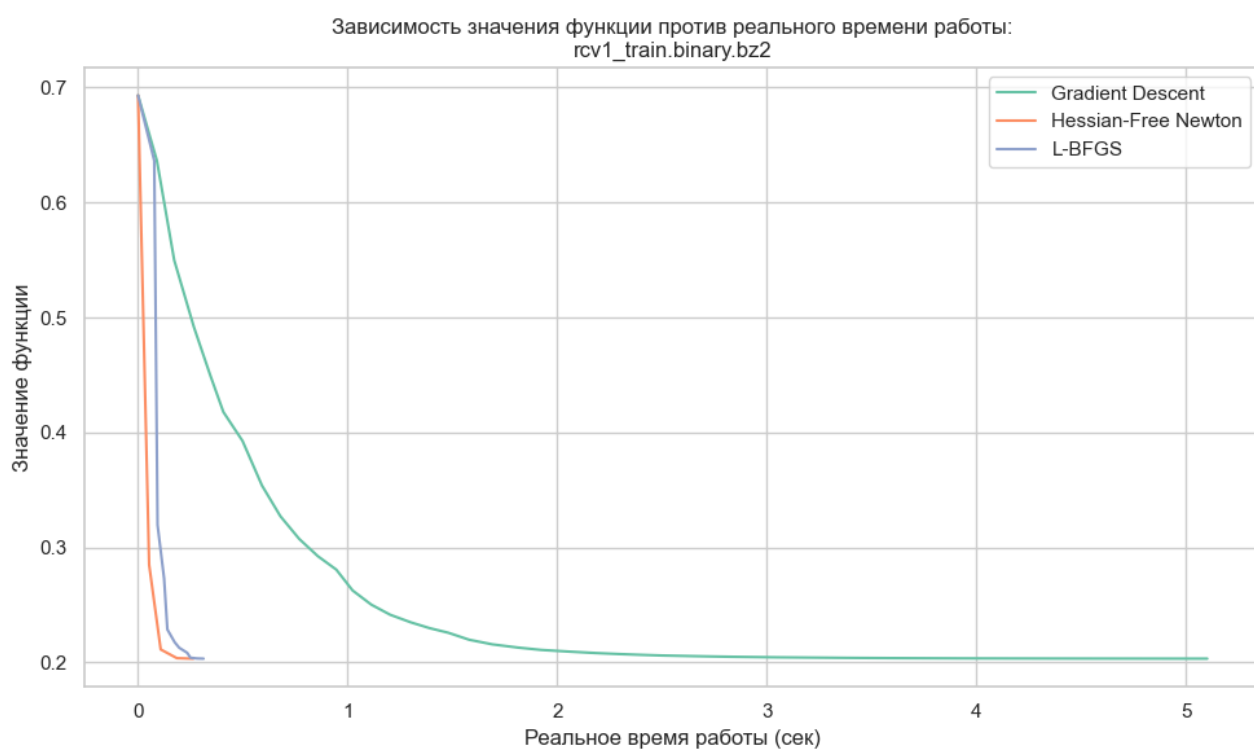


Рис. 11: rcv1\_train.binary

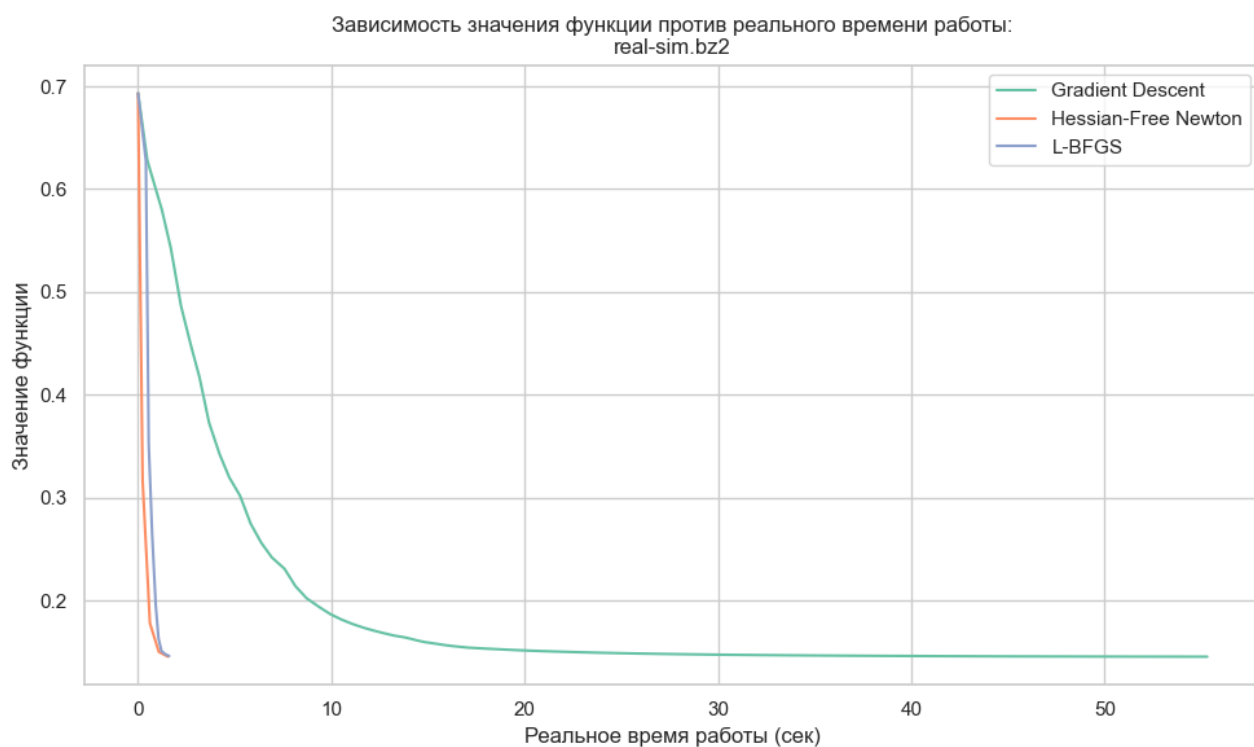


Рис. 12: real-sim.bz2

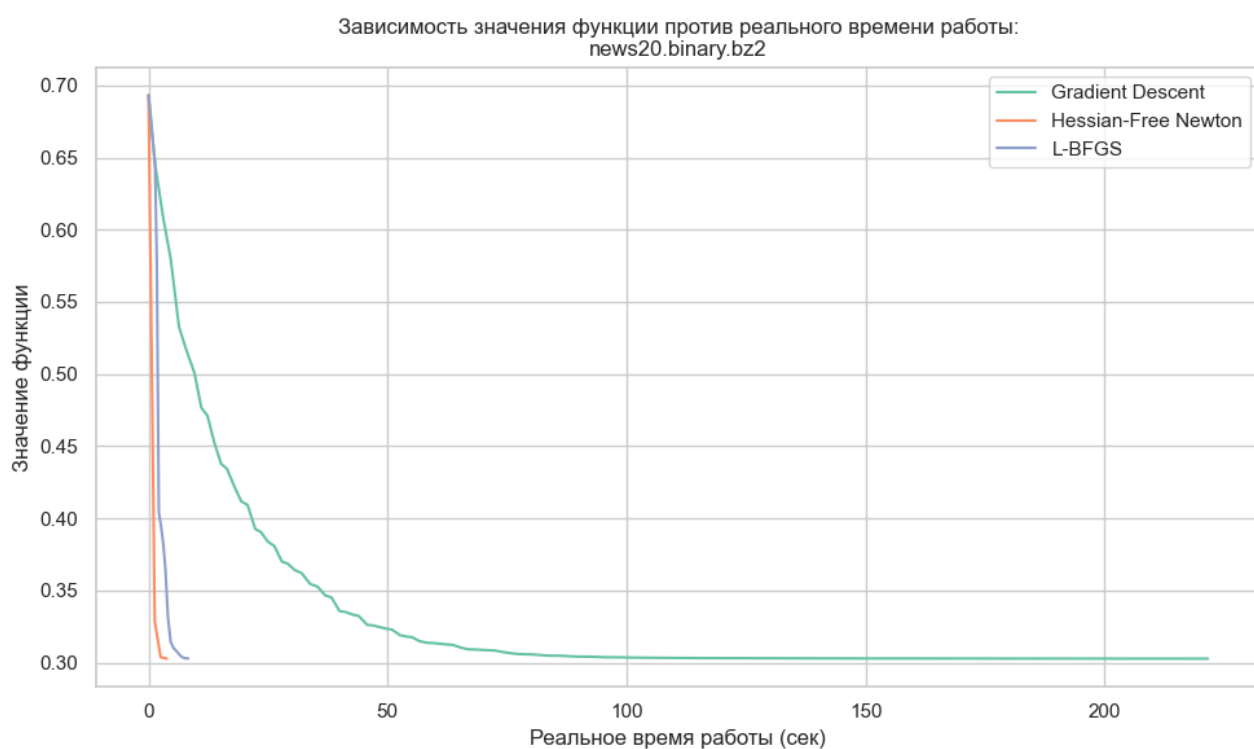


Рис. 13: news20.binary.bz2

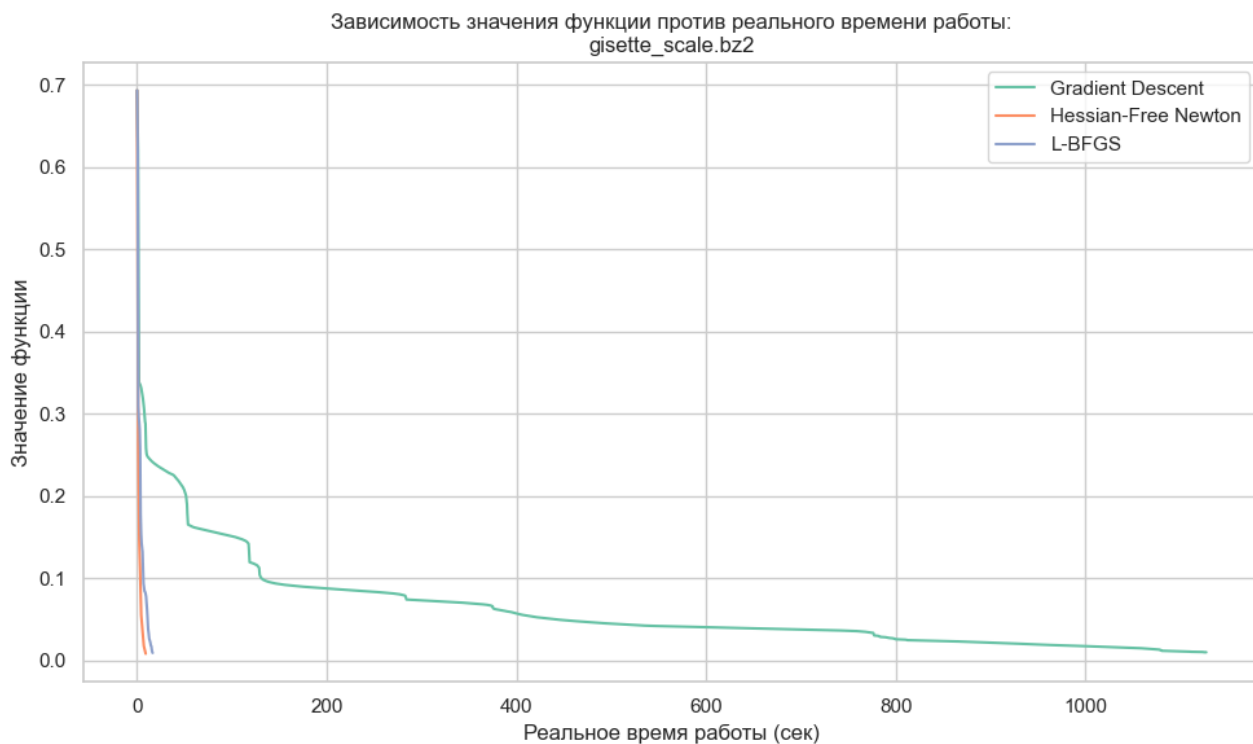


Рис. 14: gisette\_scale.bz2

**Комментарий.** По времени работы HFN оказывается самым быстрым на большинстве датасетов, кроме `w8a`, где быстрее L-BFGS. Градиентный спуск значительно уступает обоим методам, особенно на больших по размеру датасетах. Наиболее тяжёлым оказывается `gisette_scale` — на нём все методы работают на порядок дольше, чем на остальных, что подтверждает влияние количества ненулевых значений на вычислительные затраты.

