Лабораторная работа 1: Методы градиентного спуска и метод Ньютона

Белова Юлия

20 апреля 2023 г.

Содержание

1 Логистическая регрессия

Введем обозначения: $X \in \mathbb{R}^{m \times n}$, $\overrightarrow{y} \in \mathbb{R}^{m \times 1}$, $\overrightarrow{w} \in \mathbb{R}^{n \times 1}$.
Выражение для функции логистической регрессии:

$$L(\overrightarrow{w}) = \frac{1}{M}\|\ln(1_m + \exp(-\overrightarrow{y} \odot X\overrightarrow{w}))\|_1 + \frac{\lambda}{2}\|\overrightarrow{w}\|_2^2$$

Выражение для градиента логистической регрессии:

$$\nabla_{\overrightarrow{w}} L(\overrightarrow{w}) = \lambda\overrightarrow{w} - \frac{1}{M}X^T\left[\overrightarrow{y} \odot \sigma\left(-\overrightarrow{y} \odot X\overrightarrow{w}\right)\right]$$

Выражение для гессиана логистической регрессии:

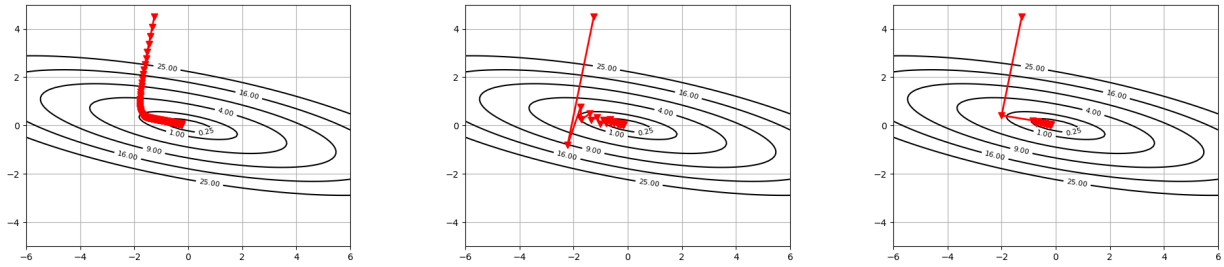
$$H(\overrightarrow{w}) = \lambda I_n + X^T\left[\sigma\left(\overrightarrow{y} \odot X\overrightarrow{w}\right)\left(1_m - \sigma\left(\overrightarrow{y} \odot X\overrightarrow{w}\right)\right)\right]X$$

2 Эксперимент 1: Траектория градиентного спуска на квадратичной функции

Для проведения эксперимента были выбраны две квадратичные функции с разным по степени числом обусловленности: 18.11 и 2.99. Также были сгенерированы три начальные точки. Для каждой функции перебирались начальные точки и стратегии выбора шага (константная стратегия, Армихо, Вульф).

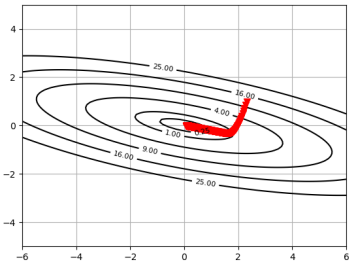
2.1 Большое число обусловленности

Рассмотрим результаты эксперимента для каждой из начальных точек:

