Методы оптимизации, лабораторная работа 3

Миляуша Сабирова

1 Эксперимент: Выбор длины шага в субградиентном методе

Тут я сгенерировала рандомную матрицу A размера 10×5 и вектор b размерности 5. В качестве коэффициента регуляризации взяла 1.

Начальные точки сгенерированы следующим образом (нумерация такая же как на графиках):
1) np.zeros(5)

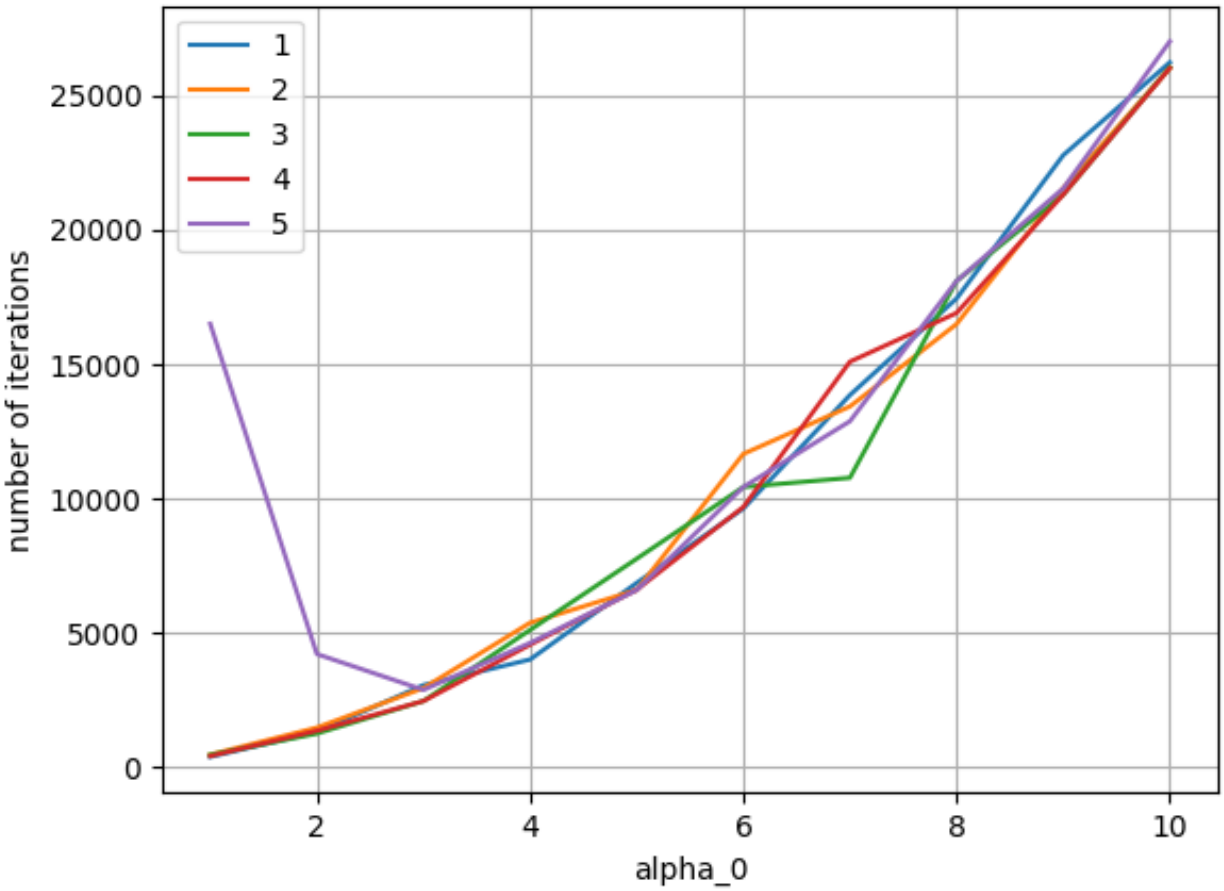
2) np.random.rand(5)

