

Методы оптимизации. 1 Лабораторная

Постановка задачи

1. Реализуйте градиентный спуск с постоянным шагом (learning rate).
2. Реализуйте метод одномерного поиска (метод дихотомии, метод Фибоначчи, метод золотого сечения) и градиентный спуск на его основе.
3. Проанализируйте траекторию градиентного спуска на примере квадратичных функций. Для этого придумайте две-три квадратичные функции от двух переменных, на которых работа методов будет отличаться.
4. Для каждой функции:
 - (a) исследуйте сходимость градиентного спуска с постоянным шагом, сравните полученные результаты для выбранных функций;
 - (b) сравните эффективность градиентного спуска с использованием одномерного поиска с точки зрения количества вычислений минимизируемой функции и ее градиентов;
 - (c) исследуйте работу методов в зависимости от выбора начальной точки;
 - (d) исследуйте влияние нормализации (scaling) на сходимость на примере масштабирования осей плохо обусловленной функции;
 - (e) в каждом случае нарисуйте графики с линиями уровня и траекториями методов;
5. Реализуйте генератор случайных квадратичных функций n переменных с числом обусловленности k .
6. Исследуйте зависимость числа итераций $T(n, k)$, необходимых градиентному спуску для сходимости в зависимости от размерности пространства $2 \leq n \leq 1000$ и числа обусловленности оптимизируемой функции $1 \leq k \leq 1000$.
7. Для получения более корректных результатов проведите множественный эксперимент и усредните полученные значения числа итераций.

Дополнительное задание

Реализуйте одномерный поиск с учетом условий Вольфе и исследуйте его эффективность. Сравните полученные результаты с реализованными ранее методами.

1) Градиентный спуск - это метод нахождения локального минимума (максимума) функции.

$f: \mathbb{R}^n \rightarrow \mathbb{R}$

Алгоритм.

- 1) Выбрать начальное приближение - точку r_0
- 2) $r_{i+1} = r_i - \alpha \cdot \text{grad}(f)(r_i)$, α - шаг спуска

2) Градиентный спуск на основе одномерного поиска - тот же градиентный спуск, только шаг α берется не постоянный, а на каждой итерации берется наилучший. Для подсчета шага α я использовал метод золотого сечения.

Метод золотого сечения - метод одномерного поиска локального минимума (максимума). В основе метода лежит принцип деления отрезка в пропорциях золотого сечения

3) Проанализируем работу методов на примере двух квадратичных функций от двух переменных.

1 функция

$$x^2 + 3y^2 - x - 4y$$

$$x_{\min} = 1/2, y_{\min} = 1/3$$

4) а) Градиентный спуск будет сходиться при $\alpha \leq 0.333$. Значение было вычислено с помощью бинарного поиска при разных начальных приближениях. При $\alpha = 0.25$ градиентному спуску с постоянным шагом понадобится меньше всего итераций.

б) Сравним эффективность градиентного спуска с $\alpha = 0.25$ с градиентным спуском с использованием одномерного поиска.

Градиентный спуск с постоянным шагом:

Count of gradient countings: 26

Count of function evaluations: 52

Градиентный спуск с одномерным поиском:

Count of gradient countings: 18

Count of function evaluations: 1260

Видно, что градиентный спуск с постоянным шагом выгоднее с точки зрения количества вычислений функции, но хуже с точки зрения количества вычислений градиента.

с) Градиентный спуск с постоянным шагом:

$$x_0 = 1000000000000, y_0 = 1000000000000$$

Count of gradient countings: 65

Count of function evaluations: 130

$$x_0 = 2, y_0 = 2$$

Count of gradient countings: 26
Count of function evaluations: 52

Градиентный спуск с одномерным поиском:

$x_0 = 10000000000000$, $y_0 = 10000000000000$

Count of gradient countings: 41

Count of function evaluations: 2870

$x_0 = 2$, $y_0 = 2$

Count of gradient countings: 18

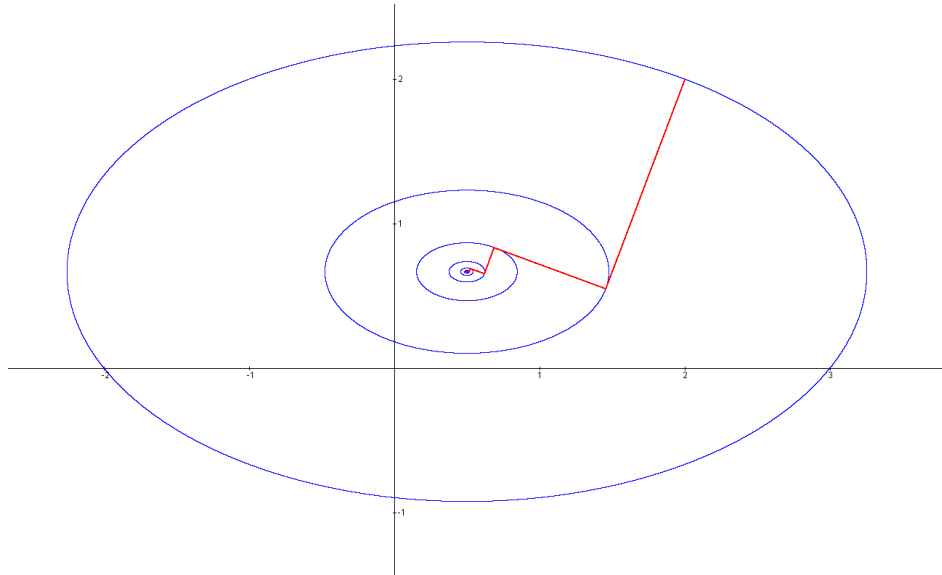
Count of function evaluations: 1260

Видно, что оба метода очень слабо зависят от выбора начального приближения.

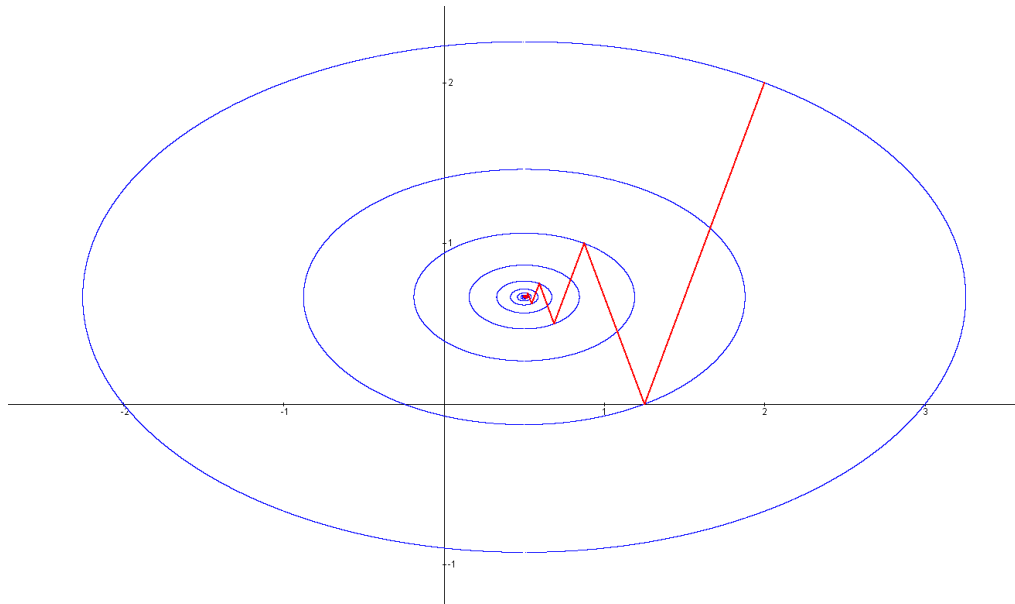
d) Исследуем влияние масштабирования на сходимость градиентного спуска с использованием одномерного поиска. Вместо $f(x,y)$ будем градиентным спуском проходить по функции $f(ax,y)$.

при $a \rightarrow 0$ градиентный спуск будет сходиться, но чем ближе a к 0, тем больше понадобится итераций, при $|a| > 4875$ градиентный спуск перестает сходиться. Если спускаться по $f(x,ay)$, то при $a \rightarrow 0$ градиентный спуск будет сходиться, но чем ближе a к 0, тем больше понадобится итераций, при $|a| > 2819$ градиентный спуск перестает сходиться.

e) α ищется методом золотого сечения:



alpha = 0.25:



2 функция

$$x^2 + y^2 + xy - 3y + 2x + 4$$

$$x_{\min} = -7/3 \quad y_{\min} = 8/3$$

а) Градиентный спуск будет сходиться при $\alpha \leq 0.666$. Значение было вычислено с помощью бинарного поиска при разных начальных приближениях. При $\alpha = 0.48$ градиентному спуску с постоянным шагом понадобится меньше всего итераций.

б) Сравним эффективность градиентного спуска с $\alpha = 0.48$ с градиентным спуском с использованием одномерного поиска.

