Национальный исследовательский университет «Высшая школа экономики»
Санкт-Петербургская школа физико-математических и компьютерных наук

Отчет по Лабораторной работе 2: Продвинутые методы безусловной оптимизации

Выполнил: Кудреватых П.С.

Санкт-Петербург 2024

Содержание

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

В этой задаче продолжается оптимизация функции квадратичного оракула вида:

$$L = \frac{1}{2} \cdot \langle Ax, x \rangle - \langle b, x \rangle$$

Исследуем, как зависит число итераций, необходимое методу сопряженных градиентов для сходимости, от следующих двух параметров:
1) числа обусловленности $\kappa \geq 1$ оптимизируемой функции
2) размерности пространства n оптимизируемых переменных

Входные данные для матрицы A будем генерировать случайным образом, по следующему рецепту: возьмем идею о постоении матрицы из методички, а потом поделим ее на число обусловленности, нормировав ее, как предложил Данил Сморчков. Без этого эксперимент с постоянным шагом давал странные результаты. Вектор b возьмем как выборку из нормального распределения для каждой задачи. Начальное приближение я задаю как вектор из единиц с гауссовым нормальным шумом.

Зададим тестировочную сетку параметров, под которые мы будем генерировать тестовые данные. Размерность переберем как степени двойки, число обусловленности - линейно, начиная от 10 и заканчивая 1000. Верхнюю границу числа обусловленности я взял, посоветовавшись с Андреем Широбоковым, так как нужно было найти оптимум между вычислительным временем и размером сетки. Для увеличения точности измерения для каждой «точки» на сетке эксперимент выполнен 10 раз.