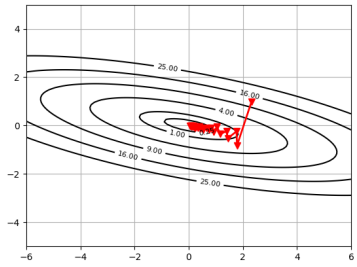


(a) Постоянный шаг: 384 шага (b) Армихо: 23 шага (c) Вульф: 15 шагов

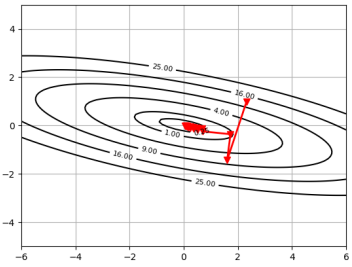
Рис. 1: Начальная точка 1



(a) Постоянный шаг: 558 шагов

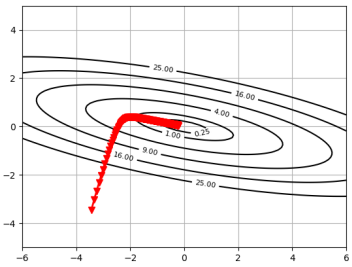


(b) Армихо: 30 шагов

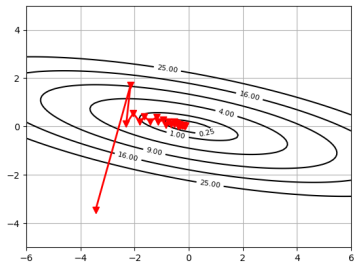


(c) Вульф: 19 шагов

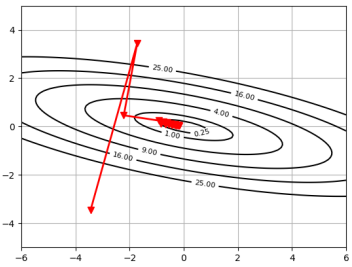
Рис. 2: Начальная точка 2



(a) Постоянный шаг: 422 шага



(b) Армихо: 26 шагов

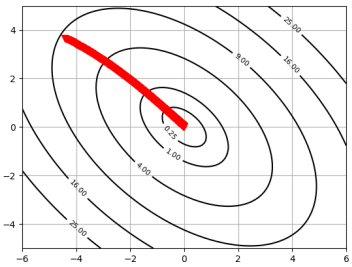


(c) Вульф: 20 шагов

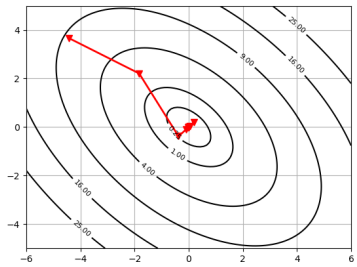
Рис. 3: Начальная точка 3

Для функции с большим числом обусловленности градиентный спуск хуже всего работает при постоянном шаге. Метод Вульфа сходится к оптимуму чуть быстрее метода Армихо. Видно, что спуск с использованием метода Армихо движется к оптимуму зигзагами, в то время как Вульф идет напрямую.

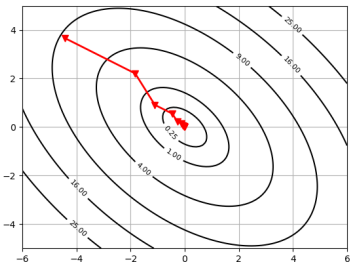
2.2 Маленькое число обусловленности



(a) Постоянный шаг: 1142 шага

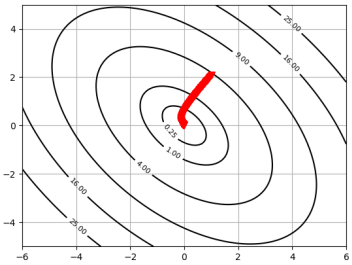


(b) Армихо: 10 шагов

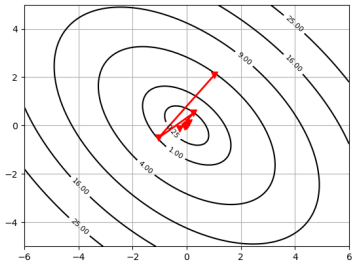


(c) Вульф: 10 шагов

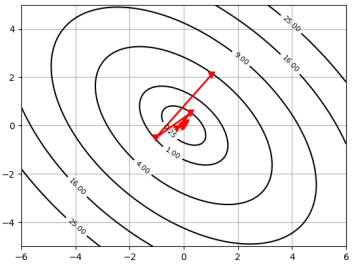
Рис. 4: Начальная точка 1



(a) Постоянный шаг: 718 шагов

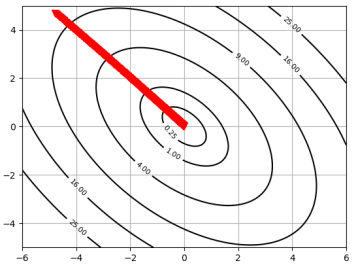


(b) Армихо: 10 шагов

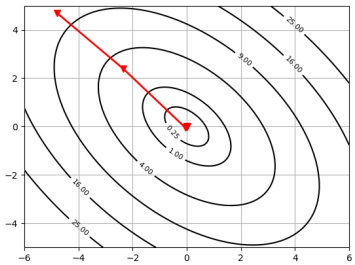


(c) Вульф: 10 шагов

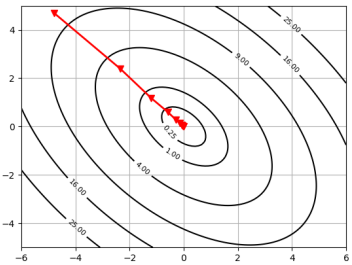
Рис. 5: Начальная точка 2



(a) Постоянный шаг: 1150 шагов



(b) Армихо: 7 шагов



(c) Вульф: 10 шагов

Рис. 6: Начальная точка 3

Несмотря на низкую обусловленность матрицы, постоянный шаг дает плохие результаты: методу нужно сделать очень много шагов, в то время как Армихо и Вульф очень быстро справляются с задачей. Таким образом, чем выше обусловленность матрицы, тем дольше работает градиентный спуск.