Градиентный спуск с постоянным шагом:

Count of gradient countings: 26

Count of function evaluations: 52

Градиентный спуск с одномерным поиском:

Count of gradient countings: 20

Count of function evaluations: 1400

Видно, что градиентный спуск с постоянным шагом выгоднее с точки зрения количества вычислений функции, но хуже с точки зрения количества вычислений градиента.

с) Градиентный спуск с постоянным шагом:

$x_0 = 1000000000000$, $y_0 = 1000000000000$

Count of gradient countings: 55

Count of function evaluations: 110

$x_0 = 2, y_0 = 2$

Count of gradient countings: 26

Count of function evaluations: 52

Градиентный спуск с одномерным поиском:

$x_0 = 1000000000000, y_0 = 1000000000000$

Count of gradient countings: 7

Count of function evaluations: 490

$x_0 = 2, y_0 = 2$

Count of gradient countings: 20

Count of function evaluations: 1400

Видно, что оба метода очень слабо зависят от выбора начального приближения.

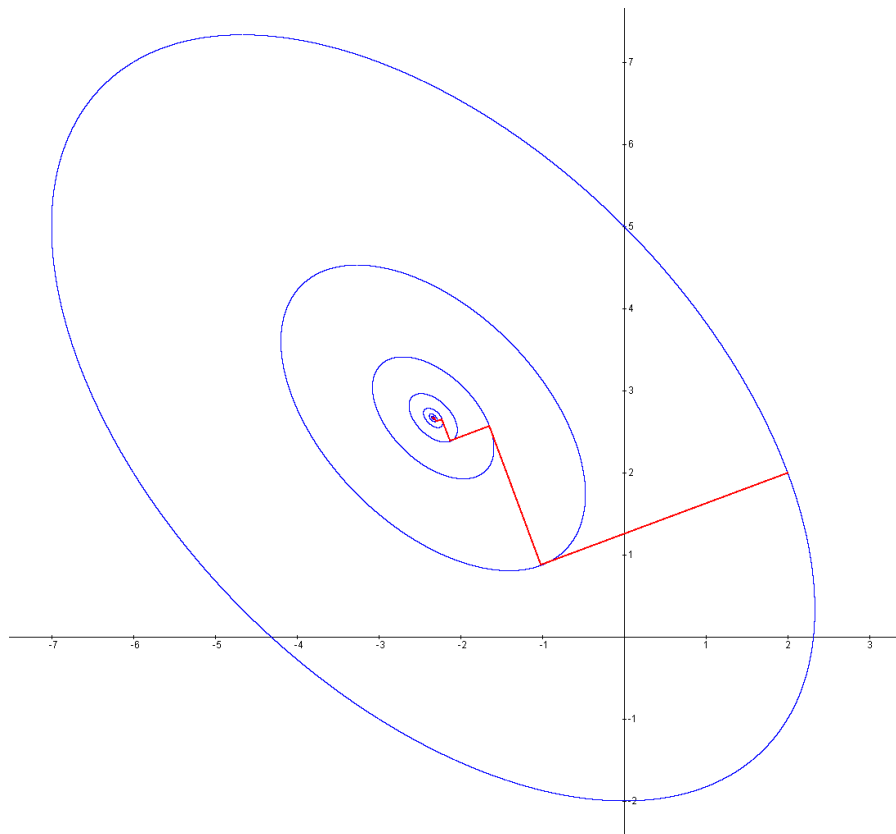
Что показалось мне необычным, так это то, что градиентный спуск с одномерным поиском, нашел минимум гораздо быстрее при начальном приближении

$x_0 = 1000000000000, y_0 = 1000000000000$.

