Домашнее задание 1 по предмету "Методы оптимизации"

Михаил Лобанов, Б05-926

27 ноября 2021 г.

1 Теоретические результаты

В рамках решения задачи о логистической регрессии мы минимизируем следующую функцию:

$$f = \frac{1}{m}\sum_{i=1}^{m} \ln\left(1 + \exp\left(-b_i, \langle a_i, x\rangle\right)\right) + \frac{\lambda}{2}\|x\|_2^2 \xrightarrow[x\in\mathbb{R}^n]{} \min \tag{1}$$

Градиент и гессиан выражаются следующим образом:

$$\nabla f = \lambda \cdot x - \frac{1}{m}A^T(b \cdot \frac{1}{1 + \exp\left(b^T \cdot Ax\right)}) \tag{2}$$

$$\nabla^2 f = \frac{1}{m}A^T \frac{1}{1 + \exp\left(b^T \cdot Ax\right)}(1 - \frac{1}{1 + \exp\left(b^T \cdot Ax\right)})I_n A + \lambda \cdot I_n \tag{3}$$

Функция в матрично-веткорной форме

$$f = \frac{1}{m}\left\langle 1_m, \ln\left(1 + \exp\left(-b \odot Ax\right)\right)\right\rangle + \frac{\lambda}{2}\|x\|_2^2 \tag{4}$$

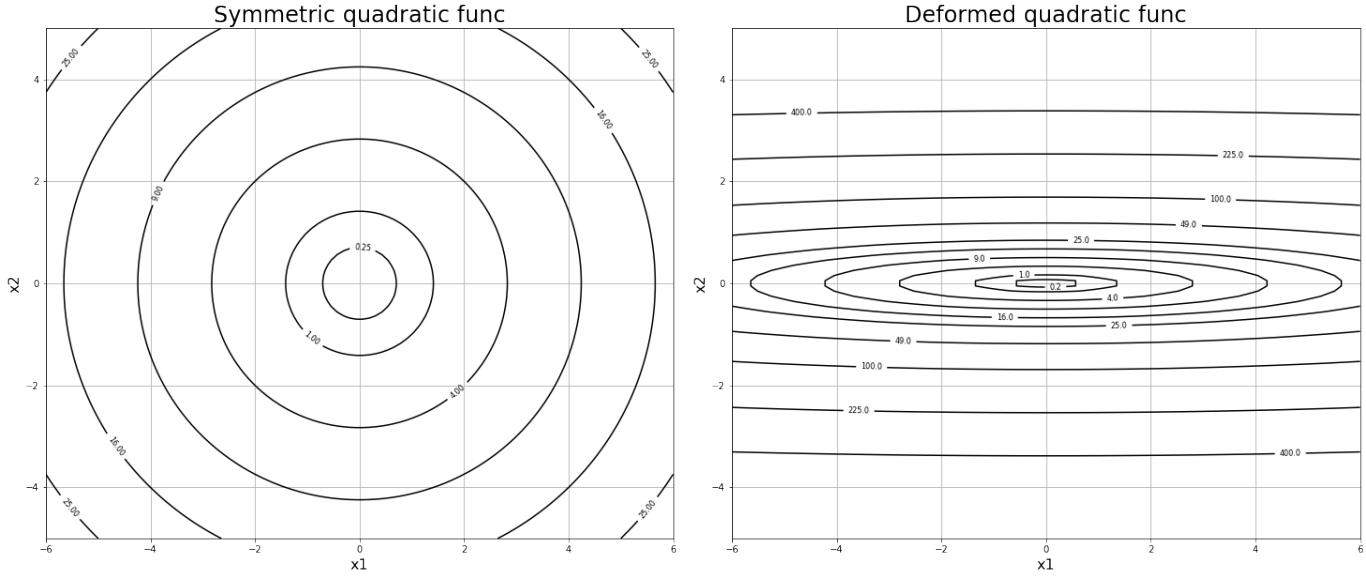
Пояснение. *Размерности* $b \mapsto (m, 1), A \mapsto (m, n), x \mapsto (n, 1)$.