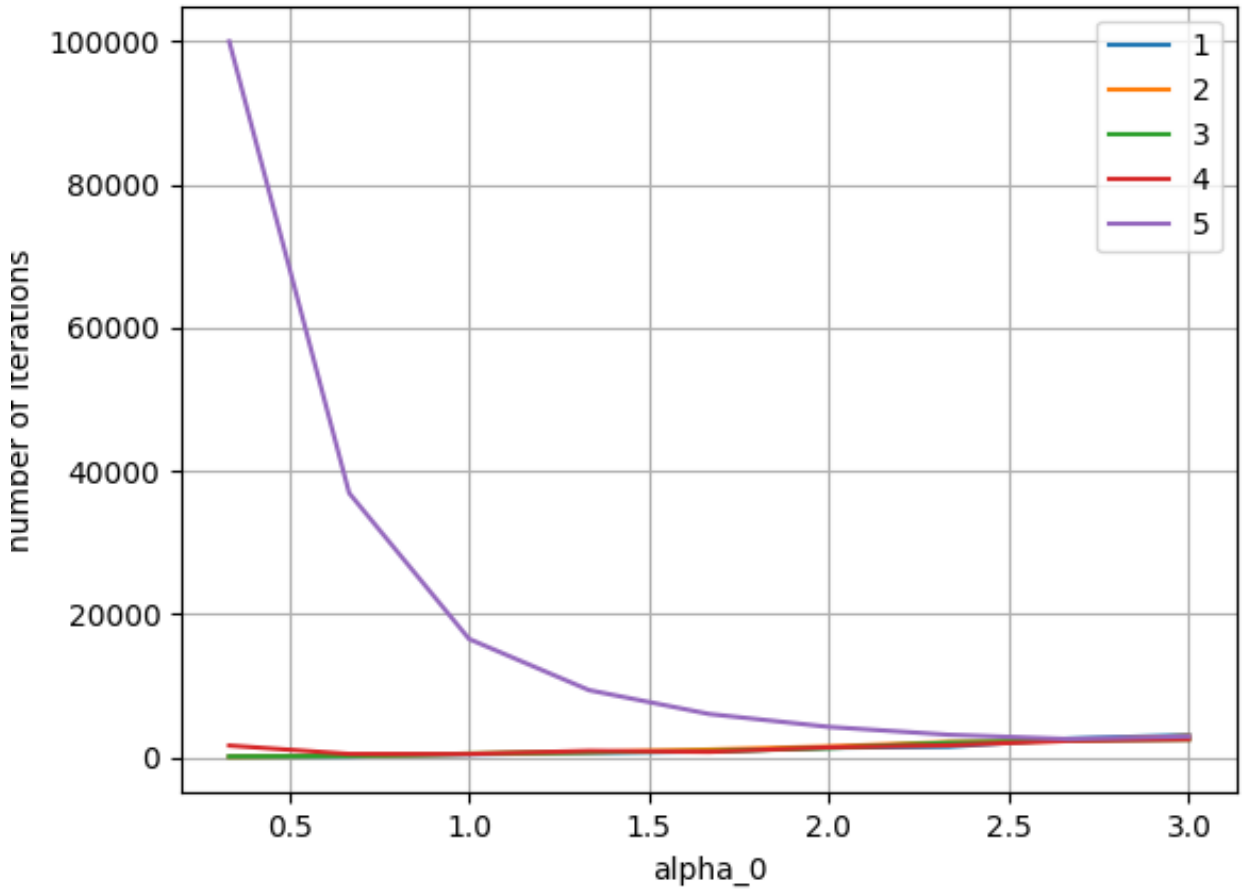
3) np.ones(5)

4) np.ones(5) * 10,

5) np.ones(5) * 100,

То есть в 4) и 5) пунктах начальные точки расположены значительно дальше от точек, указанных в 1-3 пунктах