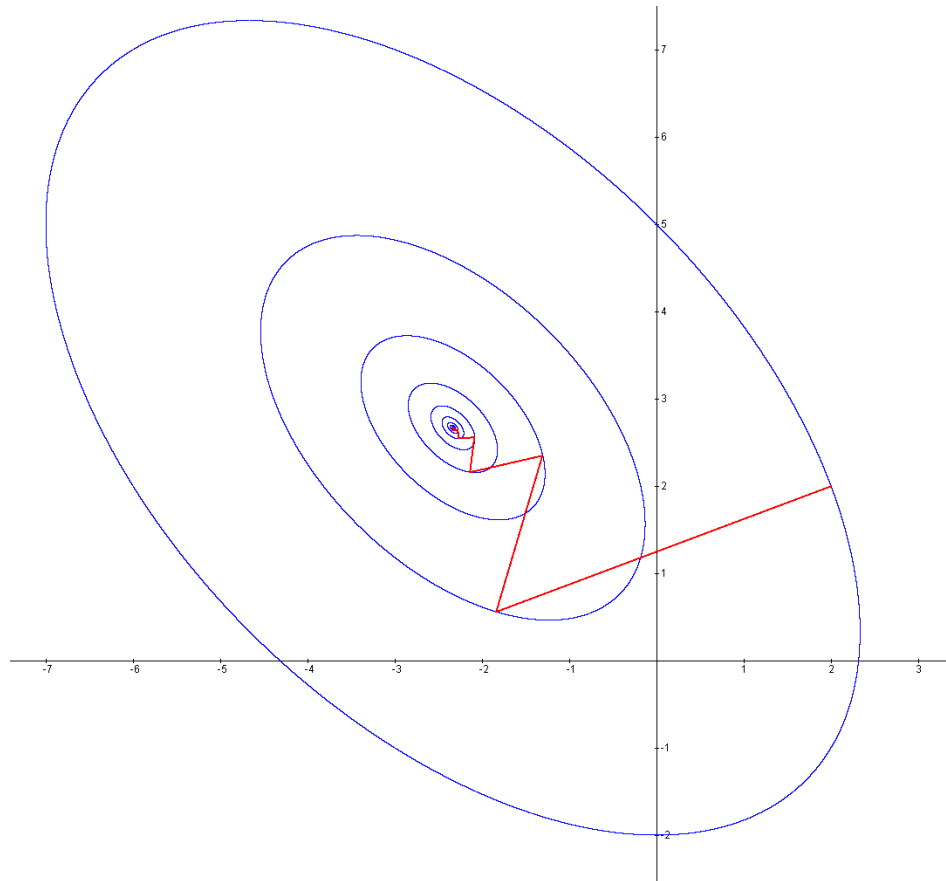
d) Исследуем влияние масштабирования на сходимость градиентного спуска с использованием одномерного поиска. Вместо $f(x,y)$ будем градиентным спуском проходить по функции $f(ax,y)$.

при $a \rightarrow 0$ градиентный спуск будет сходиться, но чем ближе a к 0, тем больше понадобится итераций, при $|a| > 4882$ градиентный спуск перестает сходиться. Если спускаться по $f(x,ay)$, то при $a \rightarrow 0$ градиентный спуск будет сходиться, но чем ближе a к 0, тем больше понадобится итераций, при $|a| > 4882$ градиентный спуск перестает сходиться.

e) α ищется методом золотого сечения:



$\alpha = 0.48$:



5) Надо реализовать генератор функций от n переменных и числом обусловленности k . $f(x) = x^T A x + b^T x + c$, где c - случайная константа, b - случайный вектор, A - случайная матрица квадратичной формы.

Число обусловленности матрицы - это $\lambda_{\max} / \lambda_{\min}$, то есть отношение максимального собственного числа к минимальному собственному числу.

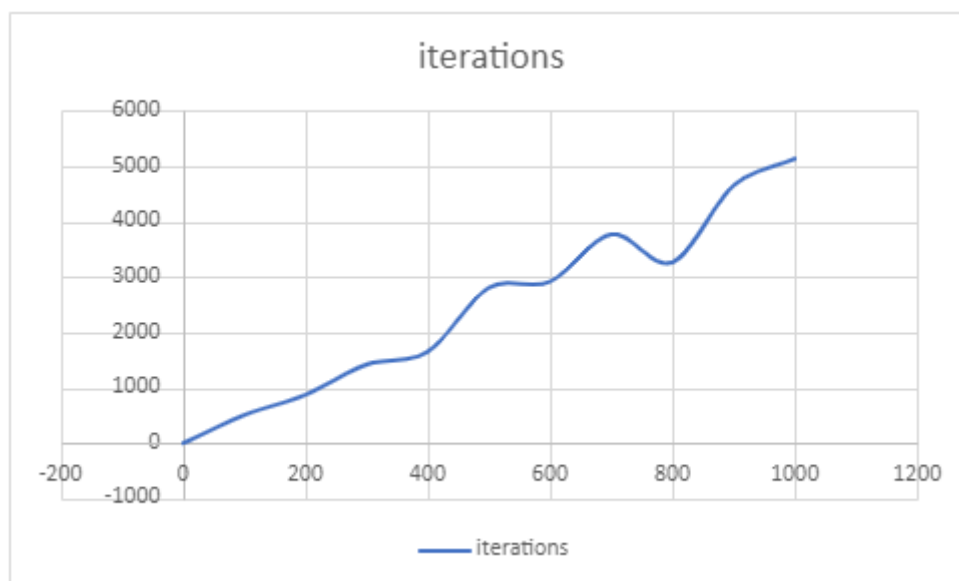
Чтобы получить случайную матрицу с числом обусловленности k , сгенерируем случайную диагональную матрицу diag с $\lambda_{\max} = k$, $\lambda_{\min} = 1$.