2

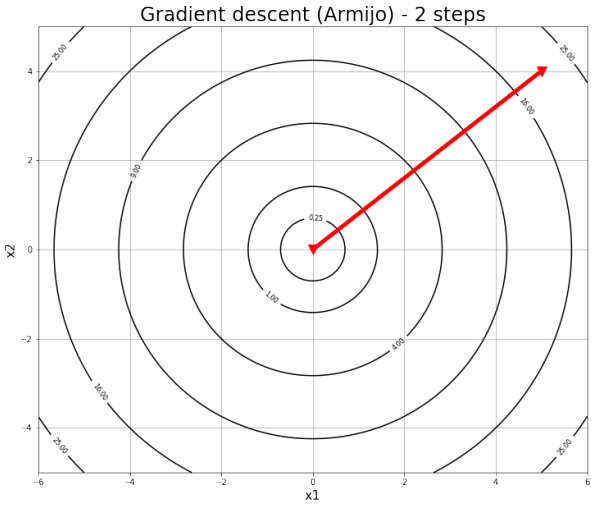
3 Эксперименты

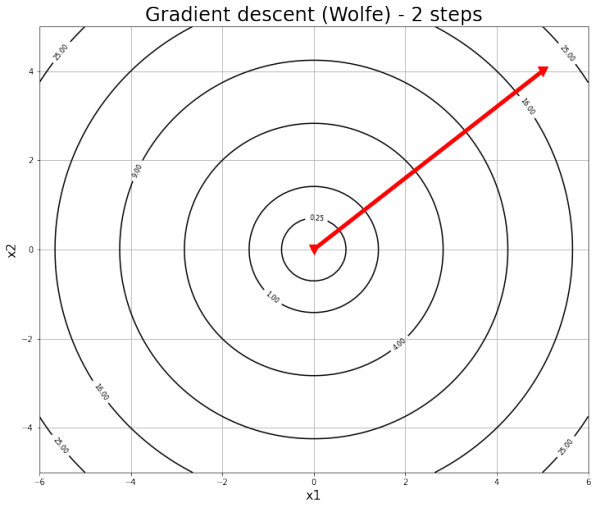
3.1 Траектория градиентного спуска на квадратичной функции

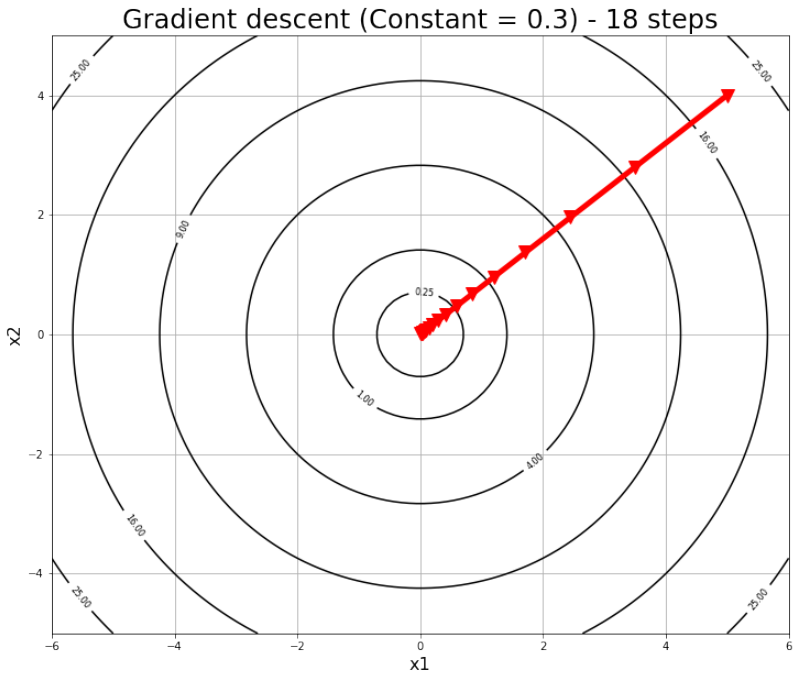
Градиентный спуск чувствителен к сжатию осей. Рассмотрим две квадратичных функции, одна с обычными осями, а другая - "сплюснутая".



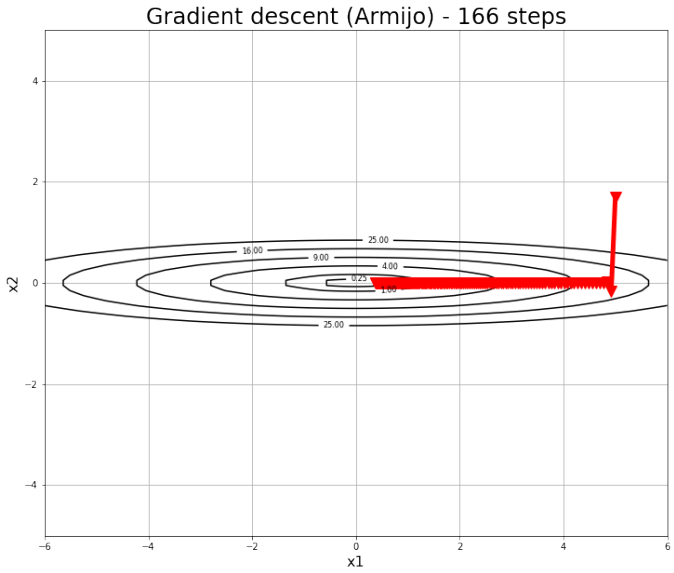
Симметричная функция. С ней справляется градиентный спуск в любой модификации: и с константным подбором шага, и с правилом Вульфа, и с правилом Армихо. Разве что в константном случае шагов требуется много.