3 Эксперимент 2: Зависимость числа итераций градиентного спуска от числа обусловленности и размерности пространства

В данном эксперименте рассматривается квадратичная задача. Исследуем, сколько итераций необходимо совершить методу градиентного спуска до сходимости в зависимости от числа обусловленности матрицы и количества оптимизируемых переменных. Для проведения эксперимента были выбраны следующие параметры:

1. Размерность пространства n: $[10, 10^2, 10^3, 10^4, 10^5]$
2. Число обусловленности матрицы k: числа в интервале $[1, 1000]$ с шагом 50

Для каждой размерности пространства n задача генерировалась случайным образом 5 раз. В градиентном спуске использовались параметры по умолчанию:

- Начальная точка $x_0 = 0$
- Метод подбора шага - алгоритм Вульфа с параметрами $c_1 = 1\text{e}{-4}$, $c_2 = 0.9$

На рис. 7 отображены результаты эксперимента: ярким цветом выделены средние результаты (по случайным генерациям) для каждого n.

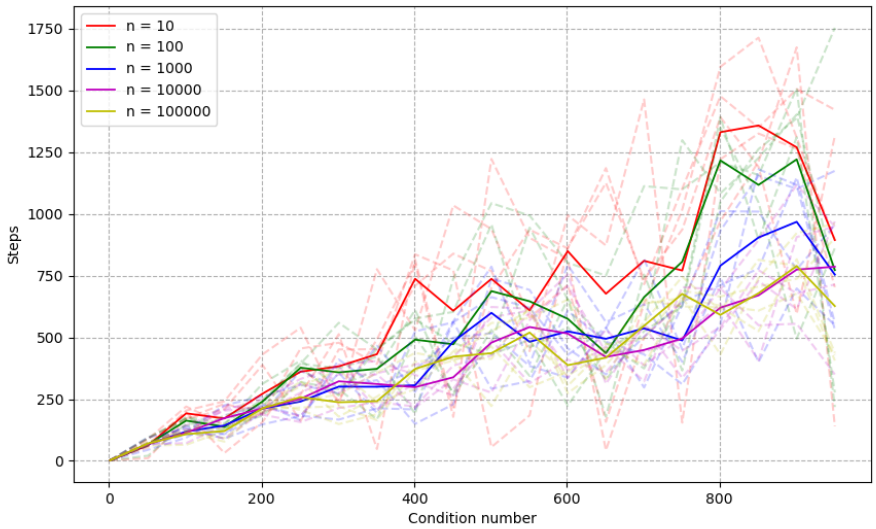


Рис. 7: Зависимость значения функции от реального времени работы метода

По результатам эксперимента можно сделать следующий вывод: для градиентного спуска наблюдается линейная зависимость числа итераций от обусловленности матрицы. С увеличением числа зависимых переменных количество итераций, необходимых для сходимости, не растет. Наоборот, при достаточно высокой обусловленности матрицы градиентный спуск работает быстрее на матрицах с большим числом признаков.

4 Эксперимент 3: Сравнение методов градиентного спуска и Ньютона на реальной задаче логистической регрессии

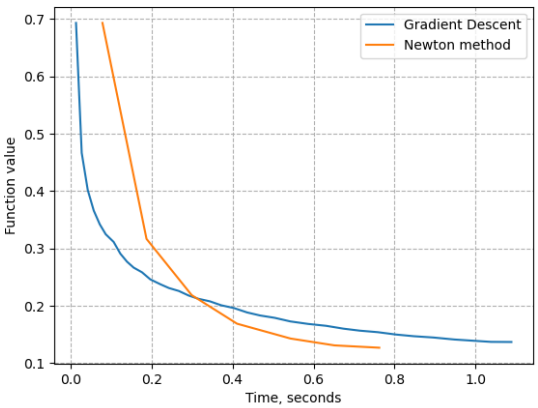
Для сравнения методов градиентного спуска и Ньютона на задаче обучения логистической регрессии были использованы три датасета. Их параметры представлены ниже:

1. *w8a*: количество наблюдений - 49 740, количество признаков - 300.
2. *gisette*: количество наблюдений - 6 000, количество признаков - 5 000. Данный датасет хранится в формате `scipy.sparse.csr_matrix`, однако 99% данных имеют ненулевые значения.
3. *real-sim*: количество наблюдений - 72 309, количество признаков - 20 958. Данный датасет хранится в формате `scipy.sparse.csr_matrix`, процент ненулевых значений в матрице с данными 0.002%.

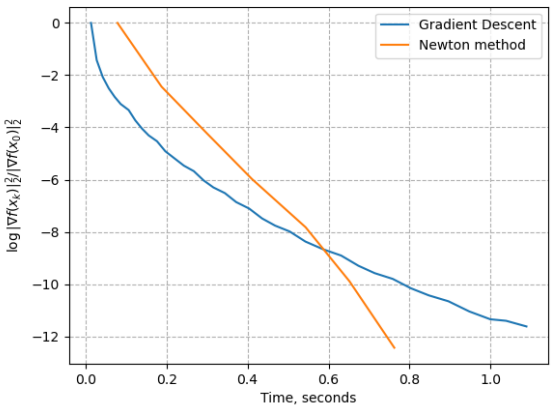
Параметры обоих методов взяты по умолчанию, начальная точка $x_0 = 0$.