# Зависимость относительного квадрата нормы градиента от реального времени работы

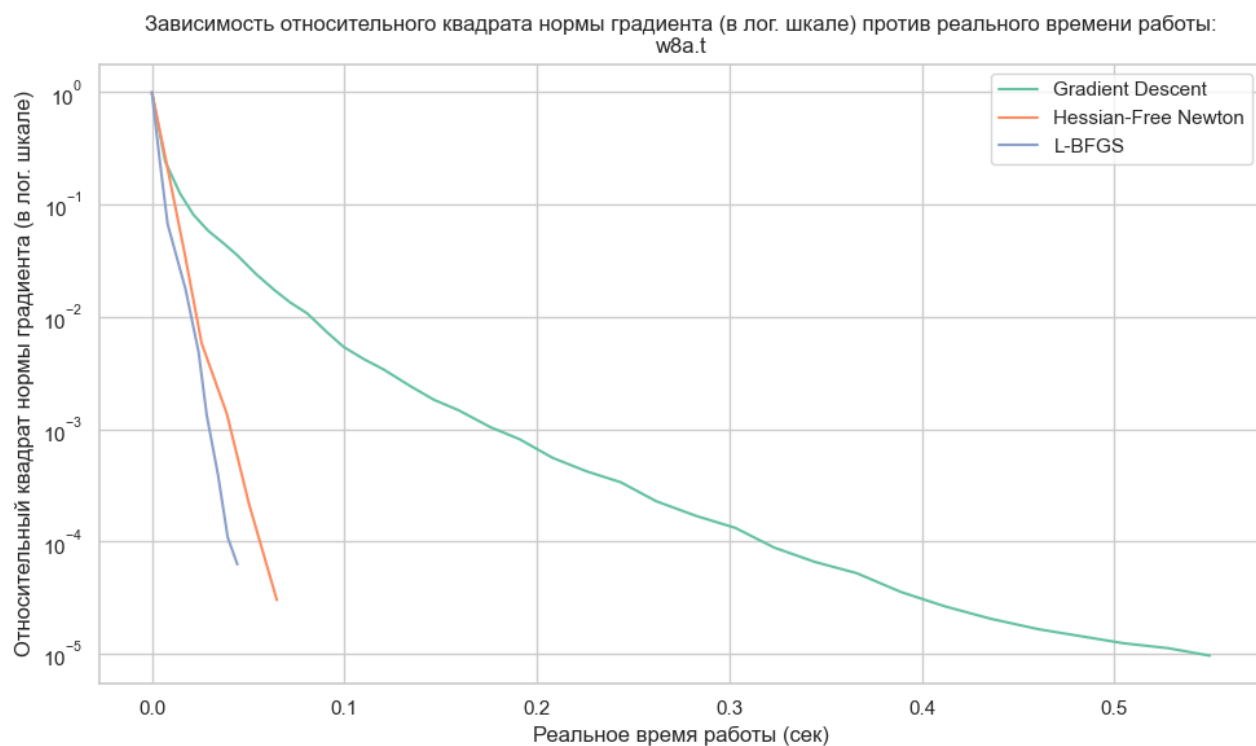


Рис. 15: w8a

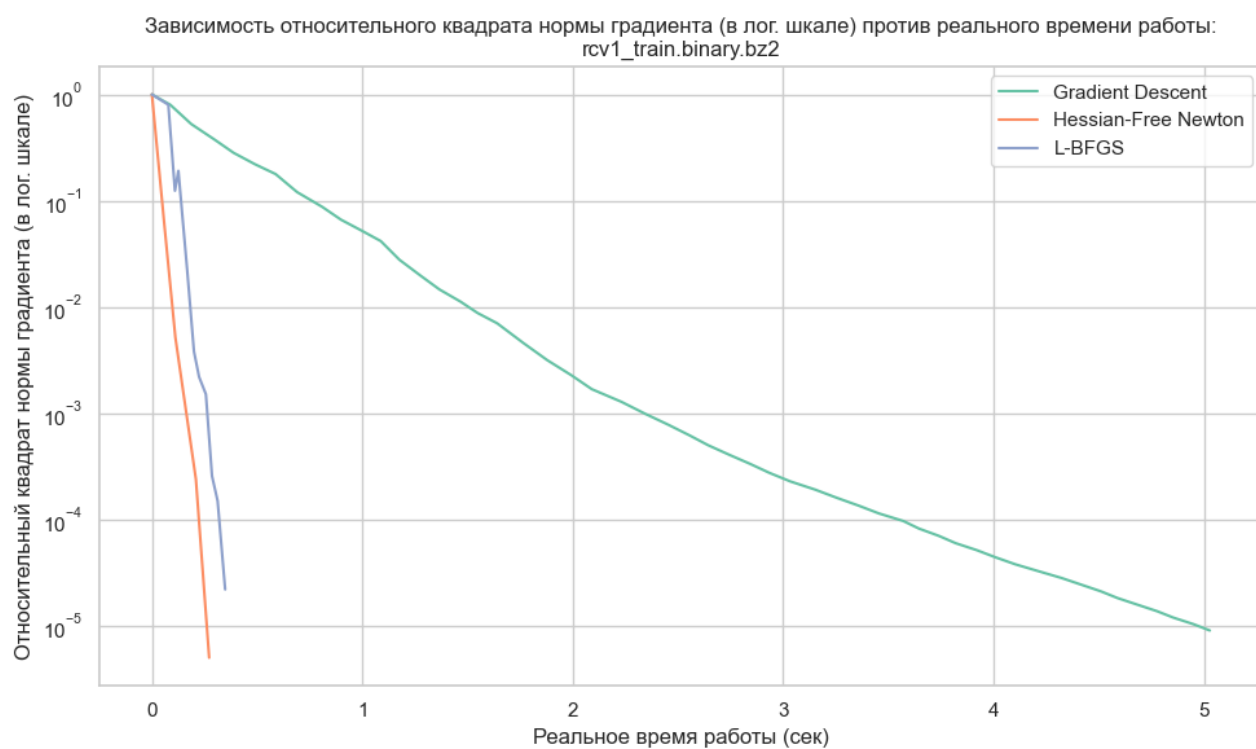


Рис. 16: rcv1\_train.binary



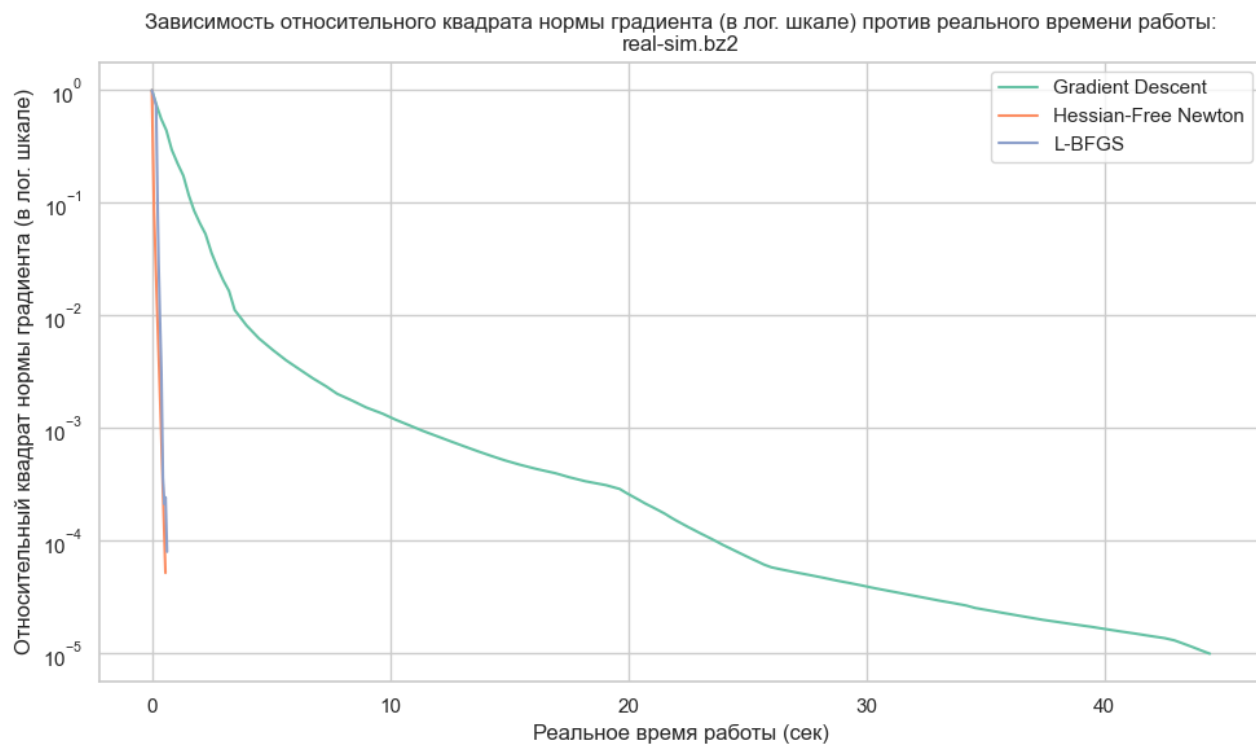


Рис. 17: real-sim.bz2

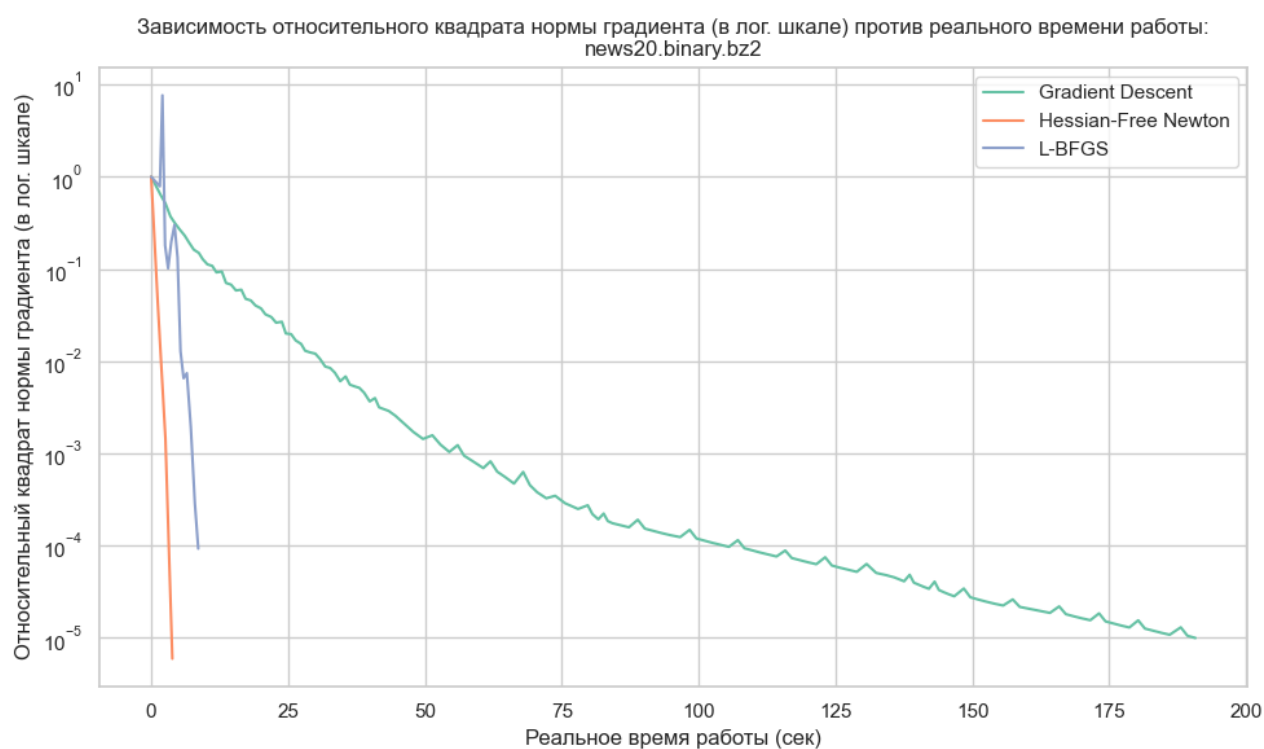


Рис. 18: news20.binary.bz2

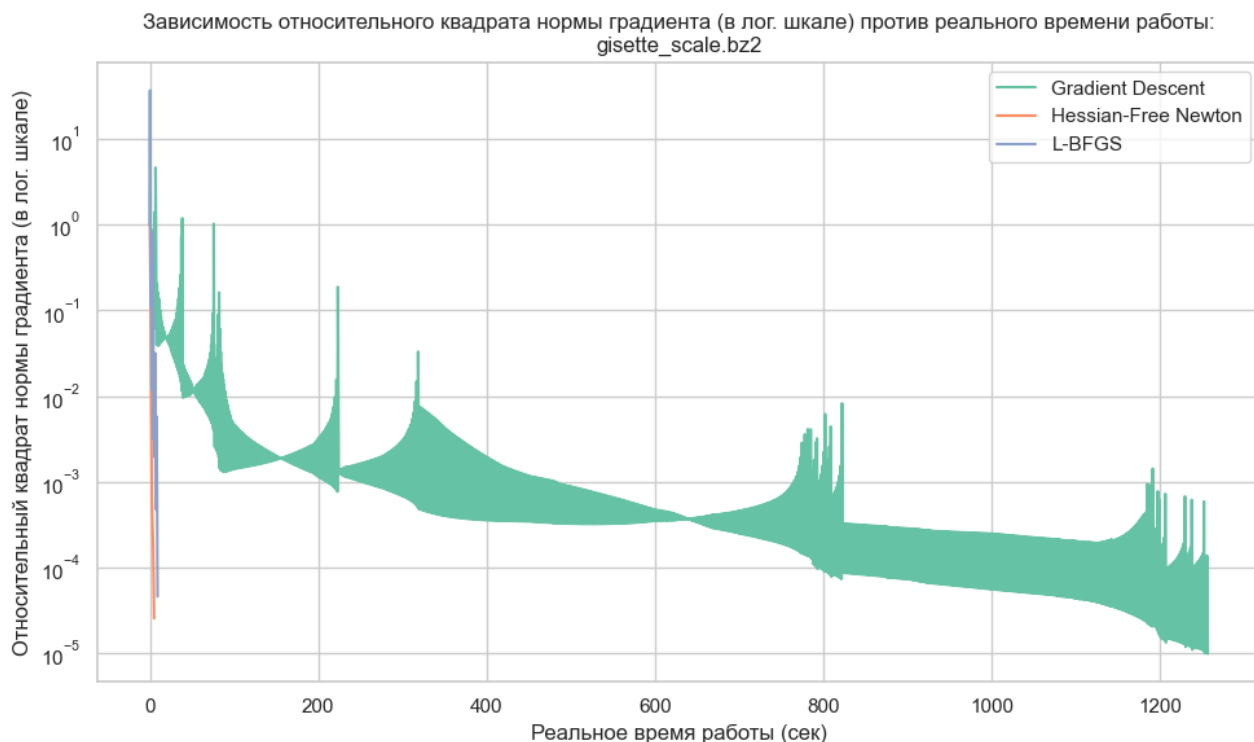


Рис. 19: gisette\_scale.bz2

**Комментарий.** HFN демонстрирует самую быструю сходимость по норме градиента на датасетах `news20.binary`, `gisette_scale`, `rcv1_train.binary`. На `realsim` он сравним с L-BFGS, а на `w8a` немного уступает ему. Градиентный спуск показывает практически линейное (в логарифмическом масштабе) и устойчивое поведение на всех датасетах, кроме `gisette_scale`, где наблюдается значительный шум и нестабильность.