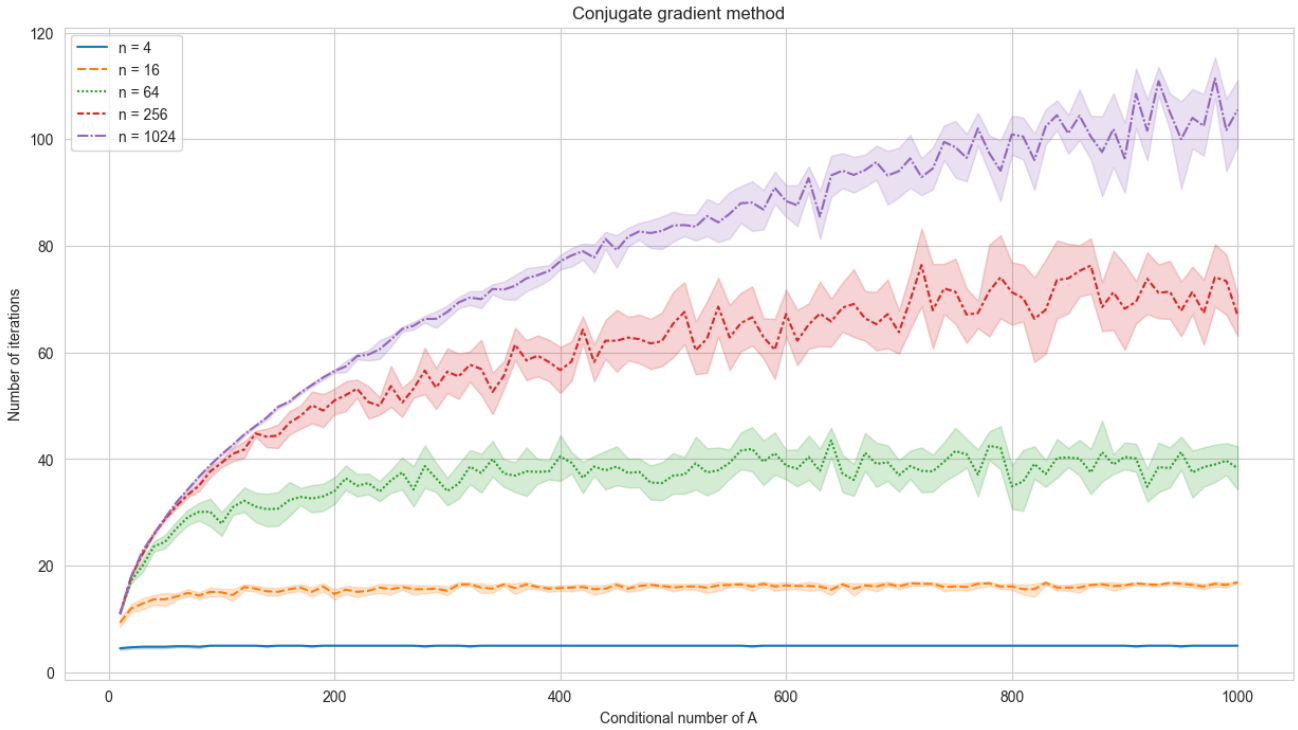

Рис. 1: Зависимость количества шагов от числа обусловленности и размерности пространства для метода сопряженных градиентов

1.1 Вывод

В этом эксперименте мы изучили зависимость количества итераций метода сопряженных градиентов на квадратичном оракуле. Видно, что эксперимент согласуется с теорией - для нахождения оптимума при увеличении размерности пространства число итераций увеличивается, но не превосходит размерности пространства. Наблюдается теоретическая зависимость $O(\sqrt{\kappa})$ (Это я подсмотрел в методичке ФКН ВШЭ).

2 Выбор размера истории в методе L-BFGS

Здесь мы изучим влияние размера истории в методе L-BFGS на поведение метода. Как мы знаем, сложность по времени у нас зависит от размера истории l и размерности пространства n как $O(nl)$.

Проведем тестирование, обучив логистическую регрессию на датасете *news*20.*binary* с точностью $1e-9$ и линейным поиском методом Вульфа ($c_1 = 1e-4, c_2 = 0.9$). Коэффициент регуляризации взят стандартным образом: $\lambda = \frac{1}{m}$.

Начальной точкой в каждом эксперименте является $\overrightarrow{x_0} = \overrightarrow{0}$. Будем рассматривать такие значения памяти, как $l = 0, l = 1, l = 5, l = 10, l = 50, l = 100$.

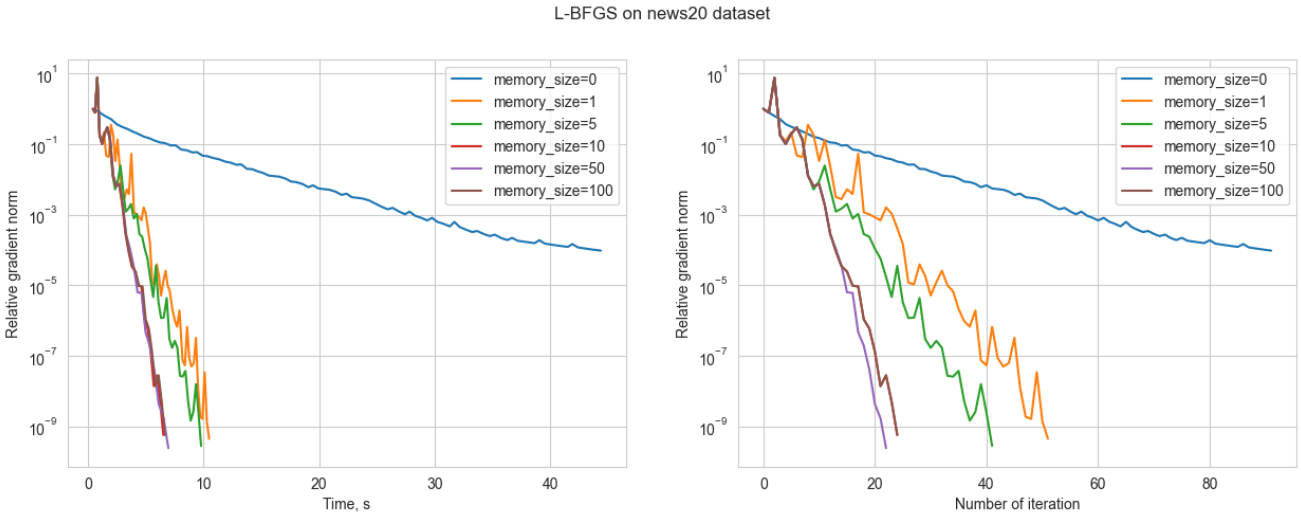

Рис. 2: Зависимость относительного квадрата нормы градиента от времени в логарифмическом масштабе (слева); Зависимость относительного квадрата нормы градиента от количества итераций в логарифмическом масштабе (справа)