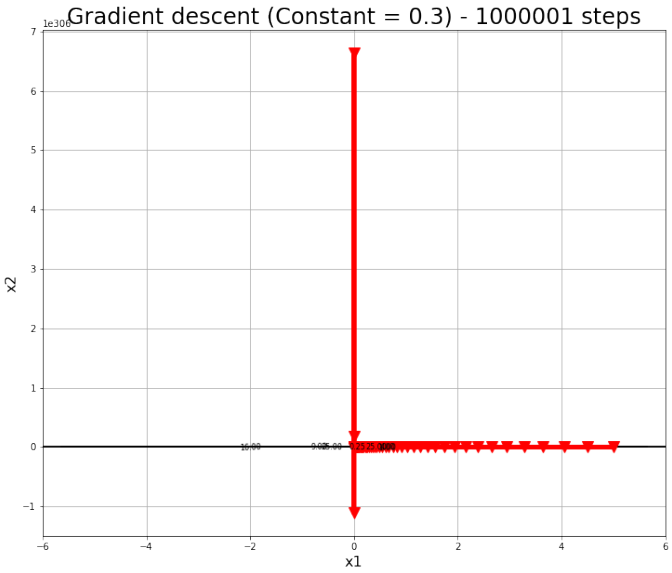


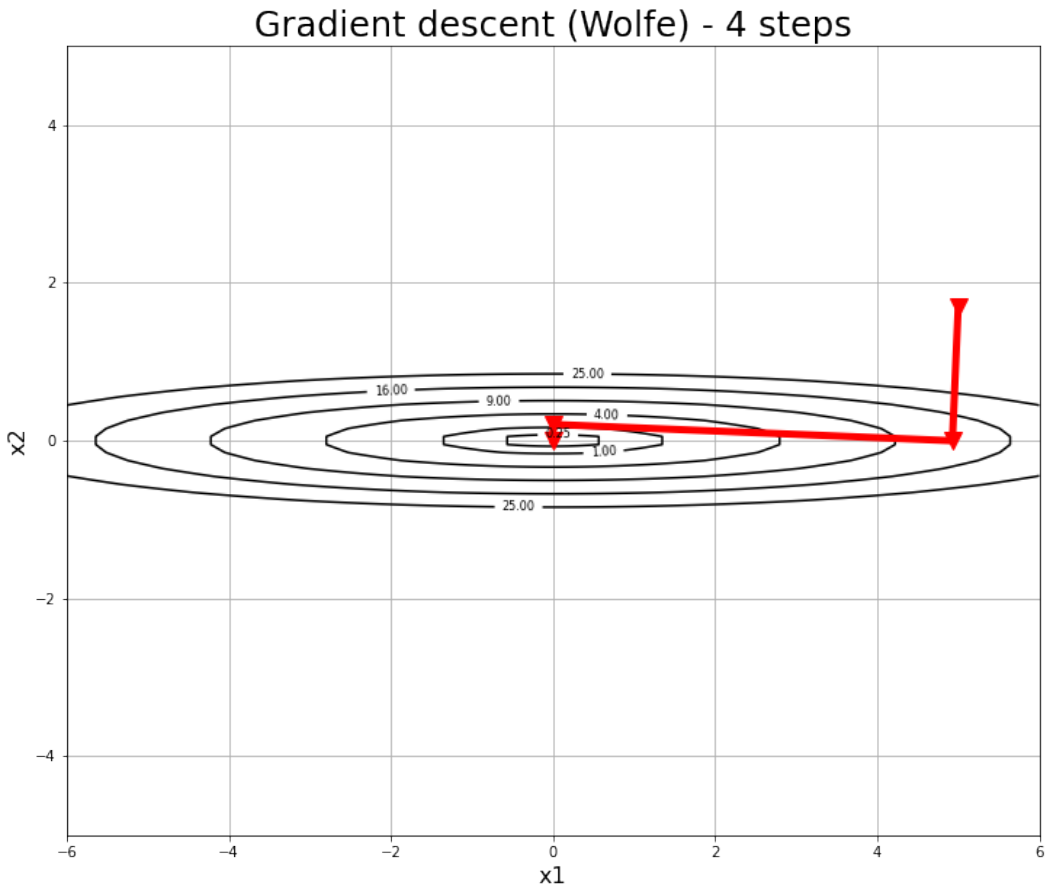




Сжатые оси. При деформации осей алгоритм начинает работать сильно хуже. На соответствующей функции GD с Армихо стал "петлять"из стороны в сторону, GD с константным шагом вообще не получилось свести, и только Вульф показал себя хорошо.







