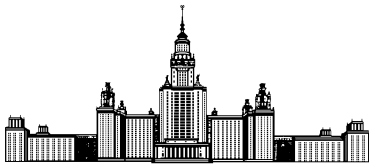
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной математики и Кибернетики
Кафедра Математических Методов Прогнозирования

Методы оптимизации в машинном обучении

Отчет по практическому заданию №1.

«Методы градиентного спуска и Ньютона»

Выполнил:
магистр 1 курса 517 группы
Таскынов Ануар

Москва, 2017

Содержание.

1 Постановка задания

В данном задании необходимо было реализовать методы градиентного спуска и Ньютона с различном выбором длины шага α_k:

- константная длина шага: $\alpha_k = \alpha_0$.
- длина шага, удовлетворяющая условию Армихо с константой $c \in (0, 0.5)$: $\phi_k(\alpha) \leq \phi_k(0) + c_1\alpha\phi_k'(0)$, где $\phi_k(\alpha) = f(x_k + \alpha d_k)$, d_k — направление спуска.
- длина шага, удовлетворяющая сильным условиям Вульфа с константами $c_1 \in (0, 0.5), c_2 \in (c_1, 1)$.

В качестве критерия остановки выбран относительный квадрат нормы градиента:

$$||\nabla f(x_k)||_2^2 \leq \epsilon||\nabla f(x_0)||_2^2,$$

где ϵ — необходимая точность, x_0 — начальная точка.

Необходимо было провести эксперименты для сравнения данных методов. Среда выполнения: Python 3.

2 Вывод формул

В третьем пункте задания необходимо было представить функцию $f(x)$ в матрично-векторной форме:

$$f(x) = \frac{1}{m}\sum_{i=1}^{m} \ln\Big(1 + \exp\big(-b_i\langle a_i, x\rangle\big)\Big) + \frac{\lambda}{2}||x||_2^2.$$

Введем вектор $1_m = (1, \ldots, 1)^T$, состоящий из m единиц, $\odot$ - поэлементное умножение, тогда функцию $f(x)$ можно представить следующим образом:

$$f(x) = \frac{1}{m}\Big\langle 1_m, \ln\big(1 + \exp(-b \odot Ax)\big)\Big\rangle + \frac{\lambda}{2}||x||_2^2. \quad (1)$$

Скалярная функция $ln(1 + \exp(x)$ применяется к вектору поэлементно. Также нужно было найти градиент и гессиан данной функции в матрично-векторной форме.

$$df(x) = d\Big(\frac{1}{m}\Big\langle 1_m, \ln\big(1+\exp(-b\odot Ax)\big)\Big\rangle + \frac{\lambda}{2}||x||_2^2\Big) =$$
$$= -\frac{1}{m}\Big\langle 1_m, \frac{\exp(-b\odot Ax)\odot b\odot Adx}{1+\exp(-b\odot Ax)}\Big\rangle + \lambda\big\langle x, dx\big\rangle =$$
$$= -\frac{1}{m}\Big\langle 1_m, \sigma(-b\odot Ax)\odot b\odot Adx\Big\rangle + \lambda\big\langle x, dx\big\rangle = -\frac{1}{m}\Big\langle \sigma(-b\odot Ax)\odot b, Adx\Big\rangle + \lambda\big\langle x, dx\big\rangle =$$
$$= \Big\langle -\frac{1}{m}A^T\big(b\odot\sigma(-b\odot Ax)\big) + \lambda x, dx\Big\rangle \quad (2)$$

Здесь $\sigma(x) = \frac{1}{1+\exp(-x)}$ - сигмоидная функция. Продифференцируем сигмоидную функцию отдельно для дальнейшего вычисления гессиана:

$$\sigma'(x) = \Big(\frac{1}{1+\exp(-x)}\Big)' = \frac{\exp(-x)}{(1+\exp(-x))^2} = \sigma(x)(1-\sigma(x)) \quad (3)$$

$$d^2f(x) = d\Big(\Big\langle -\frac{1}{m}A^T\big(b\odot\sigma(-b\odot Ax)\big) + \lambda x, dx_1\Big\rangle\Big) =$$
$$= \Big\langle \frac{1}{m}A^T\Big(b\odot\sigma(-b\odot Ax)\odot\Big(1-\sigma(-b\odot Ax)\Big)\odot b\odot Adx_2\Big) + \lambda dx_2, dx_1\Big\rangle =$$
$$= \Big\langle \Big(\frac{1}{m}A^T\Sigma A + \lambda I_n\Big)dx_2, dx_1\Big\rangle, \quad (4)$$

где $\Sigma = \operatorname{diag}\Big[\sigma\big(-b_i\langle a_i, x\rangle\big)\Big(1-\sigma\big(-b_i\langle a_i, x\rangle\big)\Big)\Big]_{i=1}^n$ – диагональная матрица. Итого:

$$\nabla f(x) = -\frac{1}{m}A^T\big(b\odot\sigma(-b\odot Ax)\big) + \lambda x. \quad (5)$$

$$\nabla^2 f(x) = \frac{1}{m}A^T\Sigma A + \lambda I_n. \quad (6)$$

3 Эксперимент: Траектория градиентного спуска на квадратичной функции

Необходимо проанализировать работу градиентного спуска для разных квадратичных функций для различного выбора стратегий и для различного начального приближения.

В качестве квадратичной функции выступает $f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$. Во всех экспериментах $b = (0, 0)^T$

В первом случае была выбрана функция с единичной матрицей A. Были выбраны $x_0 = (4, 4)^T$, $x_0 = (1, 1)^T$. На Рис. 1 константная стратегия, на Рис. 2 — стратегия Армихо, на Рис. 3 — стратегия Вульфа.