2.1 Выводы

Для датасета *news*20.*binary* была обучена логистическая регрессия методом L-BFGS с шестью различными размерами истории. Видно, что при нулевом размере истории метод ведет себя точь в точь, как градиентный спуск. Из-за этого я ограничил ему точность до $1 - e4$, чтобы не портить масштаб других

картинок. А если учитывать историю, то сходимость сразу же становится квадратичной и задача решается намного быстрее. Видно, что при увеличении размера деки истории с 1 до 50 наблюдается существенный прирост в производительности метода. А вот при смене размера истории с 50 до 100 почти ничего не меняется, даже в какой-то момент L-BFGS с размером истории 50 обгоняет своего старшего брата по количеству итераций. В итоге, в реальной задаче я бы использовал такие типичные значение размера истории, как 10-50.

3 Сравнение методов на реальной задаче логистической регрессии

В этом эксперименте нужно сравнить усеченный метод Ньютона, L-BFGS и градиентный спуск на задаче обучения логистической регрессии на реальных данных. В качестве реальных данных будут использоваться следующие наборы: w8a, gisette, real-sim, news20.binary и rcv1.binary (его test версия на 400+ Мб). Обозначим количество фичей и векторов за за n и m соответственно. Коэффициент регуляризации взят стандартным образом: $\lambda = \frac{1}{m}$.
Начальной точкой в каждом эксперименте является $\overrightarrow{x_0} = \overrightarrow{0}$.

3.1 Теория

Сперва поговорим про задачу логистической регрессии, которую нам нужно было реализовать чисто векторно. Функцией потерь здесь будет являтся:

$$L(x) = \frac{1}{m} \cdot 1_m \cdot log(1 + exp(-b \odot (A \cdot x)) + \frac{\lambda}{2}||x^2||$$

где 1_m является вектором строчкой из единиц размерности количества точек в датасете, вектор b является вектором-столбцом классов, а $A_{m,n}$ - матрицей признаков, где вектора расположены в строчках. Посчитаем ее градиент:

$$\nabla L(x) = -\frac{1}{m} \cdot A^T \cdot (b \odot \left\{ \frac{1}{1 + exp(b \odot (A \cdot x))} \right\}) + \lambda x$$

А теперь и Гессиан (подсмотрев в Yandex Handbook):

$$\nabla^2 L(x) = \frac{1}{m} \cdot A^T \cdot S \cdot A + \lambda \cdot I_n$$

где матрица $S_{m,m}$ является диагональной матрицей с элементами:

$$S = diag \left\{ \frac{1}{1 + exp(b \odot (A \cdot x))} \odot (1 - \frac{1}{1 + exp(b \odot (A \cdot x))}) \right\}$$

3.2 Проведение экспериментов

По заданию, точность метода Ньютона и L-BFGS была выставлена $1e-9$, а градиентного спуска - $1e-5$, так как он дольше сходится. Для каждого датасета считались количество итераций метода, нормы градиента на каждом шаге, значение функции потерь и время, затраченное с начала запуска. Во всех методах в качестве линейного поиска был выбран метод Вульфа с константами $c_1 = 1e-4, c_2 = 0.9$.