1
2
3
4
5
25000
20000
15000
10000
5000
0
number of iterations
2
4
6
8
10
alpha_0



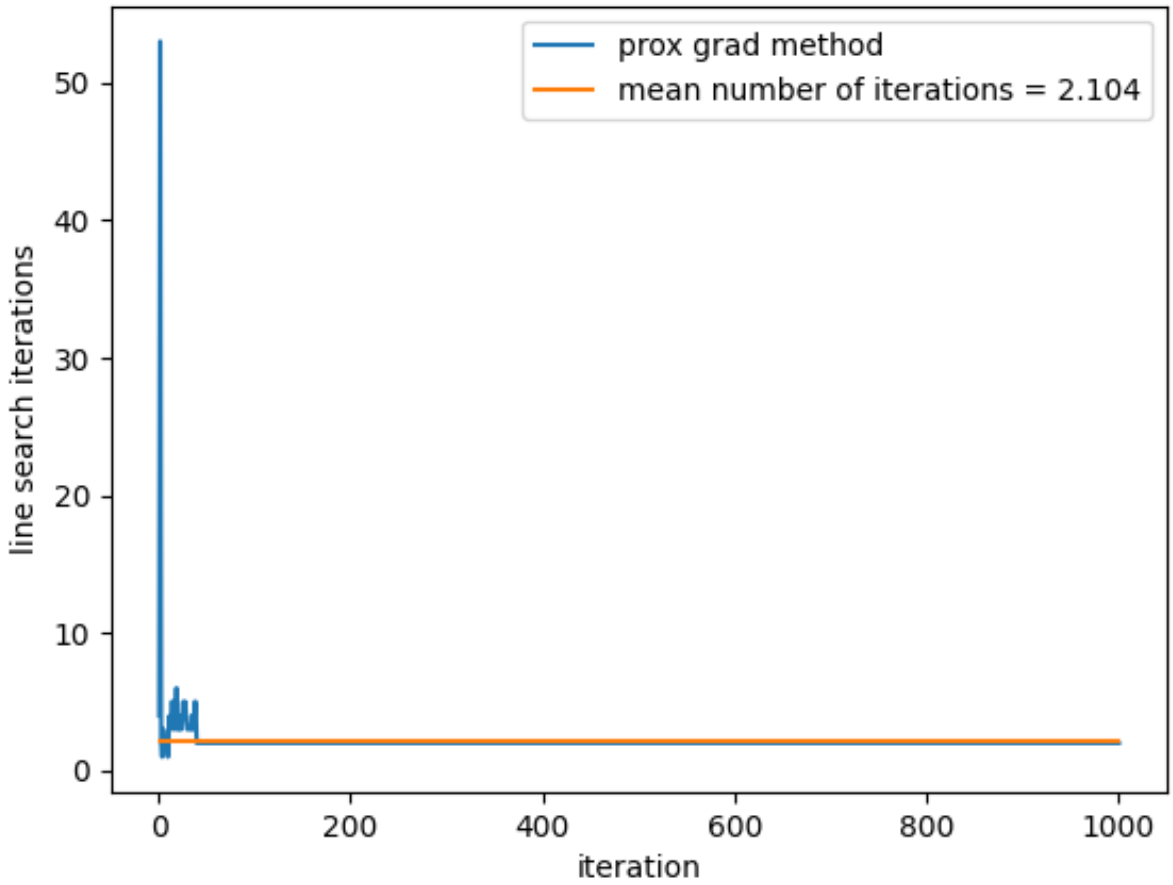
На этих графиках видно, что при $\alpha_0 >= 3$ графики для всех начальных точек ведут себя одинаково и количество итераций метода не зависит от выбора начальной точки. Видимо, на первых итерациях алгоритма из-за больших шагов алгоритма мы будем попадать в случайные точки области определения функции и только с какого-то момента будем находиться в районе искомого минимума функции, поэтому количество итераций растет с ростом α_0