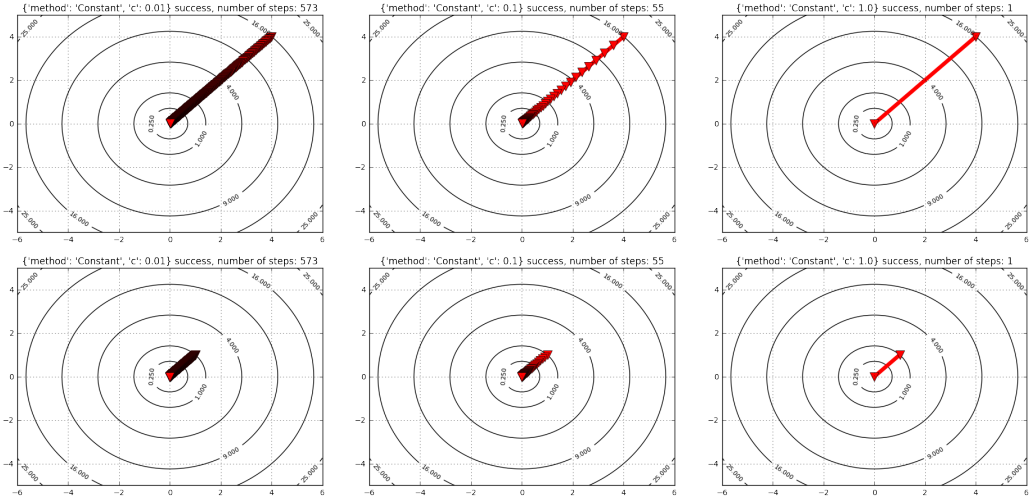


Рис. 1: Константный шаг. Одной строке соответствует одно начальное приближение, каждый столбец соответствует выбору длины шага.

В качестве сетки для константного шага были выбраны следующие значения: 0.01, 0.1, 1.0. По рисунку видно, что чем меньше шаг, тем больше нужно сделать итераций для сходимости, однако ясно, что если шаг будет слишком большим, то метод может не сойтись (см. в разделе с подбором длины шага). Видно также, что метод не зависит от начального приближения.

В стратегии Армихо выбрана следующая сетка для константы c_1: 0.4, 0.1, 0.0001. Как видно из рисунка метод делает один шаг и благополучно сходится вне зависимости от начального приближения.

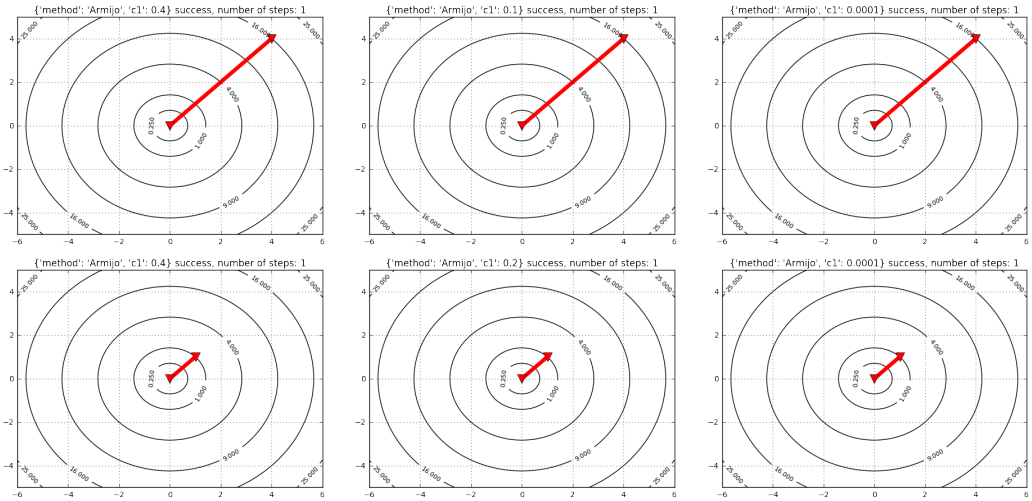


Рис. 2: Стратегия Армихо. Одной строке соответствует одно начальное приближение, каждый столбец соответствует выбору константы c_1.

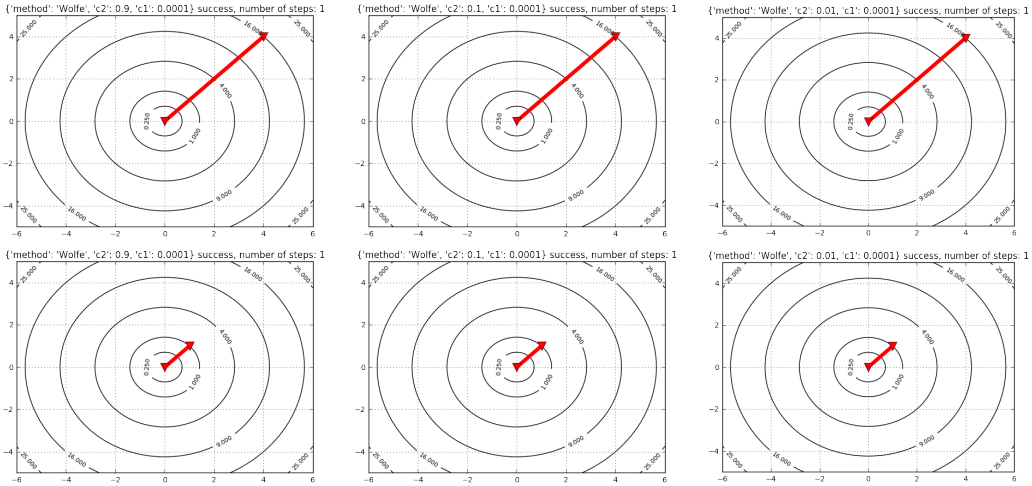


Рис. 3: Стратегия Вульфа. Одной строке соответствует одно начальное приближение, каждый столбец соответствует выбору константы c_2.

В стратегии Вульфа константа $c_1 = 0.0001$ и не меняется, а константа c_2 пробегает по следующей сетке: 0.9, 0.1, 0.01. Как и методы со стратегией Армихо здесь видна сходимость за одну итерацию.