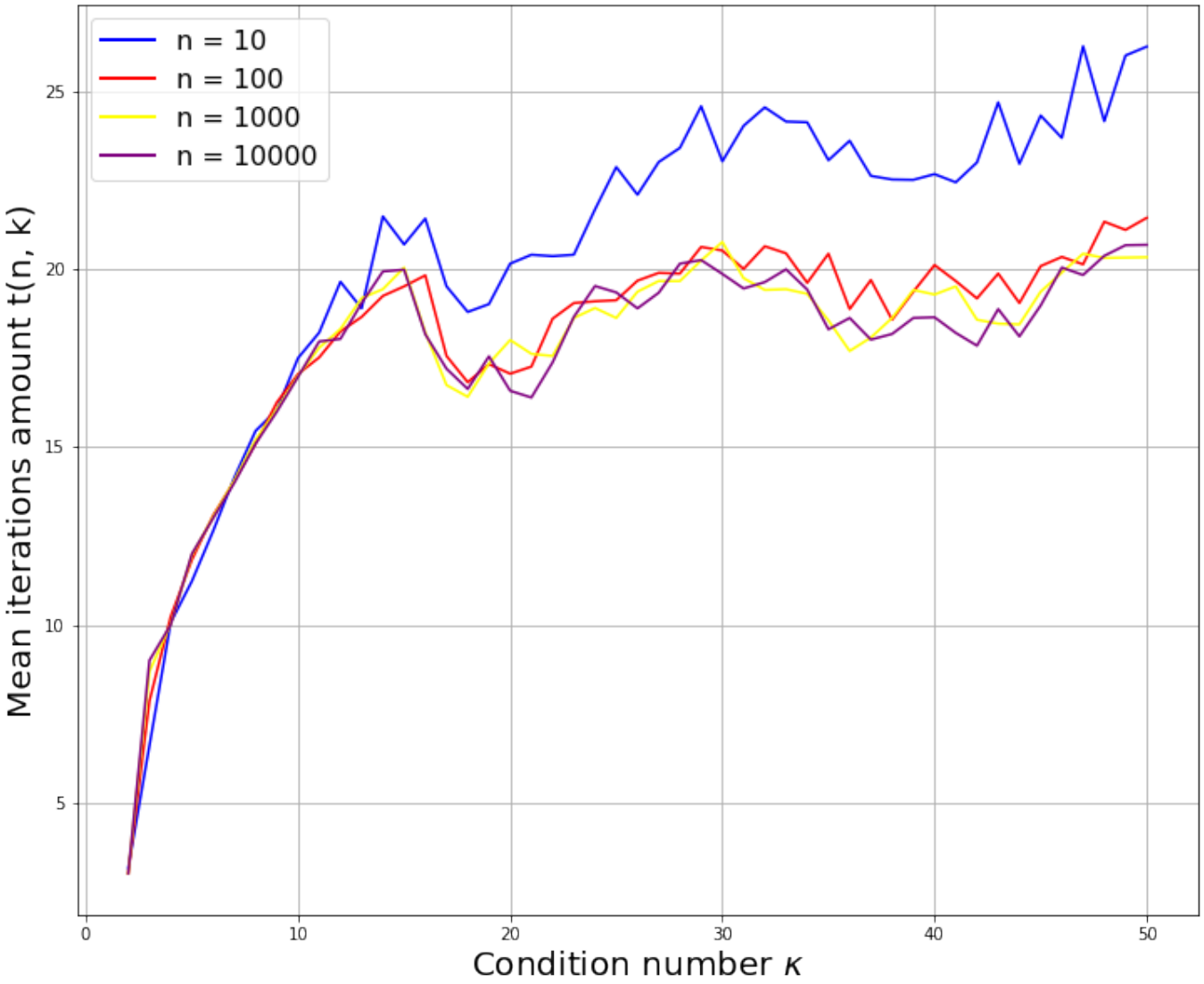
Эксперименты показали, что нет качественной зависимости между сходимостью и начальной точкой (либо из всех сходится, либо из всех не сходится). Видимо поэтому начальную точку часто выбирают случайно.

3.2 Зависимость числа итераций градиентного спуска от числа обусловленности и размерности пространства

Рассмотрим 4 различных размерности n: $[10, 10^2, 10^3, 10^4]$ и 49 чисел обусловленности κ: от 2 до 49 включительно. Для каждой пары (n, κ) сформулируем 100 задач минимизации квадратичной функции размерности n и числом обусловленности κ.

Решим каждую задачу и подсчитаем количество итераций $T_i(n, \kappa), i \in \{1, \ldots, 100\}$, которое прошел GD номер i.

Посмотрим на среднее количество итераций при каждом κ и n:



Видим, что примерно до $n = 15$ есть четка корреляция между числом обусловленности и количеством итераций. После зависимость перестает быть такой ясной.

Нанесем на график вообще все результаты эксперимента:

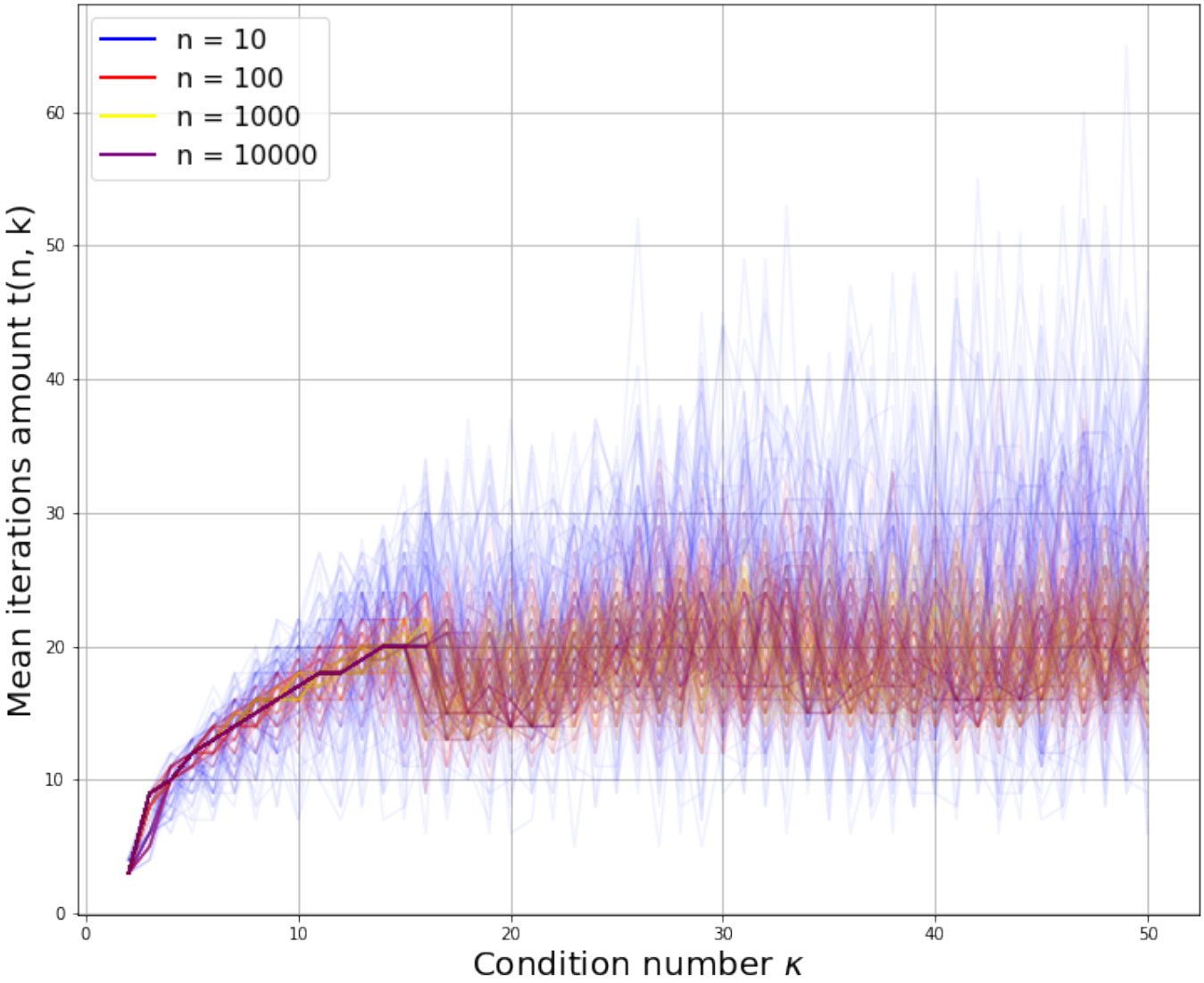


График наталкивает на мысль, что чем больше размерность, тем меньше дисперсия количества итераций, а чем больше κ, тем она выше. Проверим:

Действительно, график стандартного отклонения показывает, что чем больше κ, тем больше разброс в количестве итераций GD. И чем меньше размерность, тем это виднее.

3.3 Сравнение методов градиентного спуска и Ньютона на реальной задаче логистической регрессии

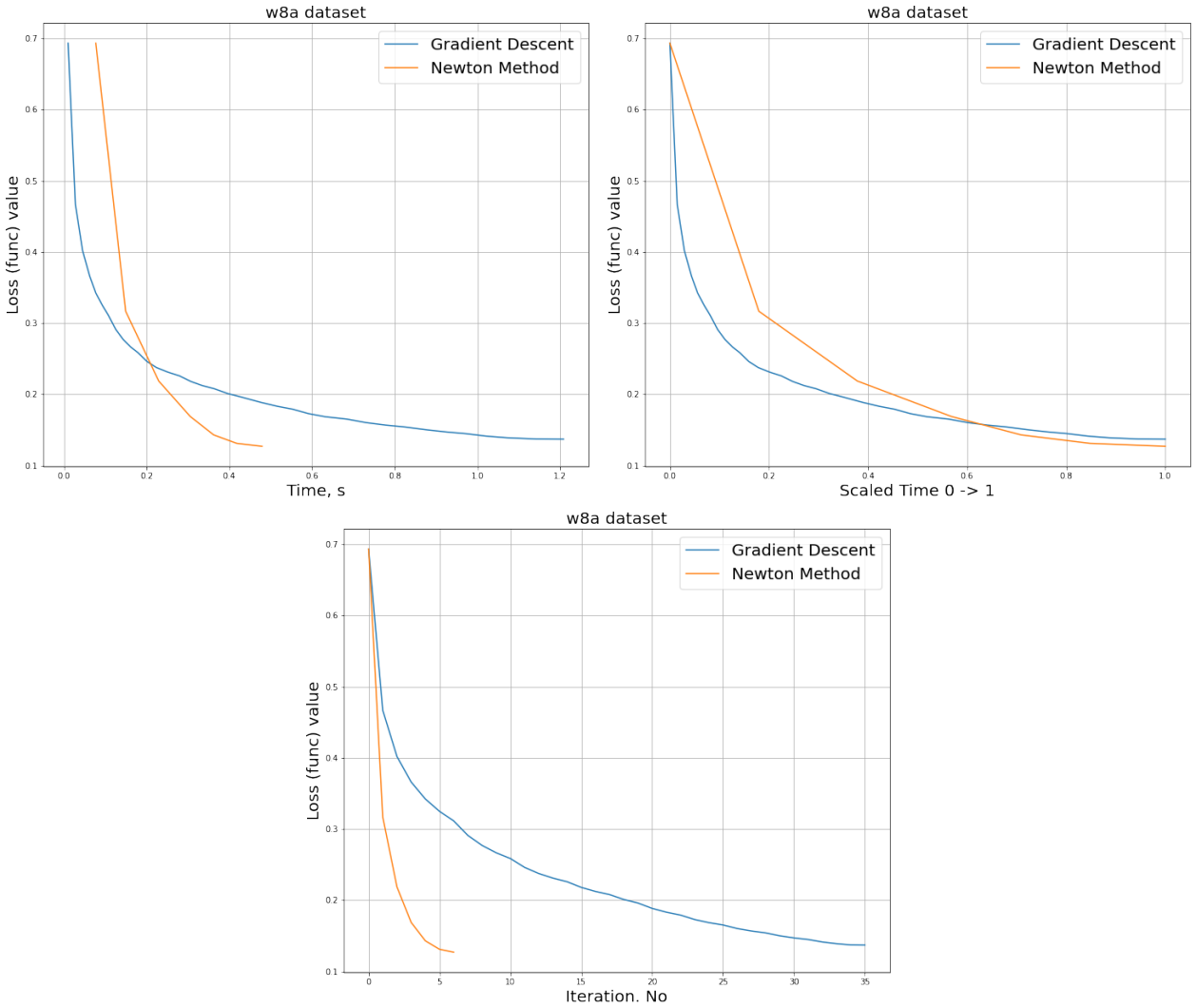
Все три датасета (w8a, gisette, real-sim) приводят к следующим выводам:

1. Метод Ньютона сходится гораздо быстрее, чем метод градиентного спуска: требуется меньше итераций, со временем (по мере спуска) GD теряет эффективность сильнее, чем NM.
2. Метод Ньютона сильно более затратный. На одну итерацию требуется больше времени и памяти.
3. Иногда NM невыгоден, поскольку хоть и сходится быстро, но слишком медленно. Пускай GD и делает в разы больше итераций, но он делает их за секунды.
4. Нормированный градиент убывает очень похоже на значение самой функции с точностью до масштаба.

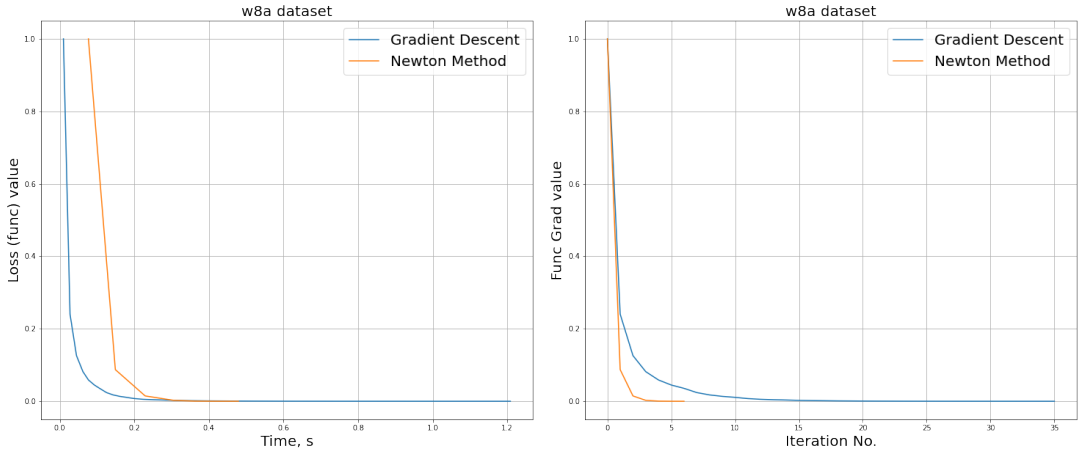
Далее приведем графики экспериментов.

Датасет w8a.

Зависимость значения функции от времени, от масштабированного времени (на отрезок $[0, 1]$), от номера итерации.