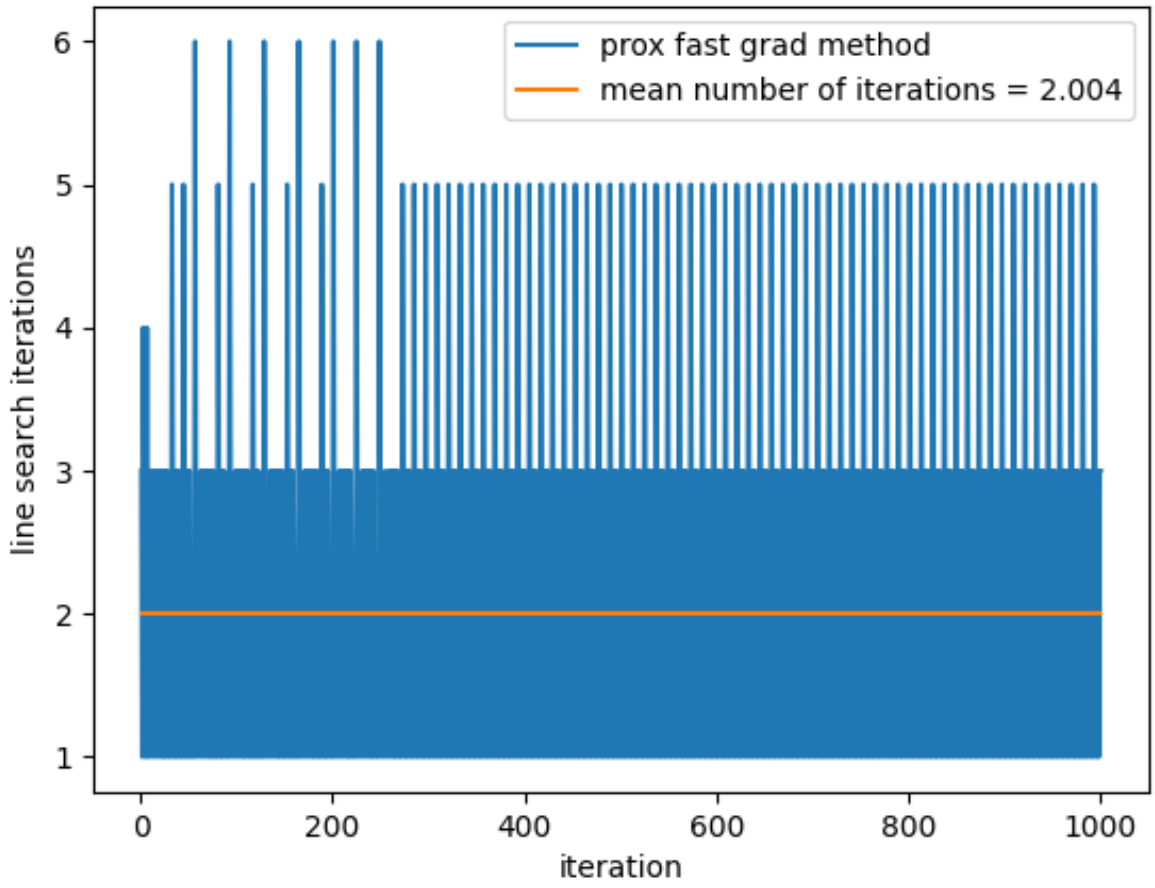
При $\alpha_0 <= 3$ выделяется только график для последней начальной точки, у которой самые большие по модулю координаты. Тут видно, что до некоторого момента увеличение α_0 помогает значительно уменьшить количество итераций алгоритма, видимо, потому что большие шаги на первых итерациях алгоритма помогают приблизиться к области, в которой стоит искать минимум функции. Для других точек такой закономерности не наблюдается, потому что они изначально находятся близко к искомой точке. Таким образом, связь между наилучшим коэффициентом α_0 и начальной точкой, наблюдается только в случае, когда начальная точка находится очень далеко от предполагаемого минимума функции. На практике же, если мы выберем случайную начальную точку, мы скорее всего попадем в ситуацию, когда выбор этой точки не будет влиять на то, какой коэффициент α_0 нам стоит выбрать, чтобы алгоритм сошелся быстрее

2 Эксперимент: Среднее число итераций одномерного поиска в градиентных методах

Данные генерировала так же как в предыдущем эксперименте.