## Общий вывод

HFN показывает наилучшие результаты по числу итераций и времени сходимости на большинстве датасетов. Он демонстрирует устойчивое и быстрое поведение как по значению функции, так и по норме градиента. L-BFGS также показывает высокую эффективность и в ряде случаев обгоняет HFN на малых задачах. Градиентный спуск, напротив, требует существенно больше итераций и времени, особенно при увеличении размерности и числа ненулевых элементов.

Однако, усечённый метод Ньютона требует вычисления действия гессиана на вектор, что может быть вычислительно дорого на очень больших задачах. Кроме того, HFN требует больше памяти, чем L-BFGS и особенно градиентный спуск. L-BFGS, напротив, использует ограниченное количество памяти и не требует вычислений, связанных с гессианом. Таким образом, при выборе метода необходимо учитывать компромисс между числом итераций, временем, используемой памятью и сложностью вычислений. HFN — лучший выбор при наличии достаточных вычислительных ресурсов; L-BFGS — более практичен; градиентный спуск — базовые, но медленный вариант.

## 4 Эксперимент 2.4. Сравнение метода сопряженных градиентов и L-BFGS на квадратичной функции

**Описание эксперимента.** Целью данного эксперимента является изучение зависимости скорости сходимости методов L-BFGS и сопряжённых градиентов в терминах евклидовой нормы невязки от номера итерации на примере двух случайных строго выпуклых квадратичных задач размерностей  $n = 30$  и  $n = 300$  при различных размерах памяти для L-BFGS. В каждой задаче оба метода стартуют из одной и той же точки  $x_0 = 0$  и применяют точный линейный поиск. На одном и том же графике строится зависимость евклидовой нормы невязки

$$r_k = Ax_k - b, \quad \|r_k\|_2$$

в логарифмическом масштабе от номера итерации: метод сопряжённых градиентов отображается сплошной чёрной линией, L-BFGS — пунктиром для трёх значений памяти  $l \in \{0, 1, 5\}$ .