3.2.1 W8a dataset

Сам по себе датасет маленький, считается быстро. В итоге получились следующие зависимости:

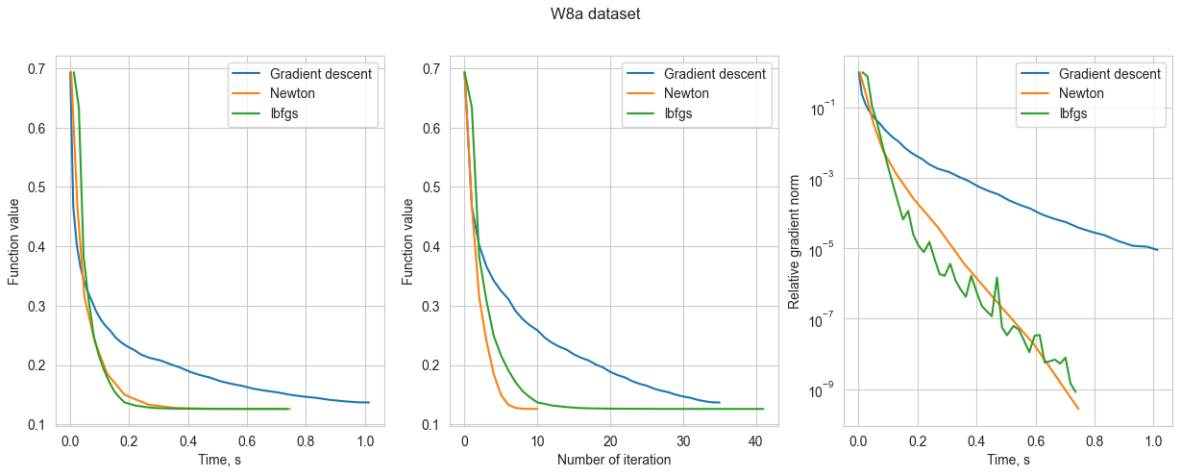

Рис. 3: Зависимость значения функции потерь в линейном масштабе от времени (слева); Зависимость значения функции потерь в линейном масштабе от количества итераций (по-середине); Зависимость относительного квадрата нормы градиента от времени в логарифмическом масштабе (справа)

По результатам видно, что усеченный метод Ньютона выигрывает на этом датасете по всем параметрам, а с небольшим отрывом идет lbfgs. Видно, что квадрат относительной нормы градиента осциллирует для метода lbfgs, в то время как усеченный метод Ньютона и градиентный спуск ведут себя стабильно.

3.2.2 Gisette dataset

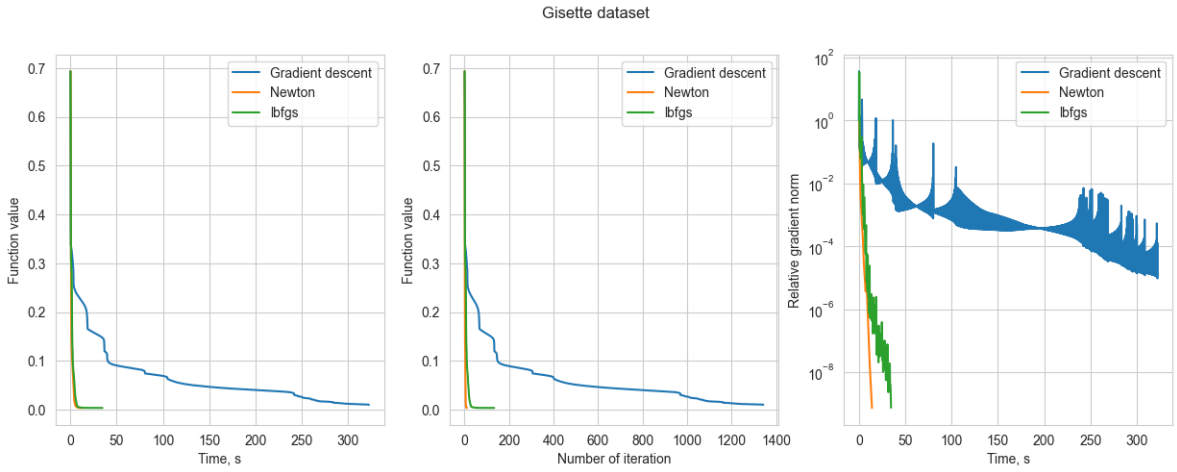

Рис. 4: Зависимость значения функции потерь в линейном масштабе от времени (слева); Зависимость значения функции потерь в линейном масштабе от количества итераций (по-середине); Зависимость относительного квадрата нормы градиента от времени в логарифмическом масштабе (справа)

Не перестану повторять, что усеченный метод Ньютона - чемпион как по итерациям, так и по скорости вычисления! В остальном, ситуация схожа с прошлым датасетом.