Ниже представлены графики для обычного градиентного метода и ускоренного градиентного метода. В первом случае среднее число итераций линейного поиска получилось равным 2.104, во втором - 2.004, что подтверждает теоретическую оценку на среднее число итераций линейного поиска на каждом шаге.





3 Эксперимент: Сравнение методов

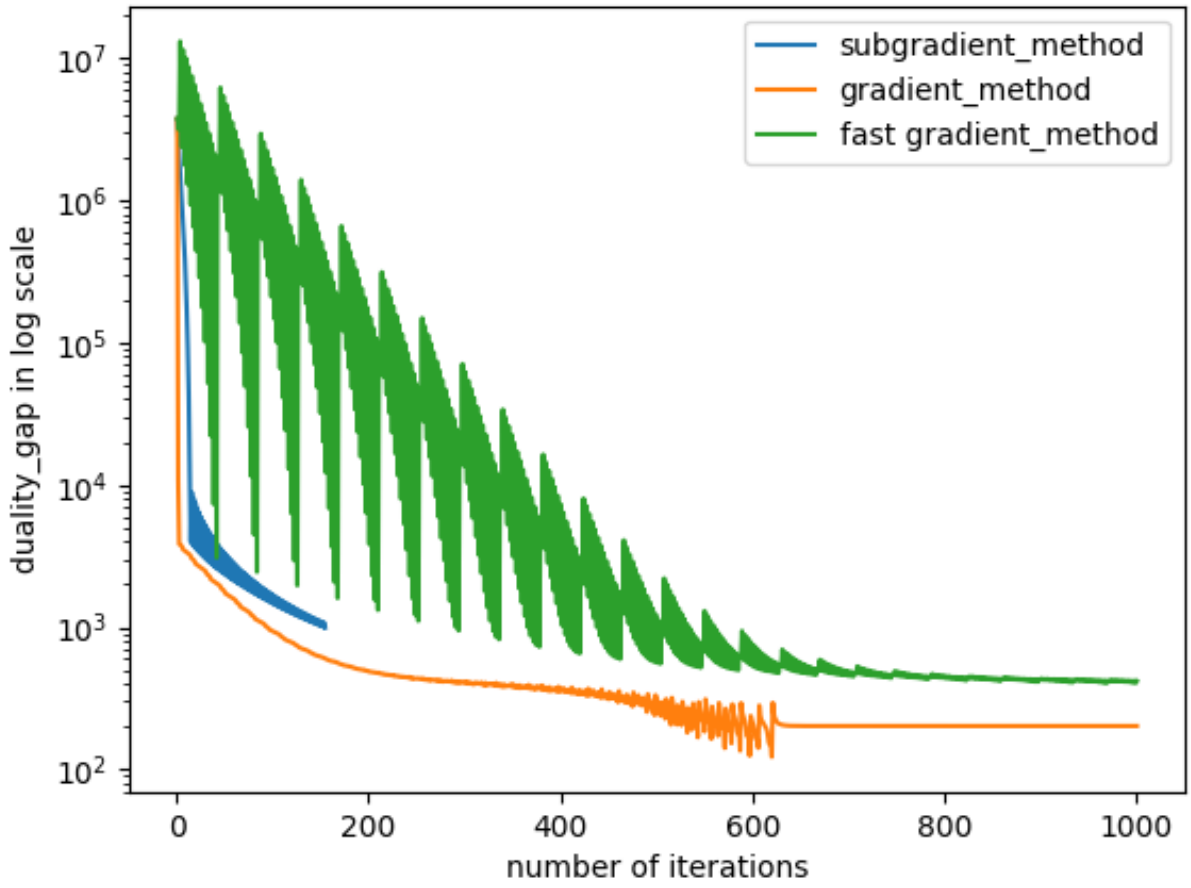
Я перебрала указанные ниже значения размерности пространства, размера выборки и коэффициента регуляризации и построила графики для всех комбинаций (все их можно увидеть в репозитории). Данные генерировала случайным образом.