**Аналитическое выражение для оптимального шага  $\alpha_k$  в квадратичной функции:**

$$\alpha_k = \frac{-\nabla f(x_k)^\top d_k}{d_k^\top A d_k} = \frac{(b - Ax_k)^\top d_k}{d_k^\top A d_k}.$$

**Основные параметры эксперимента:**

- Генерация данных с фиксированным `seed = 42` для воспроизводимости результатов;
- **Параметры для функции с  $n=30$ :**
  - Метод сопряжённых градиентов: `max_iter = 300`, `tolerance = 10-6`;
  - Метод L-BFGS: память  $l \in \{0, 1, 5\}$ , `max_iter = 300`, `tolerance = 10-6`, точный линейный поиск.
- **Параметры для функции с  $n=300$ :**
  - Метод сопряжённых градиентов: `max_iter = 1000`, `tolerance = 10-6`;
  - Метод L-BFGS: память  $l \in \{0, 1, 5\}$ , `max_iter = 1000`, `tolerance = 10-6`, точный линейный поиск.

## Результаты эксперимента

На рисунках 20 и 21 представлены зависимости евклидовой нормы невязки  $r_k = Ax_k - b$  от номера итерации для методов L-BFGS и сопряжённых градиентов при различных значениях параметра истории  $l$  и размерностях задачи  $n = 30$  и  $n = 300$ .

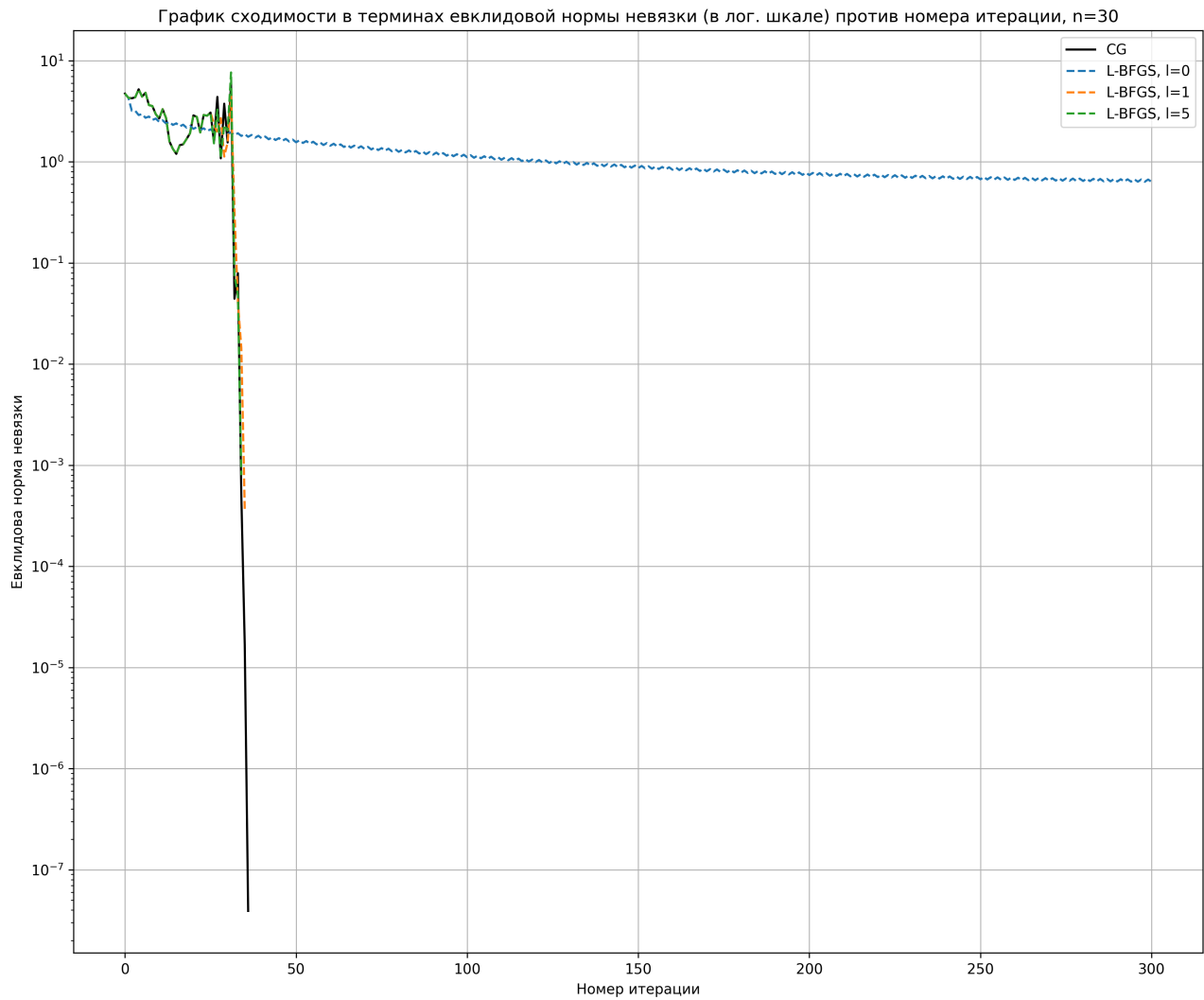


Рис. 20: Сходимость методов при  $n = 30$  (логарифмическая шкала): L-BFGS с  $l = 0, 1, 5$  (сплошные линии) и метод сопряжённых градиентов CG (пунктир).