3.2.3 Real-sim dataset

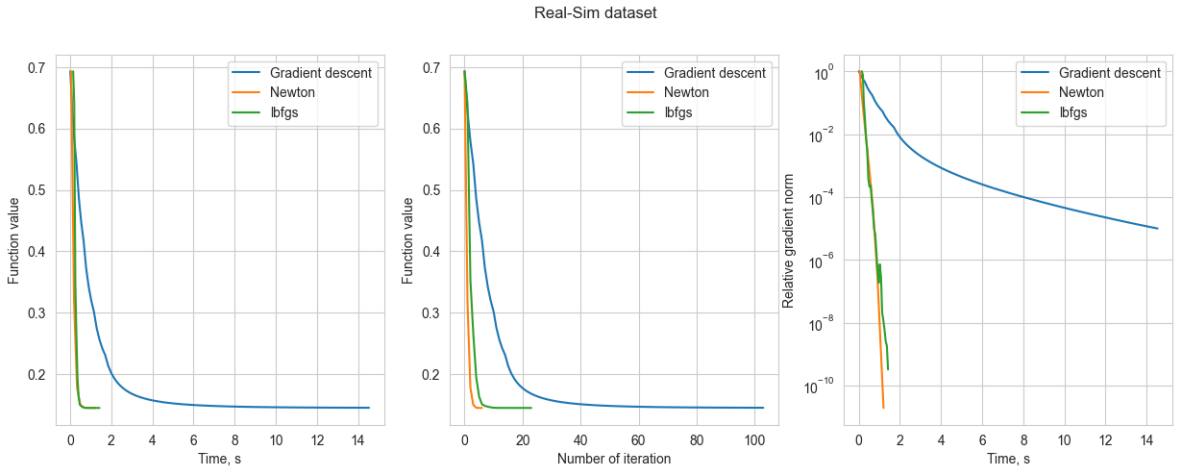

Рис. 5: Зависимость значения функции потерь в линейном масштабе от времени (слева); Зависимость значения функции потерь в линейном масштабе от количества итераций (по-середине); Зависимость относительного квадрата нормы градиента от времени в логарифмическом масштабе (справа)

