

Методы оптимизации.
Отчет по лабораторной работе 1.
Новицкий Илья, Остриченко Илья, Петров Георгий.
М3234

19.03.2024

Ссылка на GitHub : https://github.com/Saintghetto17/Optimization_Lab_1

Постановка задачи: найти минимальное значение функции $f : \mathbb{R}^n \rightarrow \mathbb{R}$, используя методы 0-1 порядков.

В нашем случае пришлось работать с квадратичными функциями при $n = 2$. В качестве исследуемых функций мы взяли:

- (1) $f_1(x, y) = x^2 + (2x - 4y)^2 + (x - 5)^2$. Точка минимума $(\frac{5}{2}, \frac{5}{4}, \frac{25}{2})$,
- (2) $f_2(x, y) = x^2 + y^2 - xy + 2x - 4y + 3$. Точка минимума $(2, 0, -1)$,
- (3) Bonus: $f_3(x, y) = 100x^2 + y^2$. Точка минимума $(0, 0, 0)$.

Каждую из функций мы протестировали на двух методах градиентного спуска, а именно: методе с постоянным шагом и методе одномерного поиска. А также на методе Нельдера-Мида, который заключается в том, что поиск минимума происходит за счет отражений, сжатий и растяжений симплекса. Каждый из методов имеет свои собственные преимущества. Например, градиентные методы являются представителями метода 1 порядка, это значит алгоритм поиска использует производную первого порядка исследуемой функции, в то время как Нельдер-Мид является методом 0 порядка.

Nota bene : мы обрезаем часть таблицы (последние две колонки, так как они не сильно влияли на результат. Их смысл заключался в критерии остановки метода)

1. Градиентные методы. Постоянный шаг. Реализация алгоритма градиентного спуска на наших функциях работает ожидаемо корректно. Рассмотрим таблицы, в которых отражена работа метода с постоянным шагом.

В качестве возможных постоянных шагов мы взяли 0.1, 0.001. Заметно, что выбор шага очень влияет на конечный результат. Так, в 0-3-6 экспериментах для f_1 минимальное значение так и не достигается, а для бонусной плохо-обусловленной функции 21-24 выявляют тот же результат.

С чем это связано? Как известно, следующий шаг градиентного метода вычисляется по формуле $x_{k+1} = x_k - \alpha \cdot grad(x_k)$. При этом, может случиться такое, что прыжки по вектору градиента устремят точку "вникуда" или просто заиклят. Обычно такое происходит, когда выбранный коэффициент недостаточно мал. Заметно, что, для f_1 количество итераций наиболее мало, по сравнению с двумя оставшимися. Эта характеристика ни в коем случае не означает, что алгоритм работает плохо для оставшихся функций, так как помимо самого алгоритма важен и выбор начальной точки. В случае с f_1 нам можно сказать повезло. Из таблицы видно, что метод достаточно точно находит минимальное значение. Различие буквально порядка 10^{-4} .

table1.png

Рис. 1: Таблица для f_1

table2.png

Рис. 2: Таблица для f_2

table3.png

Рис. 3: Таблица для f_3

Тем не менее такой метод достаточно не конструктивен. Сразу возникает вопрос, а как мы выбрали шаг, который выдаст корректный результат, как мы его подобрали, и что, если для разных функций шаг может варьироваться? Именно поэтому и существует метод градиентного спуска, основанный на одномерном спуске, или как его еще называют "метод наискорейшего спуска".

1. Градиентные методы. Метод наискорейшего спуска..

В чем заключается метод? Будем конструктивно искать такое α , при котором $f(x_k - \alpha \cdot \text{grad}(x_k))$ принимает минимальное значение по α . Заметим, что теперь наша функция является функцией одного аргумента, а также является унимодальной (без док-ва). На помощь приходят методы тернарного спуска и дихото-

мии (бонус 1). Оба метода сужают область поиска на какую-то фиксированную константу. Такой подход поиска шага более универсален, а засчет правильного подбора мы уменьшим количество итераций алгоритма.