В качестве следующей квадратичной задачи была выбрана матрица A: $a_{1,1} = a_{2,2} = 1$, $a_{2,1} = a_{1,2} = 0.9$. На Рис. 4, 5, 6 представлены соот-

ветственно константная стратегия, стратегия Армихо и Вульфа. Были выбраны следующие начальные приближения: $x_0 = (0, 4)^T$, $x_0 = (4, 3)^T$. Сетка для константной стратегии не изменилась, выбор констант для стратегий Армихо и Вульфа такая же как и для первой задачи.

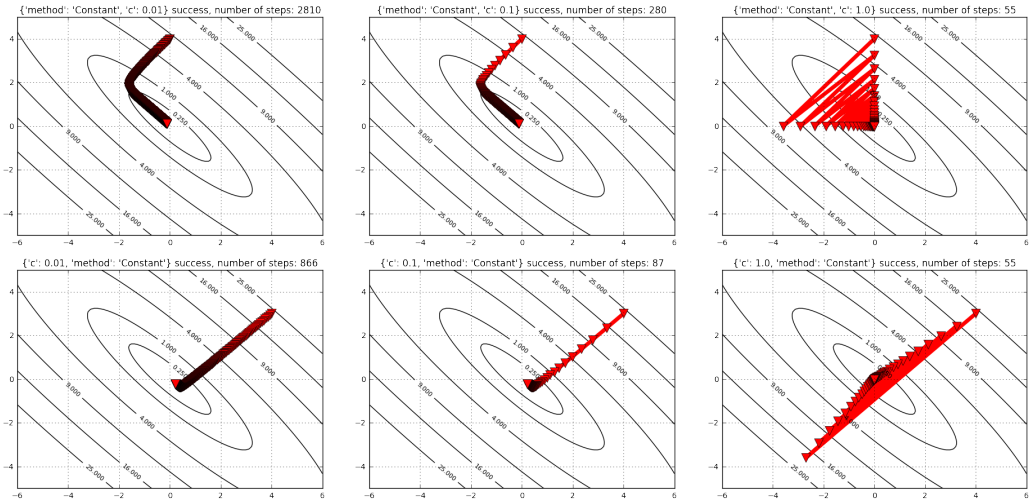


Рис. 4: Константный шаг. Одной строке соответствует одно начальное приближение, каждый столбец соответствует выбору длины шага.

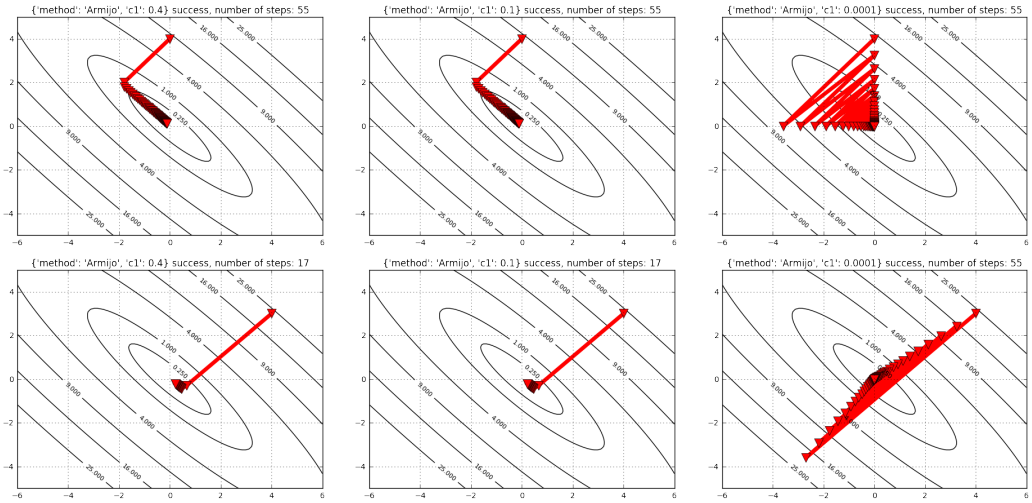


Рис. 5: Стратегия Армихо. Одной строке соответствует одно начальное приближение, каждый столбец соответствует выбору константы c_1.

Видно, что в константной стратегии выбор начального приближения играет важную роль. Чем лучше начальное приближение, тем меньше итераций необходимо. Также можно заметить, что с возрастанием длины шага уменьшается количество итераций.

В стратегии Армихо количество итераций зависит от начального приближения и от константы c_1. Чем меньше данная константа, тем больше нужно итераций для сходимости.

В стратегии Вульфа чем меньше константа c_2, тем меньше нужно итераций для сходимости. Также видно, что эта стратегия не зависит от выбора начального приближения, как стратегия Армихо.

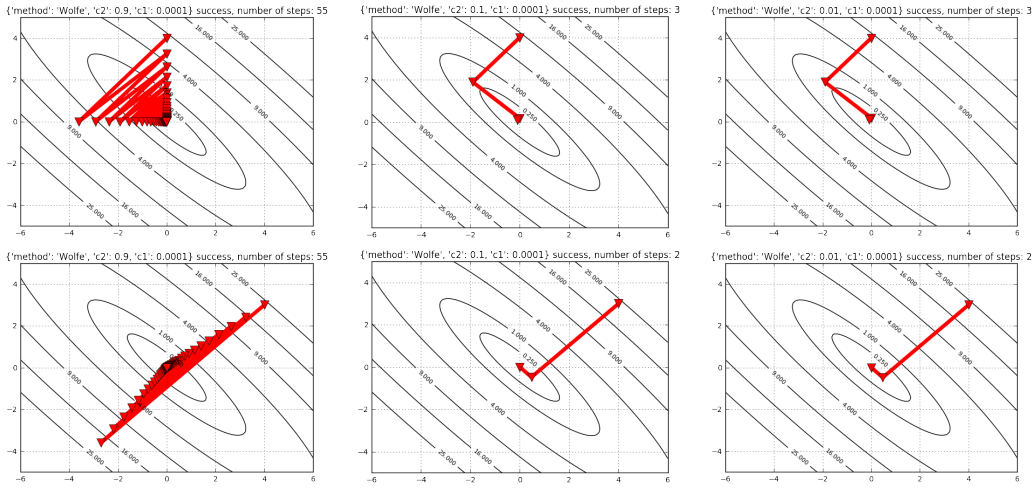


Рис. 6: Стратегия Вульфа. Одной строке соответствует одно начальное приближение, каждый столбец соответствует выбору константы c_2.