Градиентному спуску нужно $\mathcal{O}(n)$ памяти на хранение значений функции и градиента, стоимость итерации $\mathcal{O}(q) + \mathcal{O}(n)$, которая складывается из времени работы при обращении к оракулу и линейного поиска шага. Метод Ньютона требует $\mathcal{O}(n^2)$ памяти (все то же самое + гессиан), а стоимость одной итерации составляет $\mathcal{O}(n^3) + \mathcal{O}(qn^2)$, которое складывается из времени работы при обращении к оракулу (значение функции, градиент, гессиан), решения системы линейных уравнений для нахождения направления и времени на поиск оптимального шага.

4.1 w8a



(a) Зависимость значения функции от реального времени работы метода

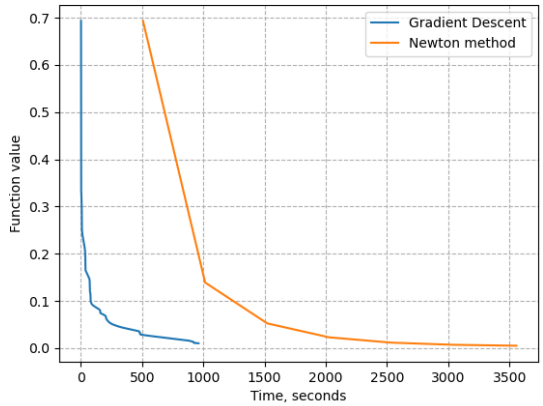


(b) Зависимость относительного квадрата нормы градиента от реального времени работы

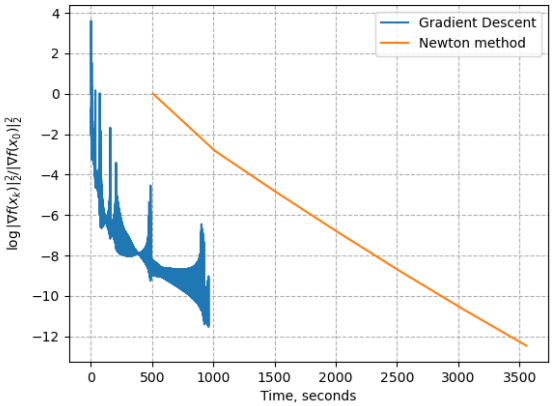
Градиентному спуску потребовалось 36 итераций, а методу Ньютона 7 итераций. Итерация градиентного спуска стоит примерно 0.03 секунд, а метода Ньютона 0.11 секунд. При этом метод Ньютона сошелся чуть быстрее. Из графика зависимости относительного квадрата нормы градиента против реального времени работы можно увидеть, что в

случае работы методы Ньютона логарифм относительной невязки уменьшается быстрее вблизи оптимального решения (следующая значащая цифра получилась быстрее, чем в методе градиентного спуска). Вывод: если нам не нужна высокая точность решения, то используем градиентный спуск, потому что вначале он работает быстрее.

4.2 gisette



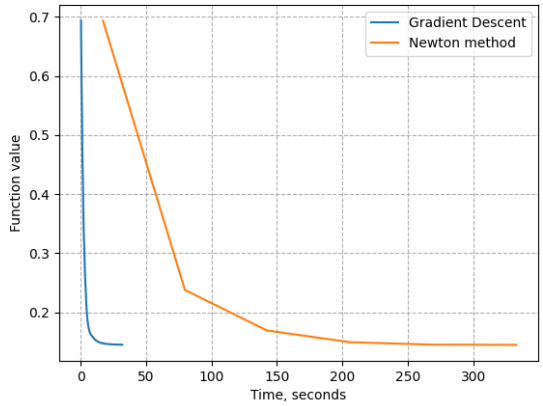
(a) Зависимость значения функции от реального времени работы метода



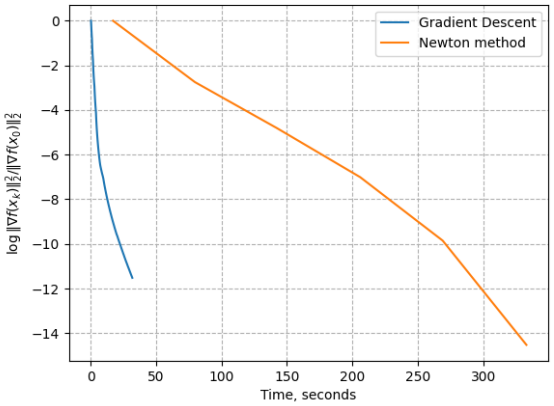
(b) Зависимость относительного квадрата нормы градиента от реального времени работы