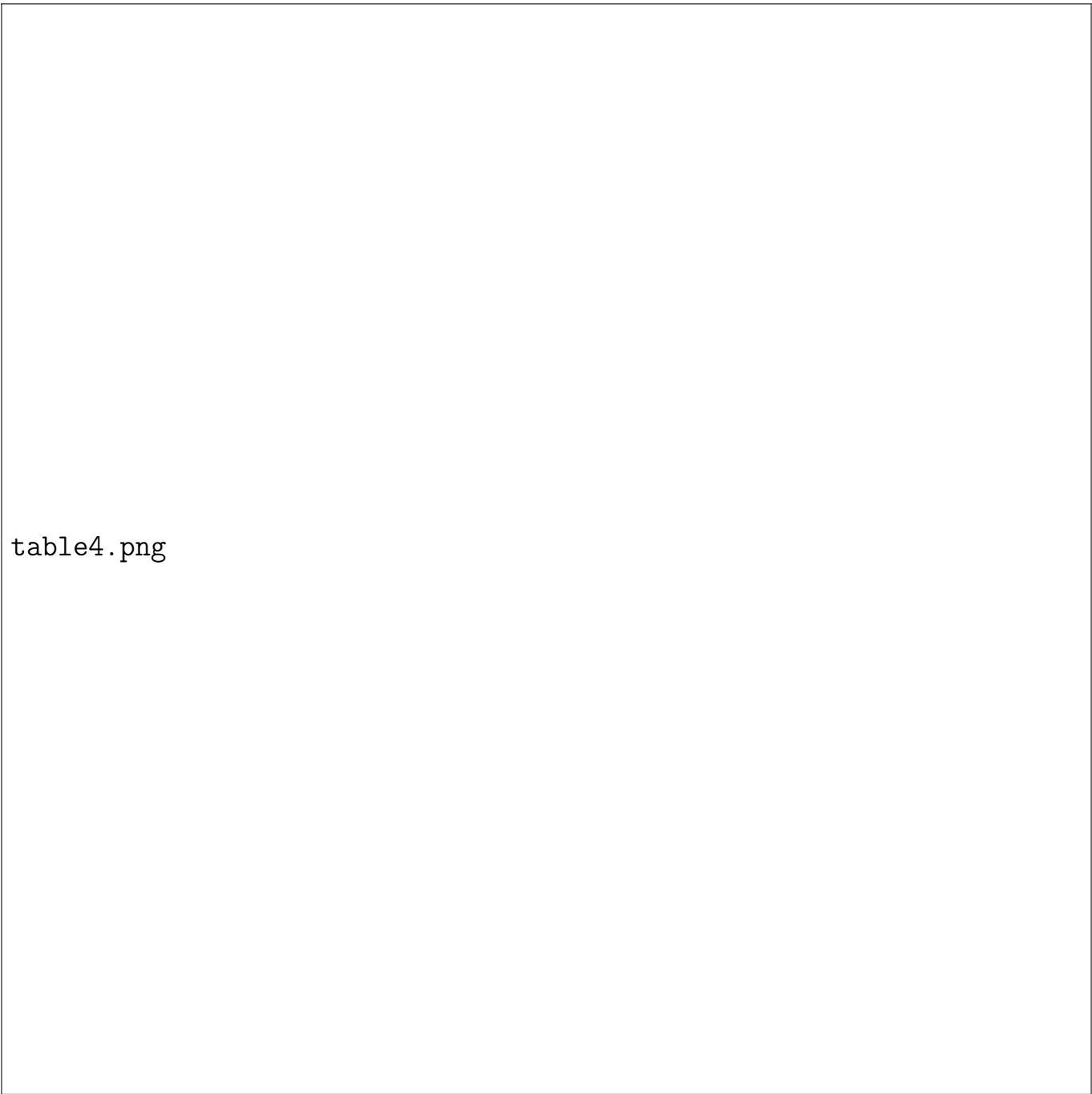


Рис. 4: Таблица для f_1 . Тернарный поиск

table6.png

Рис. 5: Таблица для f_2 . Тернарный поиск

table5.png

Рис. 6: Таблица для f_3 . Тернарный поиск

dichotomy1.png

Рис. 7: Таблица для f_1 . Бонус 1 - дихотомия

dichotomy2.png

Рис. 8: Таблица для f_2 . Бонус 1 - дихотомия

dihotomy3.png


Рис. 9: Таблица для f_3 . Бонус 1 - дихотомия

Из таблицы видно, что количество итераций уменьшилось аж в 100 раз. И это неудивительно, потому что каждый раз мы выбираем лучший шаг для данной точки и функции. Конечно, внутри мы крутимся внутри функции поиска, но так или иначе, количество итераций уменьшается, а точность значений увеличивается до 10^{-8} порядка. Подряд идут фотографии таблиц для тернарного поиска и дихотомии, так как идейно они не сильно отличаются.

Нельдер-Мид.