Во второй квадратичной задаче видна зигзагообразная траектория метода градиентного спуска.

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

В данном эксперименте необходимо было для квадратичной задачи показать зависимость между различным выбором числа обусловленно-

сти κ и размерности пространства с числом итераций. Матрица A диагональная, где $\kappa = \max_i a_{ii}$, $\min_i a_{ii} = 1$, а остальные элементы заполнены случайными числами от 1 до κ; b — случайный вектор. В качестве начального приближения берется $x_0 = 0$. По сетке $[2, 1000]$ перебирается значение κ, $n \in \{10, 100, 1000, 10000, 100000\}$ и эксперимент перезапускается 3 раза, так как каждая генерация квадратичной задачи случайна. На Рис. 7 представлен результат эксперимента.

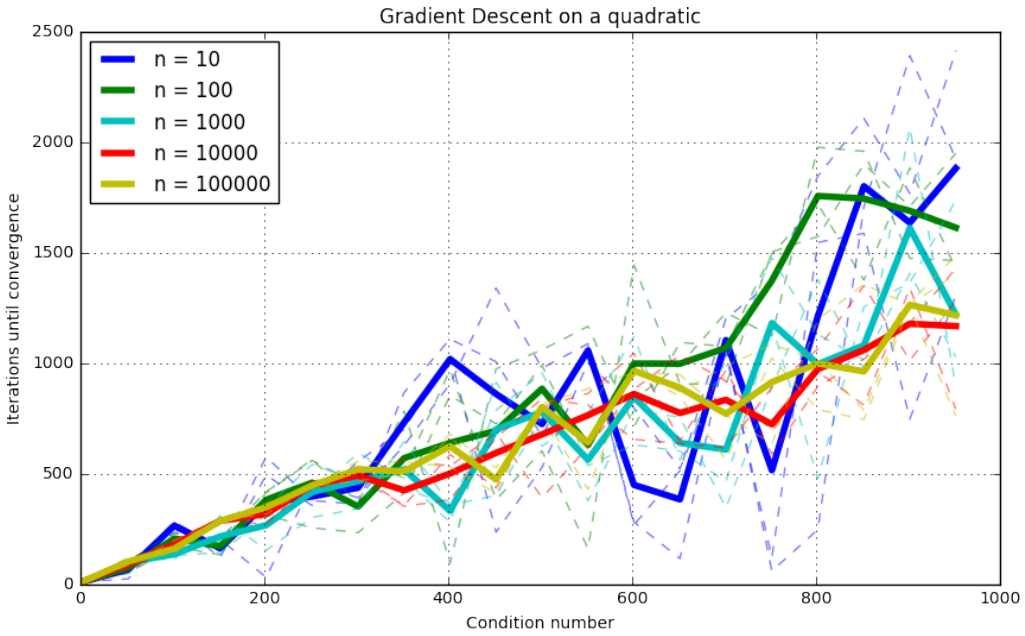


Рис. 7: Зависимость между количеством итераций против числа обусловленности и размерности пространства. Жирным выделено среднее количество итераций по трём перезапускам.

Видно, линейная зависимость между количеством итераций и числом обучловленности. Также заметно, что количество итераций для разных размерностей не сильно отличаются. Чем больше размерность, тем более зависимость становится похожей на линейную.

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

В данном эксперименте необходимо было решить задачу логистической регрессии с l_2-регуляризатором для трех наборов данных: w8a, gisette, real-sim. Предлагается решать задачу с помощью метода градиентного спуска и метода Ньютона, чтобы сравнить их реальное время работы.

Нужно было построить графики сходимости следующих двух видов:

- Зависимость значения функции от реального времени работы метода.
- Зависимость относительного квадрата нормы градиента $\frac{||\nabla f(x_k)||_2^2}{||\nabla f(x_0)||_2^2}$ в логарифмической шкале против реального времени работы.

Графики для датасетов представлены на Рис. 8, 9, 10

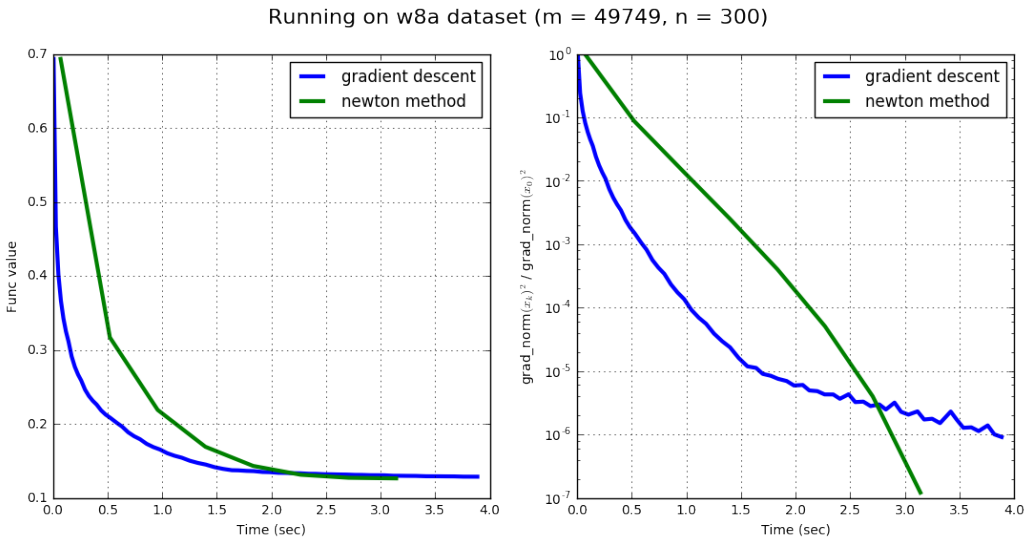


Рис. 8: Датасет w8a.