Градиентному спуску потребовалось 2004 итерации, методу Ньютона 7 итераций. Итерация градиентного спуска стоит примерно 0.5 секунд, а метода Ньютона 500 секунд. Из-за высокой размерности набора данных и высокой *density* данных метод Ньютона работал очень долго (искал гессиан, решал СЛАУ). Можно сделать вывод, что предпочтительнее использовать метод градиентного спуска в случае задач в пространстве большой размерности: итераций больше, но они значительно дешевле по времени. Метод Ньютона следует использовать для задач небольшой размерности (или на довольно разреженных данных), он будет эффективнее себя показывать ввиду квадратичной сходимости.

4.3 real-sim



(a) Зависимость значения функции от реального времени работы метода



(b) Зависимость относительного квадрата нормы градиента от реального времени работы

Градиентному спуску потребовалось 104 итерации, методу Ньютона 6 итераций. Итерация градиентного спуска стоит примерно 0.4 секунд, а метода Ньютона 57 секунд. Хоть данный датасет самый большой, но матрица с данными очень разреженная, поэтому метод Ньютона отработал сильно быстрее, чем на датасете *gisette*.

Выводы:

- Высокая размерность пространства и (или) высокая плотность данных $\rightarrow$ градиентный спуск.
- Низкая размерность пространства или разреженность данных $\rightarrow$ метод Ньютона.

5 Эксперимент 4: Стратегия выбора длины шага в градиентном спуске

Для проведения данного эксперимента были сгенерированы:

1. Квадратичная задача размерности 500 с числом обусловленности 10 (построение как в третьем эксперименте).
2. Логистическая регрессия, где $n = m = 500$. Для генерации датасета была использована встроенная функция из `sklearn`.

Три типа начальных точек:

1. $x_0 = 0$
2. $x_0 = x^* + N(10, 5)$
3. $x_0 = x^* + N(50, 5)$

Приведем результаты эксперимента для логистической регрессии.

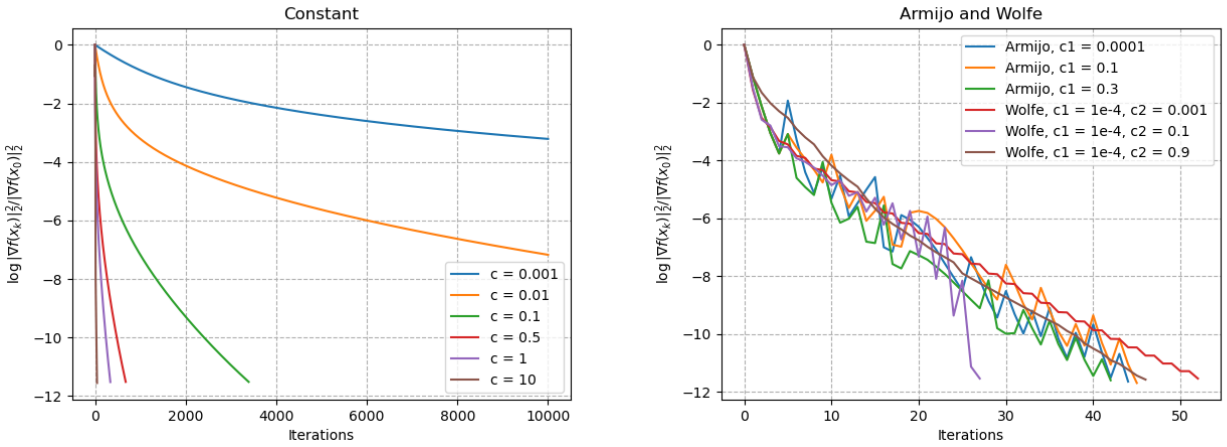


Рис. 11: Начальная точка 1

Из графиков наблюдаем, что градиентный спуск очень чувствителен к выбору постоянного шага: для больших значений шага алгоритм сошелся достаточно быстро, для маленьких не сошелся вообще. Поэтому в данном случае исследователю необходимо самому подбирать шаг для решения задачи оптимизации. Градиентный спуск с выбором шага с помощью методов Армихо и Вульфа сошелся для каждой начальной точки и для каждого перебираемого параметра. Разница между полученными результатами небольшая.

Теперь рассмотрим начальные точки, более удаленные от оптимума.