3.2.4 news20.binary

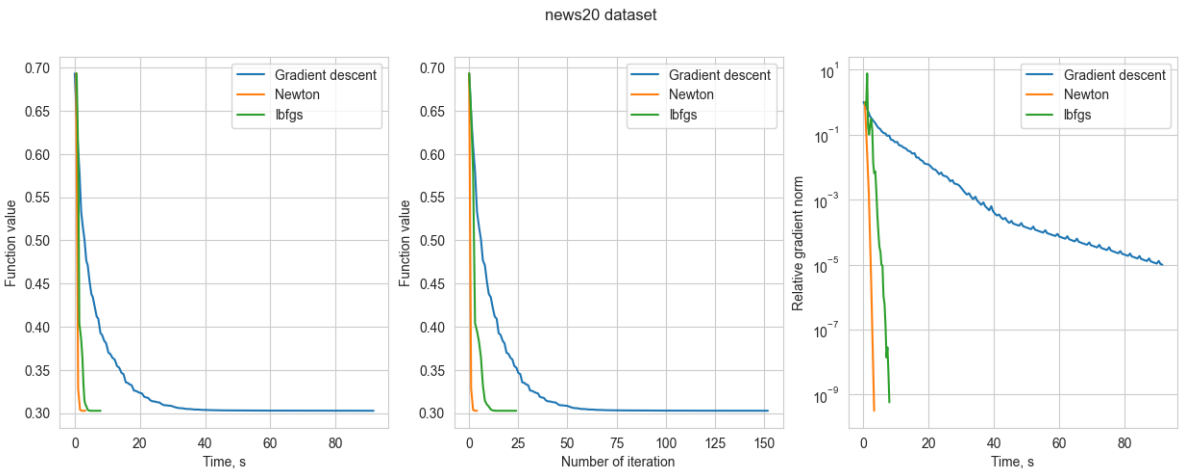

Рис. 6: Зависимость значения функции потерь в линейном масштабе от времени (слева); Зависимость значения функции потерь в линейном масштабе от количества итераций (по-середине); Зависимость относительного квадрата нормы градиента от времени в логарифмическом масштабе (справа)

3.2.5 rcv1.binary

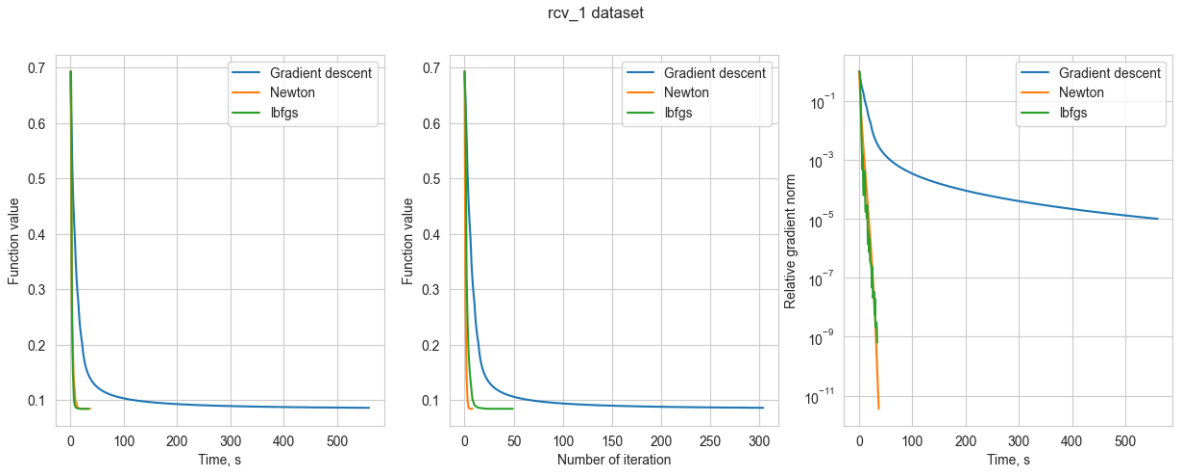

Рис. 7: Зависимость значения функции потерь в линейном масштабе от времени (слева); Зависимость значения функции потерь в линейном масштабе от количества итераций (по-середине); Зависимость относительного квадрата нормы градиента от времени в логарифмическом масштабе (справа)

Трейн версия этого датасета оказалось самой прожорливой как по памяти, так и по времени вычислений... Градиентный спуск для таких датасетов - вообще не вариант, но, при сравнении, можно увидеть, что до приемлемой точности он сойдется лет через 100!

3.3 Выводы

Для трех датасетов была обучена логистическая регрессия с тремя различным методами оптимизации. Видно, что более простой метод (GD) проигрывает как по количеству итераций, так и по скорости сходимости к точке оптимума. Его старшие методы, которые имеют представление о гессиане оптимизируемой функции, довольно похожи между собой, хоть усеченный метод Ньютона и выигрывает по скорости вычисления.

Какой из методов лучше и в каких ситуациях?

В прошлой лабораторной было не так очевидно, каким методом пользоваться для оптимизации оракула. На некоторых градиентный спуск с постоянным шагом показывал себя чудесно, на каких-то - обычный метод Ньютона. Но

здесь фаворитом стал усеченный метод Ньютона. Сравнивая результаты с Андреем Широбоковым, увидел, что у него реализация lbfgs выигрывает по времени выполнения усеченный метод Ньютона, так что немного переформулирую свое заключение: в любой ситуации стоит выбирать между lbfgs и усеченным Ньютоном, градиентный спуск - уже в прошлом!