Для датасета w8a видно, что сначала метод Ньютона уступает по реальной времени работы и по скорости сходимости, но затем начинает сходится быстрее.

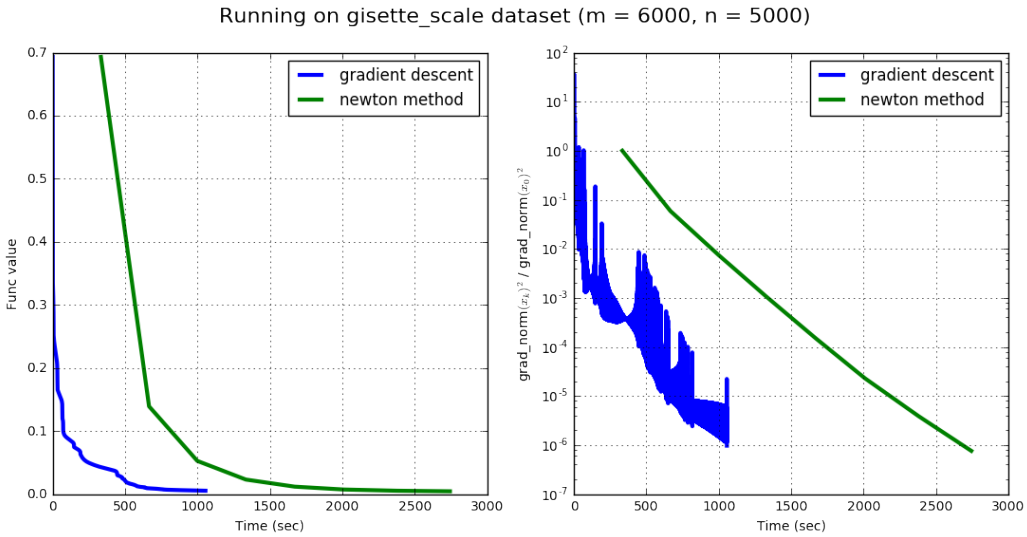


Рис. 9: Датасет gisette.

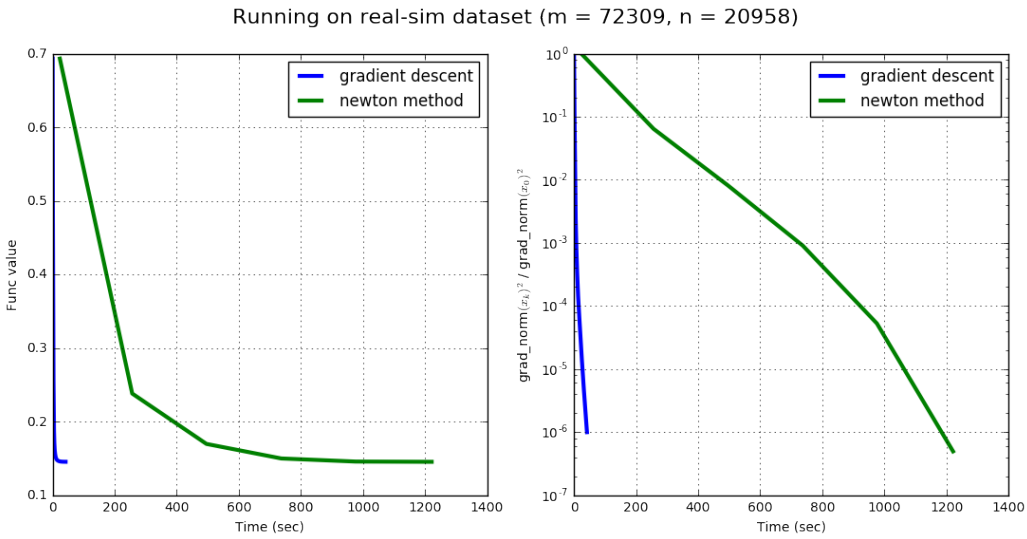


Рис. 10: Датасет real-sim.

В датасетах gisette, real-sim не наблюдается картины как на датасете w8a, так как метод градиентного спуска работает быстрее из-за того не нужно на каждой итерации решать систему уравнений с гессианом, как

в методе Ньютона. Также видно, что кривая метода Ньютона сдвинута — это из-за подсчета гессиана в данной точке. Тем самым сложность одной итерации градиентного спуска складывается только из сложности подсчета градиента $O(mn)$, а в памяти нужно хранить градиент $O(n)$ и матрицу A $O(mn)$. Для метода Ньютона сложность складывается из сложности подсчета гессиана $O(mn^2)$, сложности подсчета градиента $O(mn)$, сложности решения системы уравнения с гессианом $O(n^3)$, в памяти необходимо хранить гессиан и градиент, что в сумме составляет $O(n^2)$, а также матрицу A $O(mn)$.

Для градиентного спуска: сложность $O(mn)$, память $O(mn)$.

Для метода Ньютона: сложность $O(mn^2 + n^3)$, память $O(n^2 + mn)$.

6 (Бонусная часть) Эксперимент: Оптимизация вычислений в градиентном спуске

В данном эксперименте необходимо было сравнить работу метода с оптимизированным оракулом логистической регрессии и обычным по трем графикам:

- Зависимость значения функции от номера итерации.
- Зависимость значения функции от реального времени работы метода.
- Зависимость относительного квадрата нормы градиента $\frac{||\nabla f(x_k)||_2^2}{||\nabla f(x_0)||_2^2}$ в логарифмической шкале против реального времени работы.