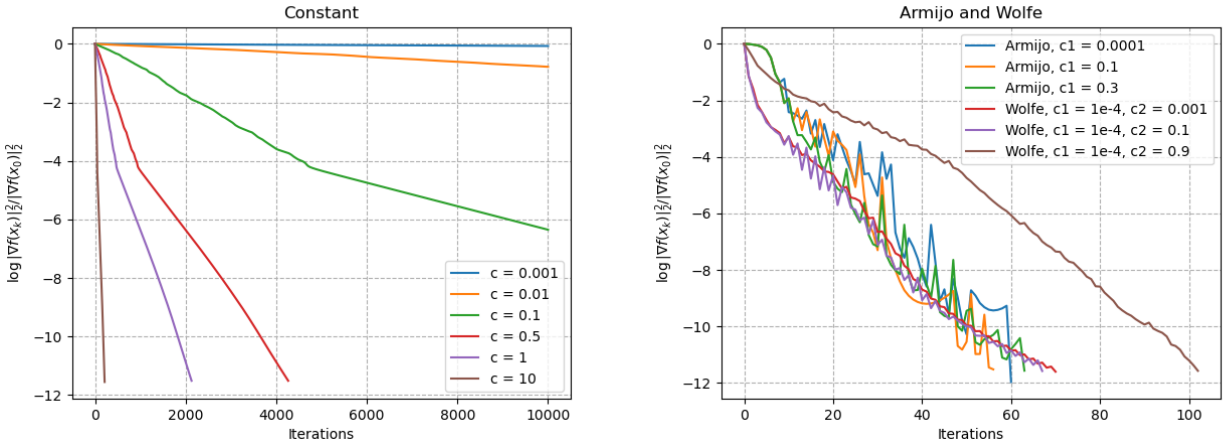


Рис. 12: Начальная точка 2

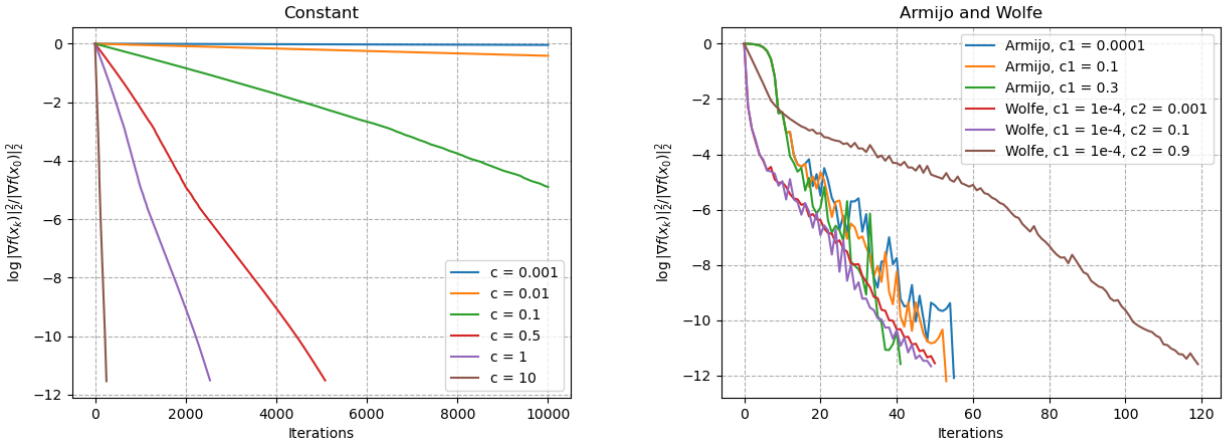


Рис. 13: Начальная точка 1

Для стратегий с постоянным шагом количество итераций до сходимости сильно подскочило для всех констант, кроме $c = 10$. Для $c = 0.1$ метод перестал сходиться. Таким образом, резуюмируя, выбирая стратегию с постоянным шагом, есть высокий риск не сойтись, если не подобрать правильную константу. Для методов Армихо и Вульфа скорость сходимости стала незначительно хуже.

Вывод: для логистической регрессии следует выбирать стратегию подбора шага Армихо или Вульфа, потому что они сами настраивают оптимальный шаг и вероятность не сойтись гораздо ниже, чем у стратегии с постоянным шагом.

Приведем результаты эксперимента для квадратичной функции. Мне не удалось построить графики для начальной точки $x_0 = 0$, потому что, хоть все методы и сошлись, но значения $\log \frac{f(x_k)}{f(x_0)}$ улетают в nan.