Как уже было сказано ранее, Нельдер Мид принципиально другой метод с точки зрения идеи. Основная задача все такая же - поиск минимального значения функции. Но данный метод является методом 0 порядка, а значит нам не нужно

брать производную функции. Иногда этот метод довольно удобен если мы хотим исследовать не гладкую функцию. Метод устроен примерно так: берем в n -мерном пространстве симплекс размера $n+1$, сортируем значения функции в точках симплекса, берем две точки, в которых достигаются максимальные значения функции соответственно, а также точку, где достигается минимальное значение. Считаем центр тяжести $n+1$ точки и в зависимости от некоторых условий отражаем, сжимаем или растягиваем наш симплекс. Но есть и минусы у этого метода. При выборе "плохих" точек симплекса мы можем убежать дальше допустимой окрестности поиска минимума. Рассмотрим таблицы, относящиеся к Нельдеру-Миду.



nelder.png

Рис. 10: Общая таблица по Нельдеру-Миду

Заметно, что библиотичная реализация этой функции в SciPy достаточно эффек-

тивна и точна. Количество итераций чуть больше, чем количество итераций в спуске с одномерным поиском, но значения точны до 10^{-8} порядка. Причем видно, что метод достаточно эффективен и для плохо-обусловленной функции.

Функция n аргументов. Бонус 2.1

Мы считаем, что данный бонус ничем не отличается от рассмотренных примеров. Идейно в код нужно добавить точки произвольной размерности и посчитать значения функции с учетом размерности. Код был написан изначально для функции двух аргументов, а принципиально что-то менять для повышения размерности - долго и не представляет научного интереса.

Плохо обусловленная функция. Бонус 2.2

В обоих методах для f_3 результат в 23 и 26 экспериментах (для постоянного шага) и 21 - 26 экспериментах (для наискорейшего спуска) сильно отличается от желаемого. Дело в том, что данная функция является плохо обусловленной, а именно : при малом изменении нормы аргумента, значения функции меняются сильно. Это затрудняет выдачу точных значений для градиентных методов. В таком случае лучше использовать Нельдера-Мида.

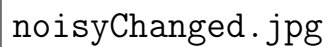
Зашумленная функция и Мультимодальная функция. Бонус 2.3

В чем смысл зашумленной функции? Смысл в том, что "поставщик" значений может прислать значения функции с немного некорректными значениями. В общем виде это выглядит так $f \rightarrow f + random(0, EPS)$.

При этом поставщик значений ожидает, что минимальное значение при зашумлении не сильно будет отличаться от минимального значения оригинальной функции. Так вышло, что зашумление достаточно сильно влияет на поиск минимального значения. Это неумудрено, так как сама функция становится недифференцируемой и неявной. Рассмотрим таблицы для постоянного шага и для изменяемого.

noisyConstant.jpg

Рис. 11: Зашумленная функция. Постоянный шаг. Бонус 2.3



noisyChanged.jpg

Рис. 12: Зашумленная функция. Изменяемый шаг. Бонус 2.3

Видно, что количество итераций может достигать больших значений. Да и вообще, нельзя рассчитывать на хоть сколько-то корректные значения с неявной и недифференцируемой функцией.

Рассмотрим мультимодальную функцию. В качестве примера мы взяли функцию Экли. В чем сложность работы с мультимодальными функциями? Мультимодальная функция — это функция с более чем одной “модой” или оптимумом (например долиной на графике). А значит, градиентный метод может забрести не в ту впадину и мы получим не тот ответ, который ожидаем. Мы ожидаем в качестве


минимального значения выбранной функции 0. Рассмотрим таблицы.

eckleylearning.jpg

Рис. 13: Функция Экли. Постоянный шаг. Бонус 2.3

eckleychanging.jpg

Рис. 14: Функция Экли. Тернарный поиск. Бонус 2.3



eckleyDih.jpg

Рис. 15: Функция Экли. Дихотомия. Бонус 2.3

Видно, что метод тернарного поиска для этой мультимодальной функции преуспел. Из этого следует, что для мультимодальных методов градиентный метод не подходит в силу нескольких оптимумов.

Картинки:

gradient.png

Рис. 16: Блуждания градиента

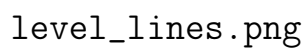
The image is a placeholder for a file named 'level_lines.png'. It is represented by the text 'level_lines.png' in a monospaced font, which is a common way to denote a missing image in a document.

Рис. 17: Функция и линии уровня

Итог:

В ходе лабораторной работы мы узнали много удивительных фактов, а также поработали с разными типами функций, численно нашли значение минимумов и оценили отклонения.