Графики представлены на Рис. 11

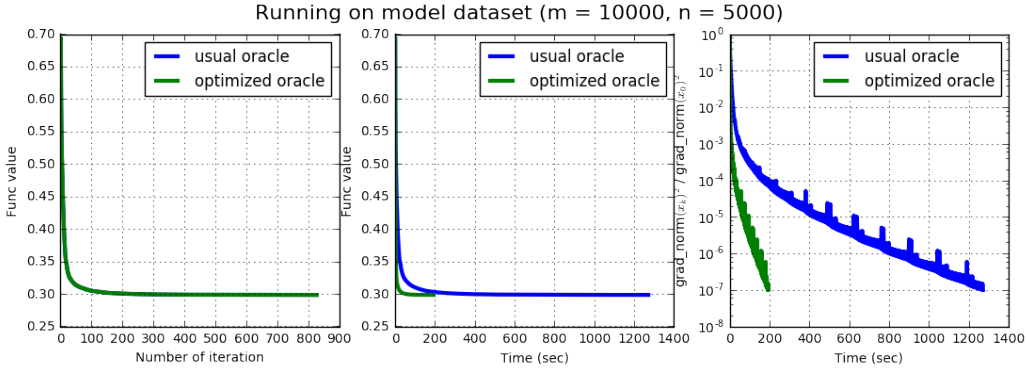


Рис. 11: Сравнение между оптимизированным оракулом и обычным.

По графикам видно, что оптимизированный оракул работает гораздо быстрее, так как не нужно каждый раз тратить $O(mn)$ для подсчет Ax_k и Ad_k в линейном поиске. Также по первому видно, что оптимизированный оракул работает также как и обычный по итерациям, так как оптимизируется сама процедура линейного поиска, а α_k при этом остается таким же как и при обычном оракуле, иными словами оптимизируется время, а не количество итераций.

7 (Бонусная часть) Эксперимент: Стратегия выбора длины шага в градиентном спуске

В данном эксперименте необходимо было показать влияние стратегии длины шага на скорость сходимости по невязке функции в квадратичной задаче и на скорость сходимости по относительной норме градиента в задаче логистической регрессии. Нужно было попробовать разные начальные приближения.

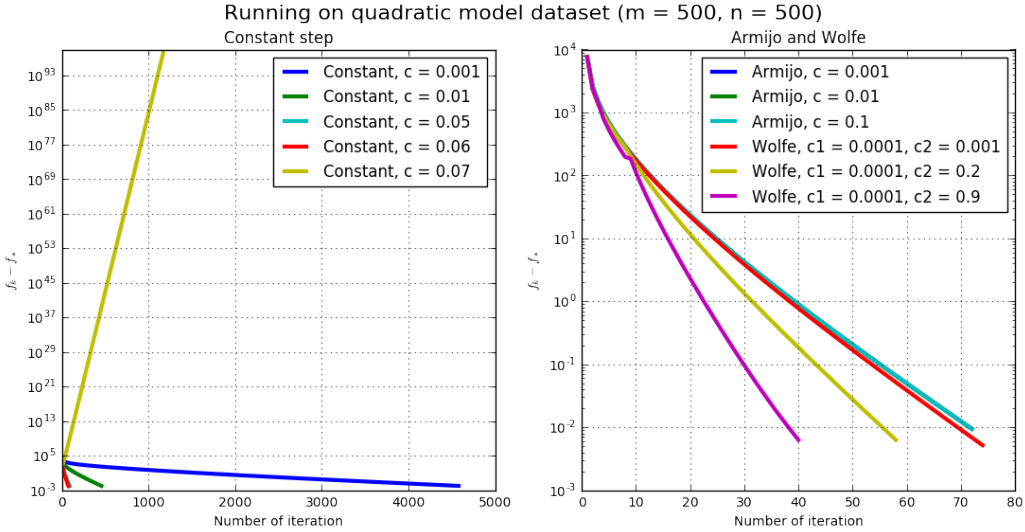


Рис. 12: Квадратичная задача.

Видно, что метод очень чувствителен к выбору константного шага. При увеличении c уменьшается количество итераций, и при $c = 0.06$ метод сходится достаточно хорошо, а при $c = 0.07$ уже полностью расходится. Адаптивные стратегии сами подбирают нужный шаг и сходятся гораздо быстрее. При увеличении константы c_2 в стратегии Вульфа увеличивается скорость сходимости. Стратегии Армихо с разными константами работают примерно одинаково.

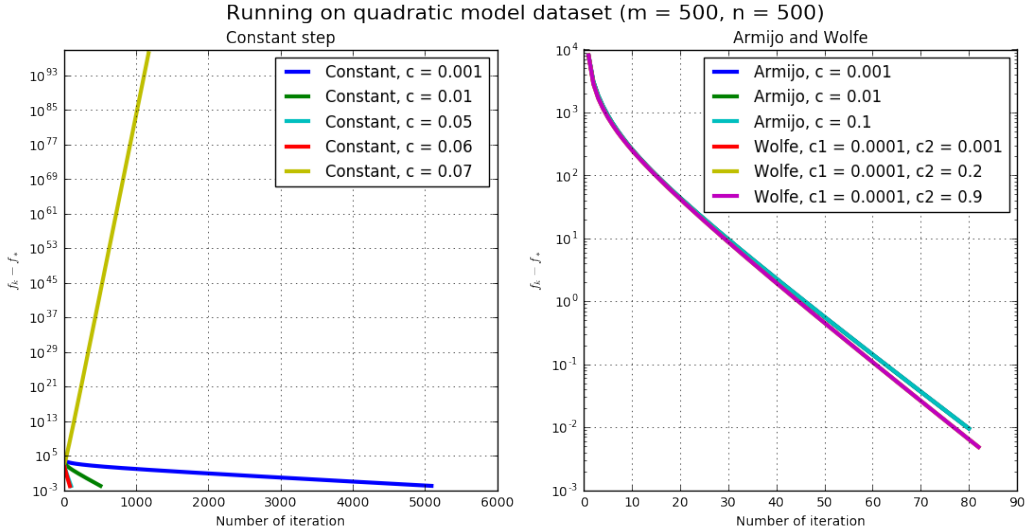


Рис. 13: Квадратичная задача.

В задаче логистической регрессии также видно, что константная стратегия при больших c расходится, а при маленьких сходится очень медленно. В стратегии Армихо при увеличении c_1 уменьшается скорость сходимости. Стратегии Вульфа работают одинаково.