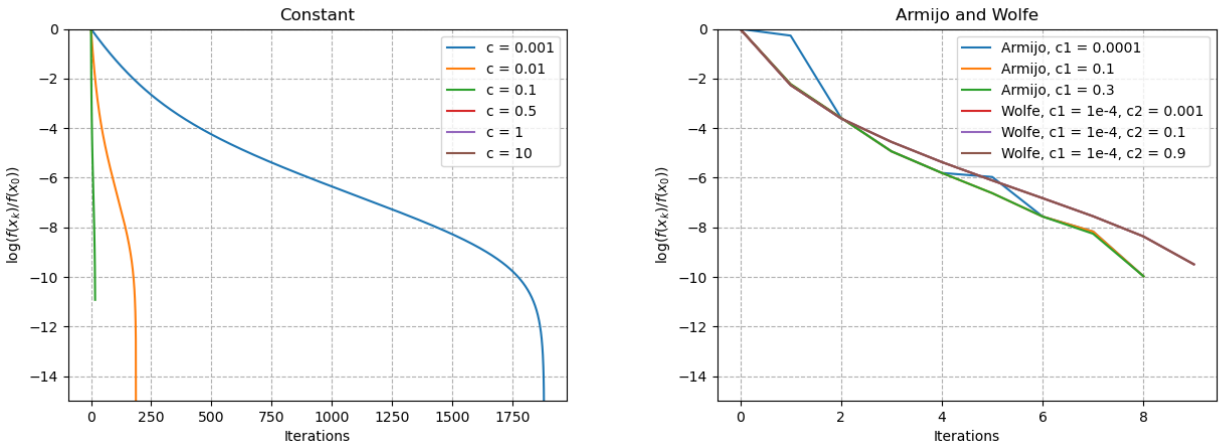


Рис. 14: Начальная точка 2

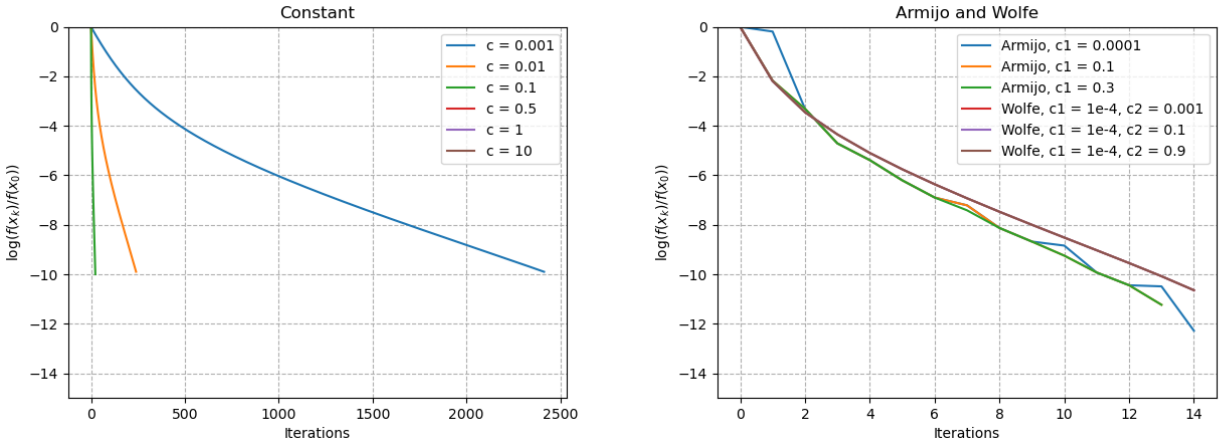


Рис. 15: Начальная точка 3

Для стратегий с выбором константного шага метод сошелся не везде (для $c \in [0.5, 1, 10]$). Для оставшихся констант метод сошелся относительно быстро, но опять же, нужно самостоятельно подбирать константу. Методы Армихо и Вульф сошлись очень быстро, разницы между ними практически не наблюдается.

Вывод: для решения задачи оптимизации квадратичной функции используем стратегии с адаптивным шагом.

6 Эксперимент 5: Стратегия выбора длины шага в методе Ньютона

В данном эксперименте для логистической регрессии была использована модельная выборка из эксперимента 4. Начальные точки и перебираемые параметры те же самые. Результаты эксперимента приведены ниже.