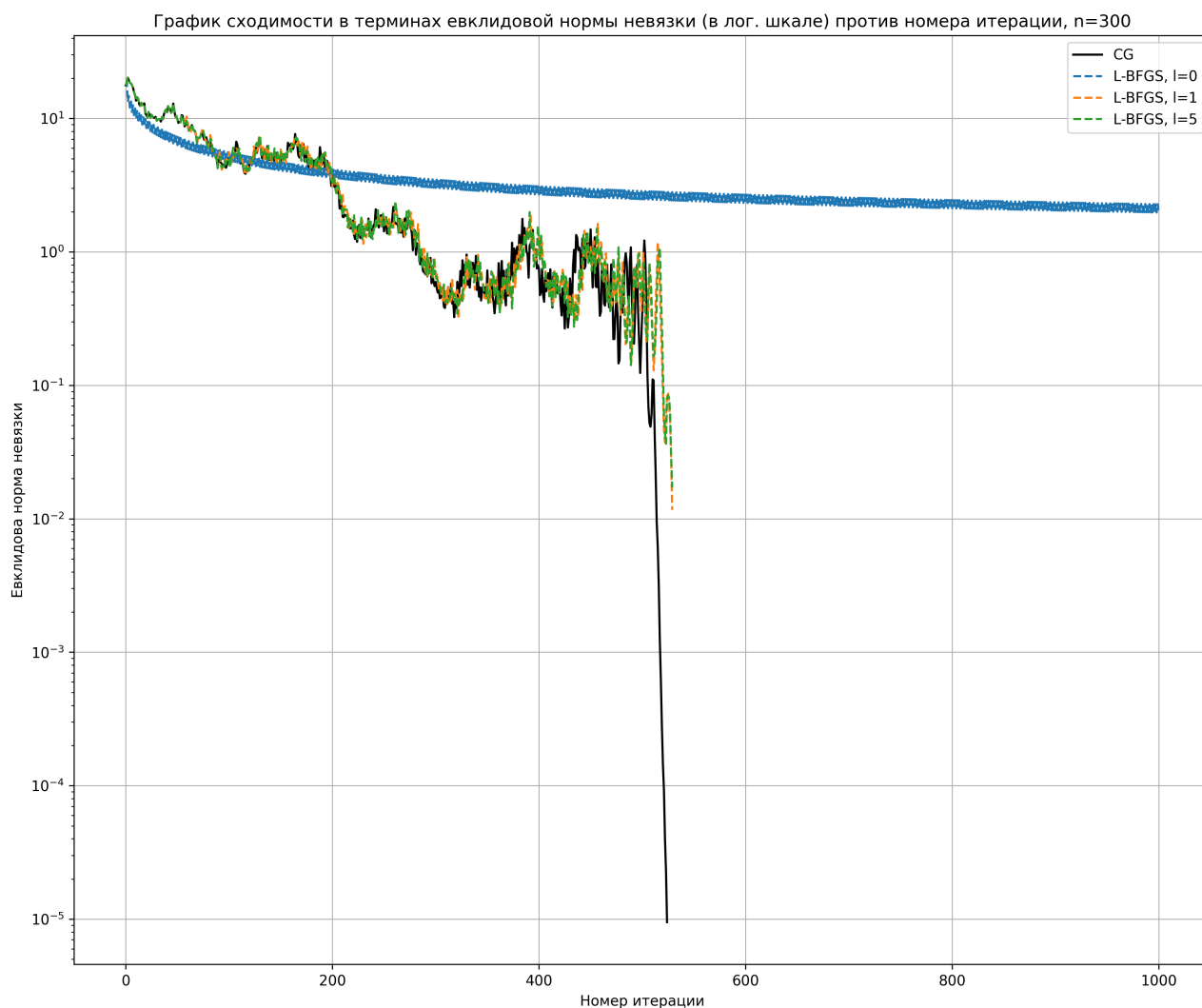


Рис. 21: Сходимость методов при  $n = 300$  (логарифмическая шкала): L-BFGS с  $l = 0, 1, 5$  (сплошные линии) и метод сопряжённых градиентов CG (пунктир).

**Проанализировав полученные графики можно сделать следующие выводы:**

1. При  $n = 30$  (рис.20) L-BFGS с  $l = 0$  демонстрирует поведение, близкое к градиентному спуску: значение невязки падает очень медленно, оставаясь порядка  $10^0$  даже после 300 итераций. Напротив, при  $l = 1$  и  $l = 5$  поведение L-BFGS практически совпадает с CG, а метод CG достигает нормы невязки порядка  $10^{-7}$  менее чем за 40 итераций, что значительно превосходит остальные.
2. При  $n = 300$  (рис.21) L-BFGS с  $l = 0$  по-прежнему медленный, его невязка порядка  $10^0$  сохраняется до 1000 итераций. Для  $l = 1$  и  $l = 5$  L-BFGS и CG снова показывает близкие кривые: они менее плавные в отличие от  $l = 0$  L-BFGS, но достигают минимального возможного значения при заданных параметрах. CG при этом достигает евклидову норму невязки порядка  $10^{-5}$  за примерно 520 итераций.

**Вывод:** метод сопряжённых градиентов CG превосходит L-BFGS с малыми размерами истории ( $l = 1$  и  $l = 5$ ) незначительно по скорости сходимости и значительно по точности на обеих тестовых задачах. При  $l = 0$  L-BFGS сводится к классическому градиентному спуску и демонстрирует значительно более медленную сходимость.

## 5 Эксперимент 2.5. Какая точность оптимизации нужна в реальных задачах?

### Описание эксперимента

Целью данного эксперимента является исследование влияния параметра точности оптимизации  $\varepsilon$  метода L-BFGS на итоговое качество решения задачи бинарной классификации с моделью логистической регрессии и  $L_2$ -регуляризацией. В качестве метрики качества используется процент ошибок классификации на тестовой выборке. В эксперименте применяются девять значений  $\varepsilon$  на логарифмической сетке от  $10^0$  (отсутствие оптимизации, возвращение начальной точки) до  $10^{-8}$  (оптимизация до машинной точности). Коэффициент регуляризации фиксируется как  $\lambda = 1/m$ , начальная точка  $x_0 = 0$ . Для воспроизводимости и сравнения используются пять реальных наборов данных, представленных в формате SVM-light.

### Основные параметры эксперимента:

- Метод оптимизации: L-BFGS.
- Модель: логистическая регрессия с  $L_2$ -регуляризацией.
- Доля данных: 80% — обучение, 20% — тест.
- Значения точности  $\varepsilon$ :

$$\{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$$

- Остальные параметры функции — по умолчанию.