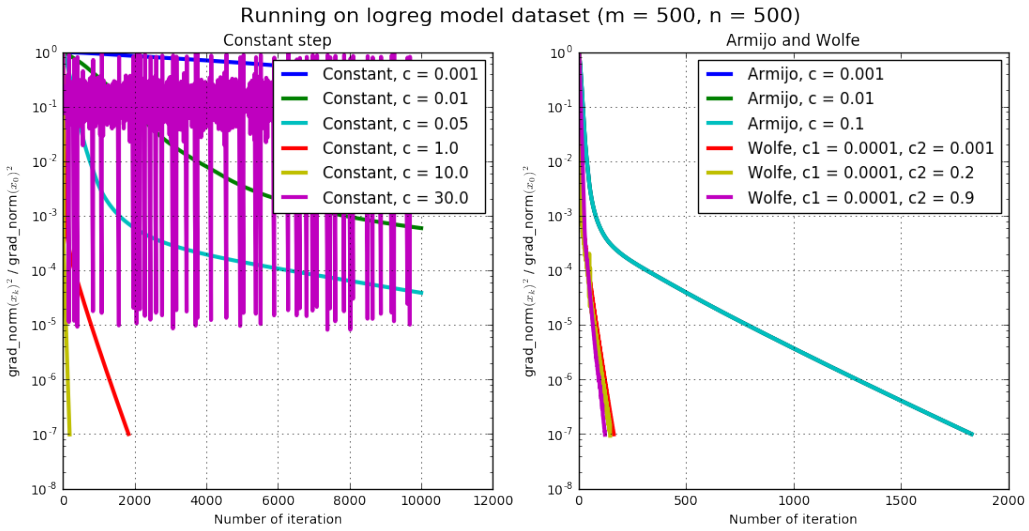


Рис. 14: Задача логистической регрессии.

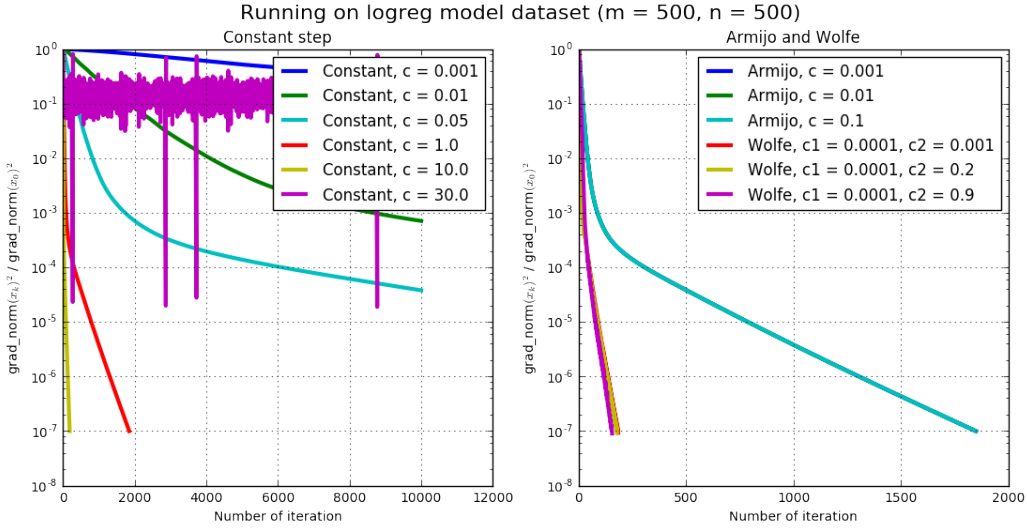


Рис. 15: Задача логистической регрессии.

8 (Бонусная часть) Эксперимент: Стратегия выбора длины шага в методе Ньютона

В данной части необходимо повторить эксперименты предыдущего пункта для метода Ньютона.

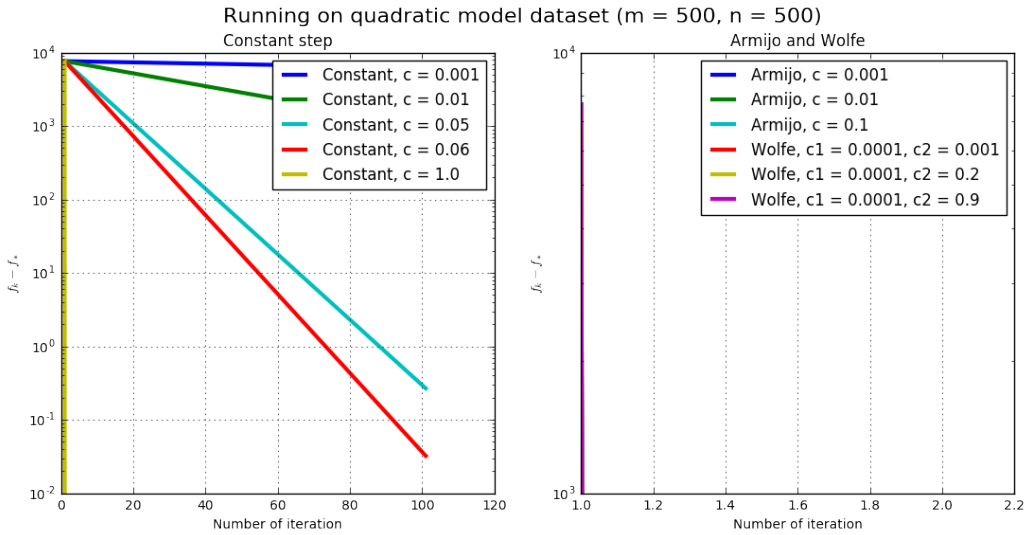


Рис. 16: Квадратичная задача.

Видно, что при маленькой константе в константном варианте, метод сходится очень долго, а при константе $c = 1.0$ сходится за одну итерацию и видна почти линейная сходимость. Стратегии Армихо и Вульфа сходятся также за одну-две итерации.