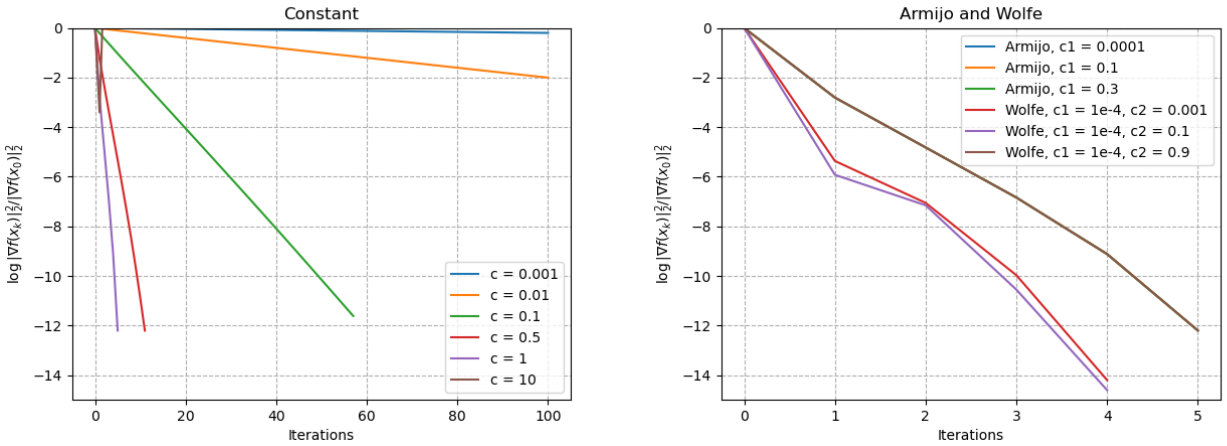


Рис. 16: Начальная точка 1

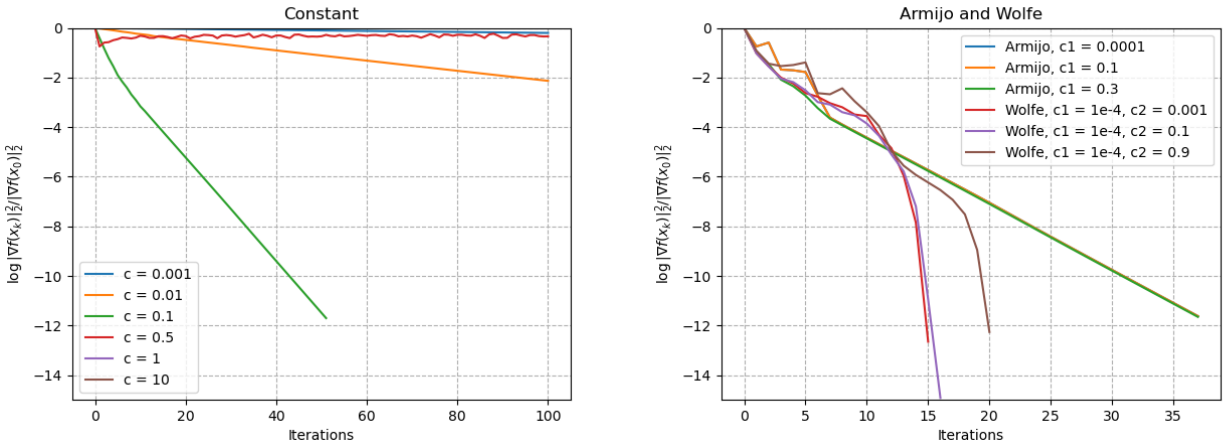


Рис. 17: Начальная точка 2

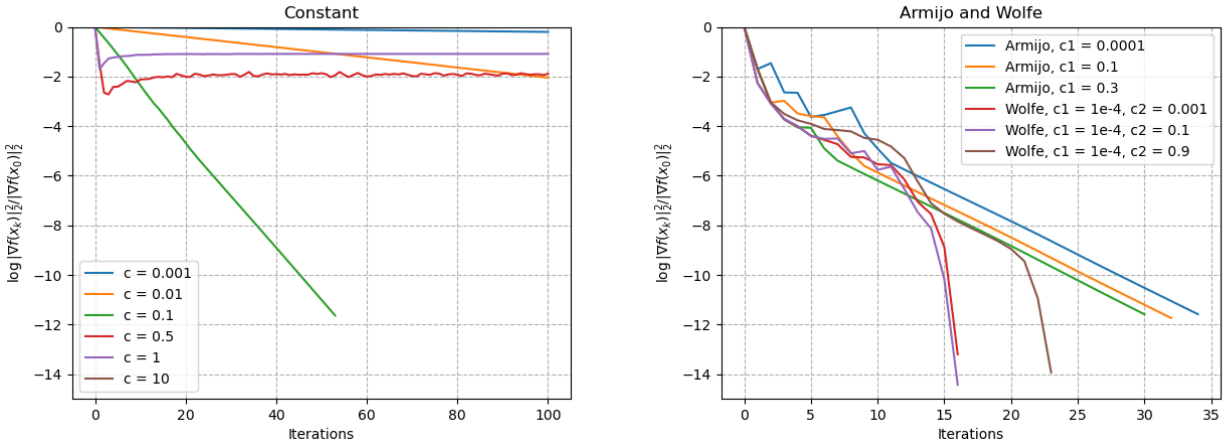


Рис. 18: Начальная точка 3

Для постоянного шага наблюдается расходимость при некоторых c, при $c = 1$ градиентный спуск сходится всего за 6 итераций. Результаты для стратегий Армихо и Вульфа практически одинаковые. По мере отдаления от оптимальной точки методу Ньютона требуется больше итераций. Лучше всего себя показал метод Вульфа с константами 0.001 и 0.01 - сходимость квадратичная.

Вывод: при использовании метода Ньютона для оптимизации функции логистической регрессии лучше всего работает стратегия подбора шага, основанная на условиях Вульфа. Константный шаг $c = 1$ хорошо себя показал вблизи оптимальной точки, однако при отдалении от нее метод не сошелся.