Notes

Kopnov Alexandr

15 мая 2023 г.

Содержание

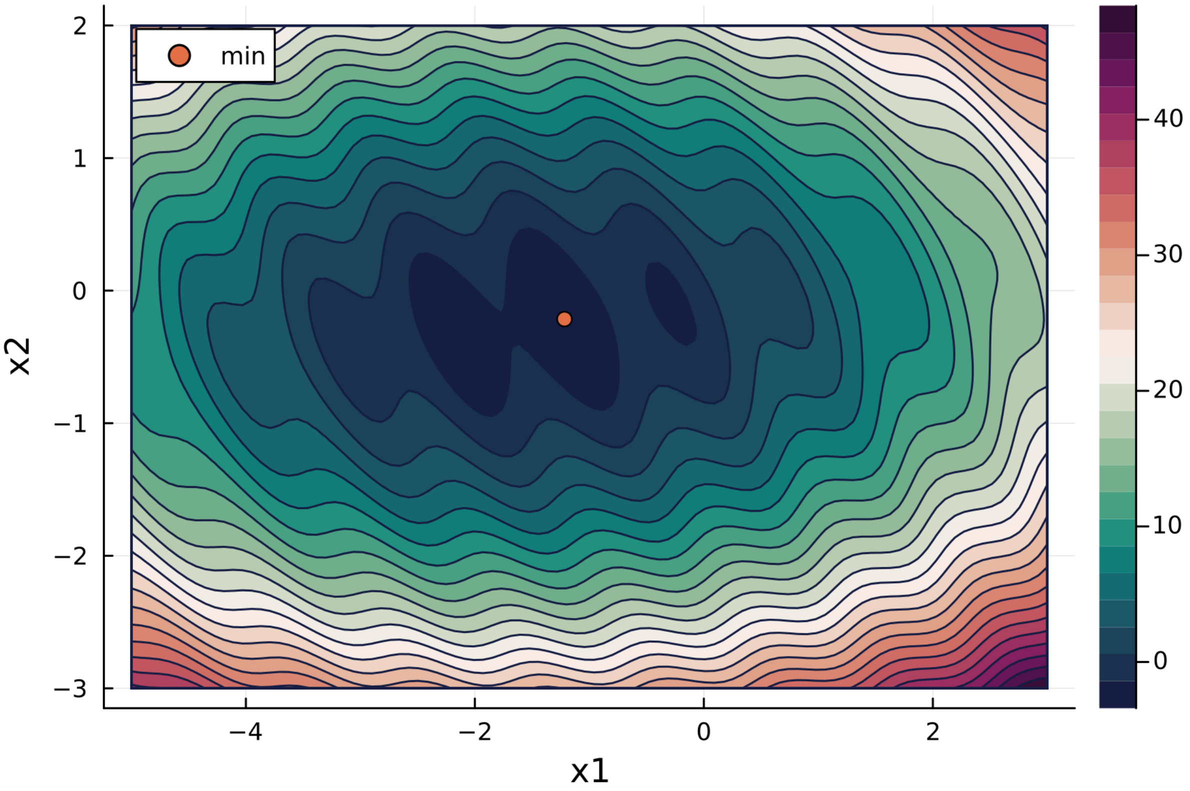
1 Тема 1

2 Методы 2
 2.1 Золотое сечение . . . 2
  2.1.1 Оценка количества шагов . . . 3
 2.2 Равномерный поиск . . . 4
  2.2.1 Оценка количества вызовов . . . 4

1 Тема

Выбор метода решения задачи одномерной оптимизации (равномерный поиск / золотое сечение) для решения вспомогательной задачи градиентным методом наискорейшего спуска

```
x1 = range(-5,3, length=100);
x2 = range(-3,2, length=100);
vals = [f([x,y]) for y in x2, x in x1];
minim = Optim.minimizer(optimize(f,zeros(2), GradientDescent()));
# p1 = surface(x1,x2,vals, c=:curl, xlabel="x1", ylabel="x2", cam=(120,30));
# scatter!([minim[1]],[minim[2]],[f(minim)], label="min")
p2 = contour(x1,x2,vals,levels=25,fill=true, c=:curl, dpi = 300, xlabel="x1", ylabel="x2");
scatter!([minim[1]],[minim[2]], label="min");
# plot(p1,p2, size=(500,300), dpi=300)
savefig("figs/plot.png")
```