n : $10, 100, 1000$

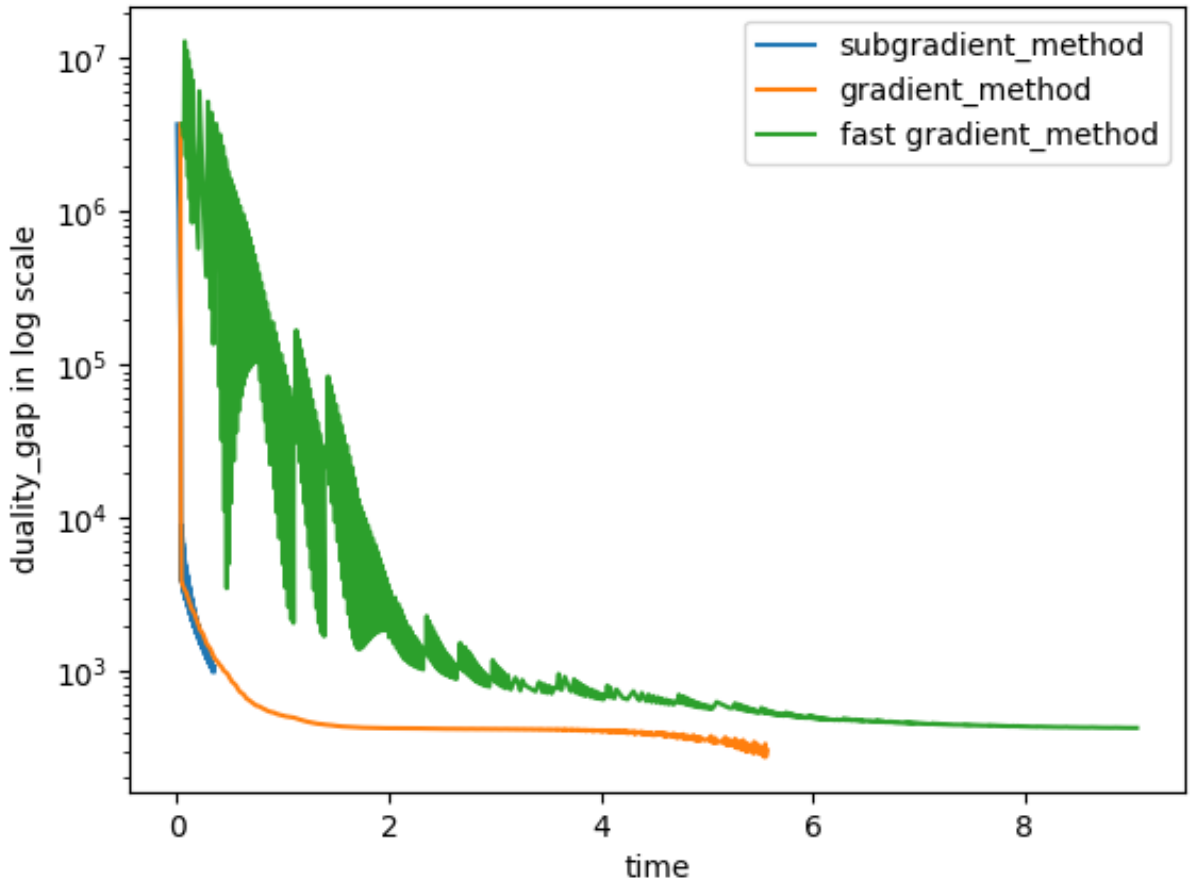
m : $100, 1000, 10000$

λ : $0.01, 0.1, 1, 10$

Например, вот какой график получился при $n = 100, m = 10000, \lambda = 1$



А вот такой для $\lambda = 0.01$ (я не стала вставлять в отчет остальные, потому что их получилось слишком много)



У меня получилось, что на практике субградиентный метод в среднем требует меньше итераций и затраченнного времени, чем оба градиентных. Можно также заметить, что в большинстве ситуаций обычный градиентный метод дает точность по зазору двойственности лучше, чем быстрый градиентный метод, и требует меньше общего число итераций. К тому же оба градиентных метода в среднем на каждой итерации делают по 2 итерации линейного поиска.