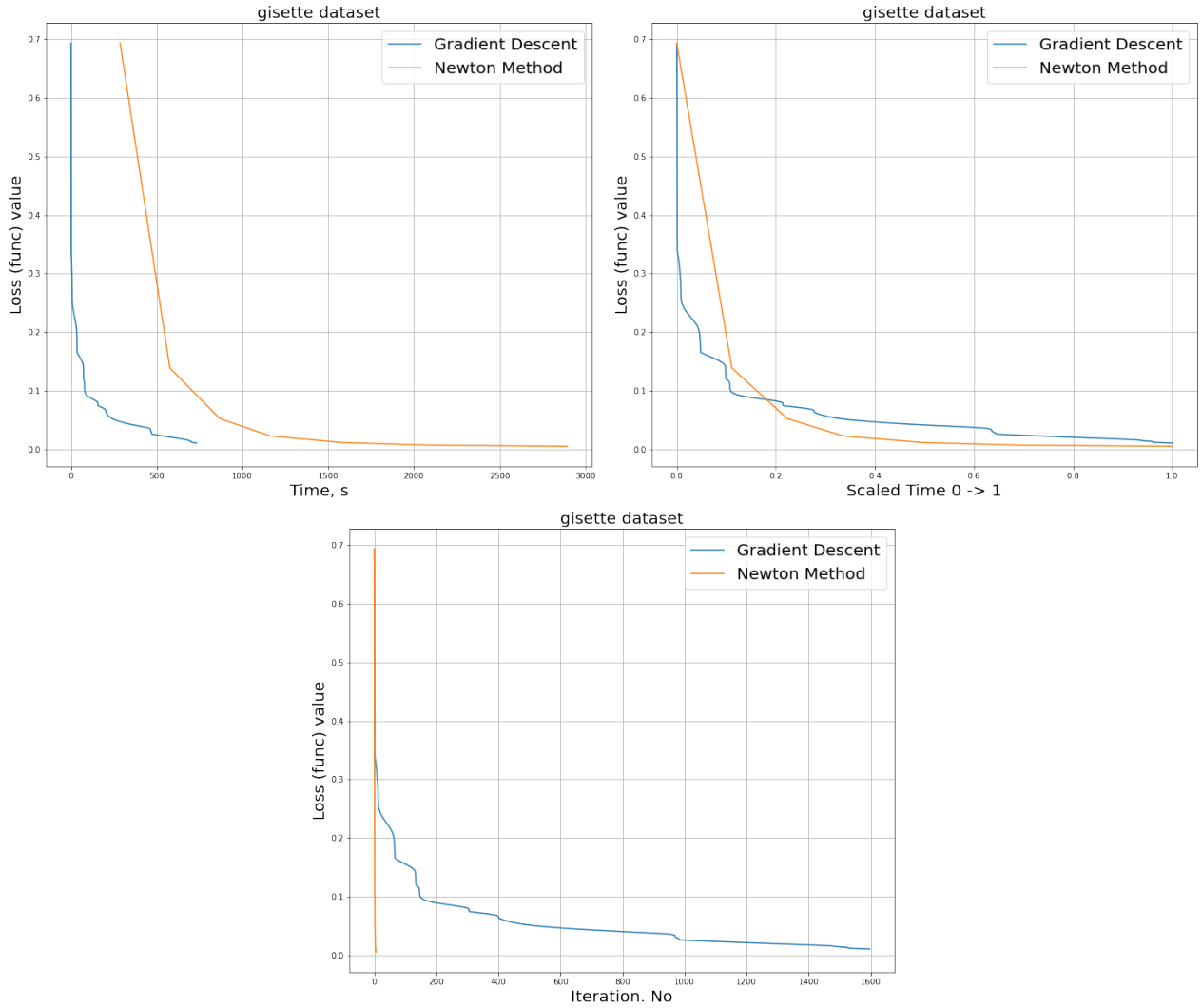


Видим, что в данном случае метод Ньютона оказался быстрее. Посмотрим на норму градиента.
Note. На одном из графиков перепутана подпись оси ординат.



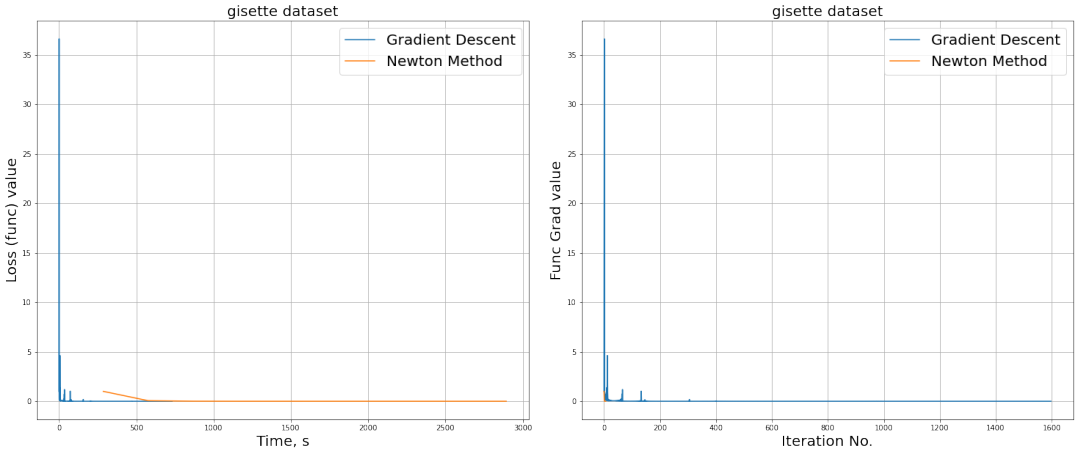
Датасет gisette.

Зависимость значения функции от времени, от масштабированного времени (на отрезок $[0, 1]$), от номера итерации.



В данном случае метод Ньютона оказался медленнее. Посмотрим на норму градиента.