В задаче логистической регрессии адаптивные стратегии не сильно отличаются друг от друга и видна квадратичная сходимость. Для константного выбора шага при $c > 1.0$ метод начинает расходится, а при остальных c сходится довольно медленно.

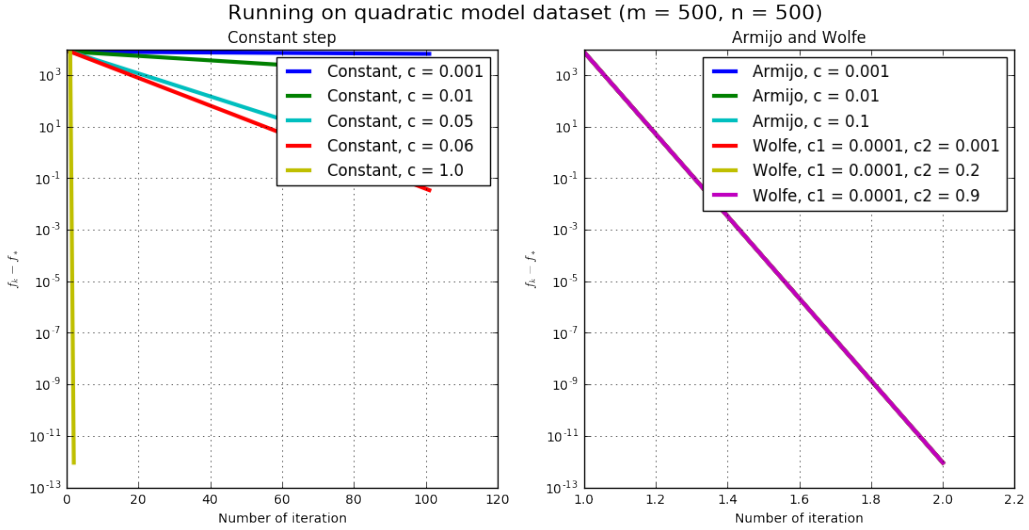


Рис. 17: Квадратичная задача.

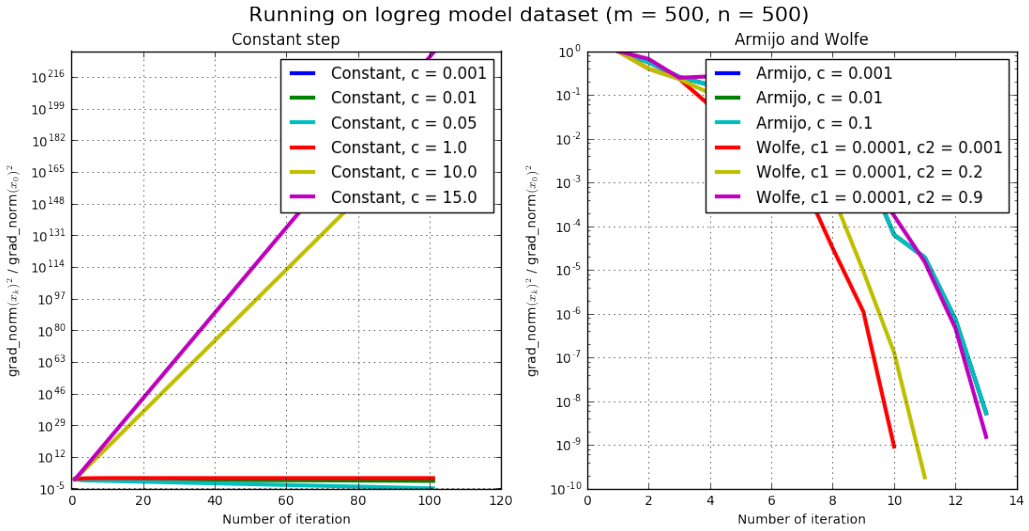


Рис. 18: Задача логистической регрессии.

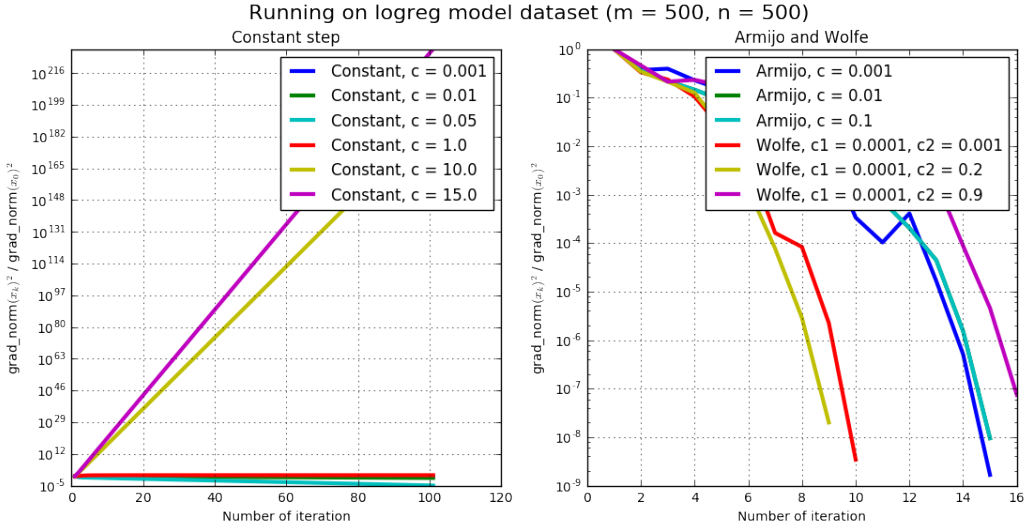


Рис. 19: Задача логистической регрессии.

9 Выводы

По графикам можно сделать следующие выводы:

- Методы с адаптивной стратегией сходятся быстрее, подбирая на каждой итерации нужный шаг.
- Градиентный спуск работает быстрее, чем метод Ньютона по времени, где имеется огромная размерность пространства. Однако метод Ньютона сходится быстрее, чем градиентный спуск в задачах с малым количеством признаков
- При увеличении размерности пространства количество итераций градиентного спуска для различных чисел обусловленности остается одинаковым.