### Характеристики датасетов:

- `w8a.t`:
  - Количество объектов:  $n = 14\,951$
  - Число признаков:  $d = 300$
  - Количество ненулевых значений:  $\text{nnz} = 174\,276$
- `rcv1_train.binary.bz2`:
  - Количество объектов:  $n = 20\,242$
  - Число признаков:  $d = 47\,236$
  - Количество ненулевых значений:  $\text{nnz} = 1\,498\,952$
- `real-sim.bz2`:
  - Количество объектов:  $n = 72\,309$
  - Число признаков:  $d = 20\,958$
  - Количество ненулевых значений:  $\text{nnz} = 3\,709\,083$
- `news20.binary.bz2`:
  - Количество объектов:  $n = 19\,996$

- Число признаков:  $d = 1\,355\,191$
- Количество ненулевых значений:  $\text{nnz} = 9\,097\,916$
- `gisette_scale.bz2`:
  - Количество объектов:  $n = 6\,000$
  - Число признаков:  $d = 5\,000$
  - Количество ненулевых значений:  $\text{nnz} = 29\,729\,997$

## Результаты эксперимента

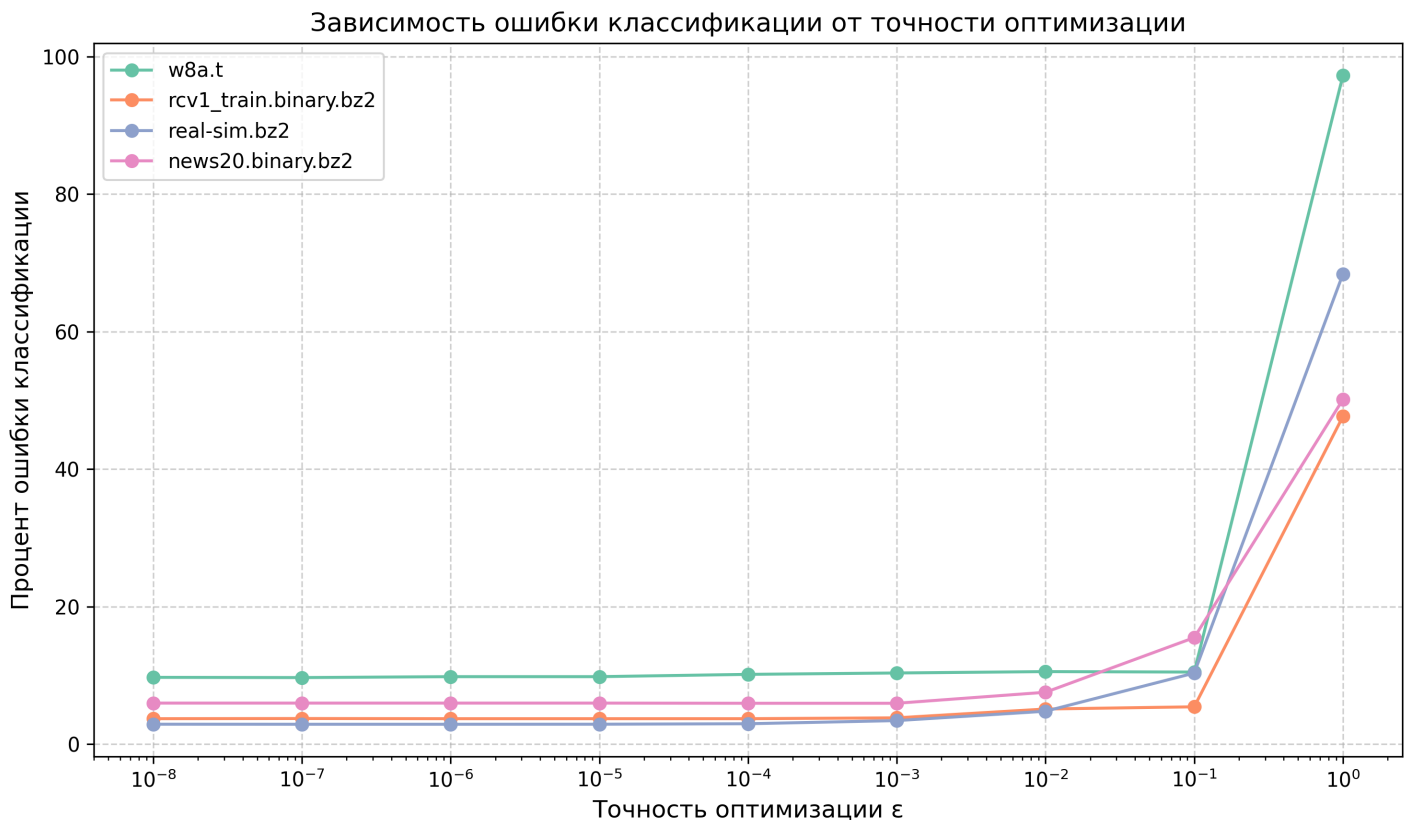


Рис. 22: Зависимость процента ошибок классификации от точности оптимизации  $\varepsilon$  для различных датасетов.

## Анализ результатов

- **Поведение при  $\varepsilon = 10^0$ :** для всех пяти наборов данных процент ошибок довольно высокий, поскольку метод возвращает начальную нулевую точку и классификатор не обучается.
  - `w8a.t`: 97.26%
  - `rcv1`: 47.67%
  - `real-sim`: 68.38%
  - `news20`: 50.12%
- **Резкое улучшение при  $\varepsilon = 10^{-1}$ :** уже при втором уровне точности наблюдается значительное падение ошибки:

- w8a.t: 10.46%
- rcv1: 5.41%
- real-sim: 10.32%
- news20: 15.47%

- **Стабилизация качества после  $\varepsilon = 10^{-4}$ :** дальнейшее уменьшение  $\varepsilon$  даёт лишь незначительный прирост:

- w8a.t: около 9.8% (минимум 9.66% при  $10^{-7}$ )
- rcv1: около 3.7% (фиксируется с  $10^{-4}$ )
- real-sim: около 2.9% (минимум 2.87% при  $10^{-6}$ – $10^{-7}$ )
- news20: около 5.9% (фиксируется с  $10^{-3}$ )

- **Влияние на время работы:** вычислительное время растёт почти экспоненциально при снижении  $\varepsilon$ , тогда как снижение ошибки после  $10^{-4}$  становится незначительным.

## Выводы

1. Для практических задач бинарной классификации оптимально выбирать точность оптимизации в диапазоне  $\varepsilon \in [10^{-2}, 10^{-4}]$ : это даёт почти максимальную точность при умеренных затратах времени.
2. Дальнейшее уменьшение  $\varepsilon$  до уровня машинной точности не оправдано: выигрыш в качестве незначителен, а время работы увеличивается в несколько раз.
3. Метод L-BFGS сохраняет устойчивость и эффективность при разумном выборе  $\varepsilon$ , показывая быстрое снижение ошибки на первых шагах и стабильную сходимость далее.