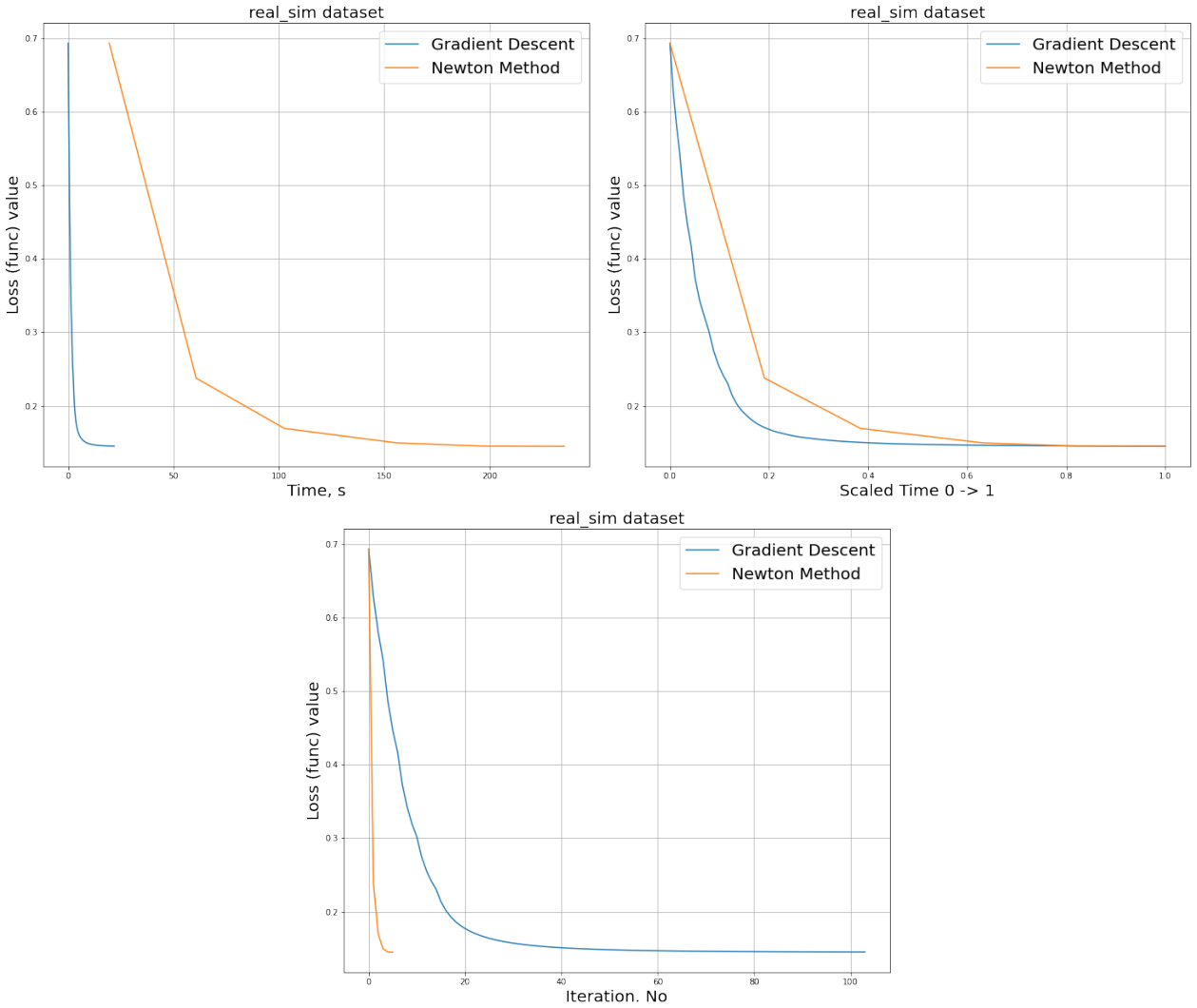
Note. На одном из графиков перепутана подпись оси ординат.



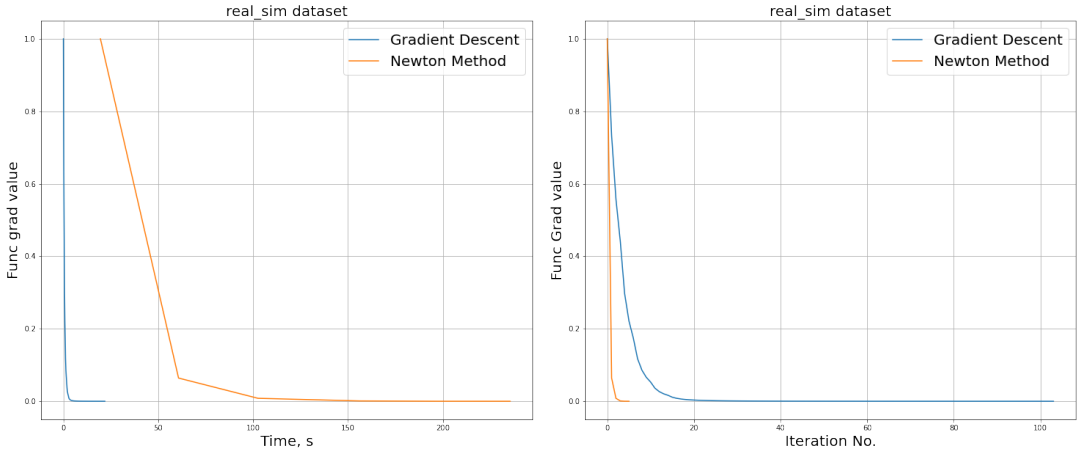
По каким-то причинам методы долго бродили вокруг оптимальной точки, пытаясь в нее попасть.

Датасет real-sim.

Зависимость значения функции от времени, от масштабированного времени (на отрезок $[0, 1]$), от номера итерации.



В данном случае метод Ньютона оказался медленнее. Посмотрим на норму градиента.
Note. На одном из графиков перепутана подпись оси ординат.



Видим, насколько быстрее метод Ньютона уменьшает градиент, чем градиентный спуск.

Приведем оценки одной итерации градиентного спуска и метода Ньютона по времени и памяти:

	GD	NM
Time	$\mathcal{O}(n^2)$	$\mathcal{O}(n^3)$
Memory	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$