Далее сгенерируем случайную ортогональную матрицу Q , тогда $A = Q \cdot \text{diag} \cdot Q^T$.

Таким образом, полученная матрица будет симметричной и ее число обусловленности будет равно k .

6) Проанализируем влияние числа обусловленности на сходимость градиентного спуска. Возьмем $n = 3$, и сравним сколько итераций понадобится градиентному спуску в зависимости от числа обусловленности k . Посмотрим количество итераций при $k = 1, 10, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000$ и возьмем среднее значение.

k	Тест 1	Тест 2	Тест 3	Тест 4	Тест 5	Тест 6	Тест 7	Тест 8	Тест 9	Тест 10	Среднее
1	2	2	2	2	2	2	2	2	2	2	2
10	75	45	10	23	12	77	9	41	82	67	44,1
100	354	279	559	710	144	568	626	681	613	528	506,2
200	942	8	992	613	1002	432	1121	1219	1225	1149	870,3
300	1912	381	1801	1867	674	1297	1616	1731	1514	1418	1421,1
400	167	2226	2256	557	2030	1015	1999	1394	2506	2331	1648,1
500	5316	680	3726	1640	2928	2865	2486	2890	2708	2866	2810,5
600	3245	3439	2902	1524	3392	2468	3779	3343	3957	1110	2915,9
700	4338	3330	3858	3974	2980	2635	4057	6826	2337	3426	3776,1
800	3530	4652	2808	4162	2507	3460	5524	677	2217	3121	3265,8
900	8570	5058	1373	7907	4488	4377	4037	1793	3505	5459	4656,7
1000	4955	4520	5066	7054	16221	684	1951	3007	4984	2945	5138,7



Затем проанализируем влияние размерности на сходимость.
 возьмем $k = 3$, и посмотрим количество итераций при $n = 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100$ и возьмем среднее значение. Берем не очень большие n , так как иначе процесс займет слишком много времени.

п	Тест 1	Тест 2	Тест 3	Тест 4	Среднее
1	3	3	3	3	3
10	25	26	25	25	25,25
20	27	26	26	25	26
30	25	26	25	25	25,25
40	25	25	27	26	25,75
50	27	26	25	26	26
60	26	25	27	27	26,25
70	25	26	25	25	25,25
80	26	25	26	27	26
90	26	26	25	25	25,5
100	26	25	25	25	25,25



Дополнительное задание:

Условия Вольфе - условия, накладываемые на шаг α градиентного спуска для того чтобы сэкономить количество итераций при поиске α .

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k,$$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k$$

$0 < c_1 < c_2 < 1$, c_1 выбирается в окрестности 0, c_2 выбирается в окрестности 1, p - направление вдоль которого шагаем, в нашем случае p - это антиградиент

Проанализируем градиентный спуск с условиями Вольфе на примере функции $0.09x^2 + 3y^2 - 0.3x - 4y$

Обычный градиентный спуск $\alpha = 0.32$:

Count of gradient countings: 207

Count of function evaluations: 414

Градиентный спуск с одномерным поиском:

Count of gradient countings: 161

Count of function evaluations: 11270

Градиентный спуск с условиями Вольфе:

Count of gradient countings: 203

Count of function evaluations: 934

Очевидно, что спуск с условиями Вольфе не выгоднее спуска с одномерным поиском с точки зрения количества вычислений градиента, т.к. во втором случае мы выбираем наилучший α , однако он выгоднее с точки зрения количества вычислений функций.

Выводы: Были реализованы метод градиентного спуска с постоянным шагом, метод градиентного спуска на основе метода золотого сечения, метод градиентного спуска с проверкой условий Вольфе, а также генератор функций от n переменных с числом обусловленности k .

Преимущества градиентного спуска:

- Количество итераций почти не зависит от начального приближения
- Количество итераций почти не зависит от размерности функции
- Относительно несложно реализовать, достаточно лишь уметь считать градиент функции

Недостатки градиентного спуска:

- Количество итераций сильно зависит от числа обусловленности функции
- Долго работает при большой размерности функции
- Может расходиться при большом числе обусловленности функции

Также были сравнены обычный градиентный спуск с постоянным шагом и градиентный спуск на основе метода золотого сечения. Градиентный спуск с постоянным шагом выгоднее с точки зрения количества вычислений функции, но градиентный спуск выгоднее с точки зрения количества вычислений градиента. Также чтобы уменьшить количество вычислений функции, можно добавить проверку условий Вольфе.