([-1.2173845735465747, -0.2146730755196121], -3.410689361818459)

2 Методы

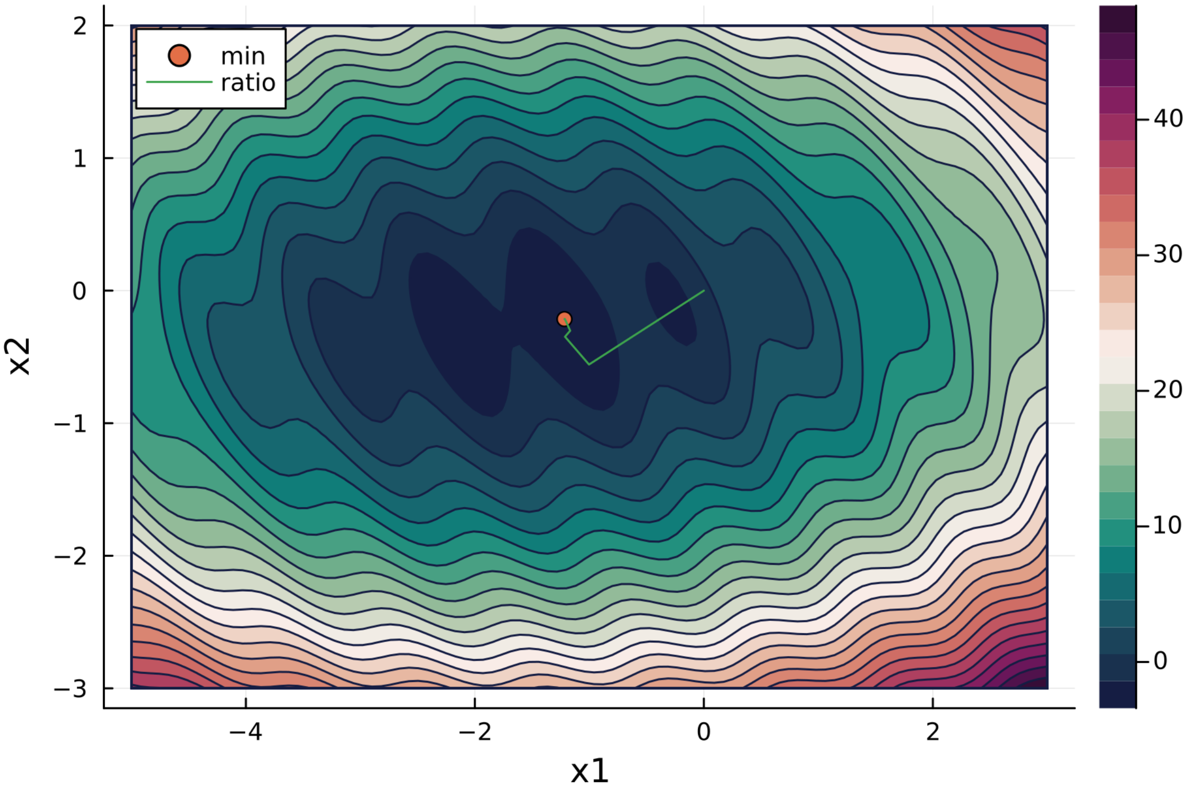
```
df = CSV.read("build/res/" * name, DataFrame; header=["x","y"], delim=' ')
@df df plot(p2, :x, :y, labels = name)
savefig("figs/" * name * ".png")
```

2.1 Золотое сечение

Рассмотрим k-й шаг алгоритма. Интервал $[a_k, b_k]$

$$\lambda_k = a_k + \alpha(b_k - a_k) \qquad \mu_k = b_k - \alpha(b_k - a_k) \qquad \alpha = \frac{3-\sqrt{5}}{2}$$

Если $f(\lambda_k) > f(\mu_k)$ на следующей итерации $[\lambda_k, b_k]$. При этом, $\lambda_{k+1} = \mu_k$ Иначе, $[a_k, \mu_k]$, $\mu_{k+1} = \lambda_k$



2.1.1 Оценка количества шагов

$[a_0, b_0]$ — начальный интервал. ϵ — заданная точность. Рассмотрим k шагов, $i \in \mathbb{N}, i \leqslant k$

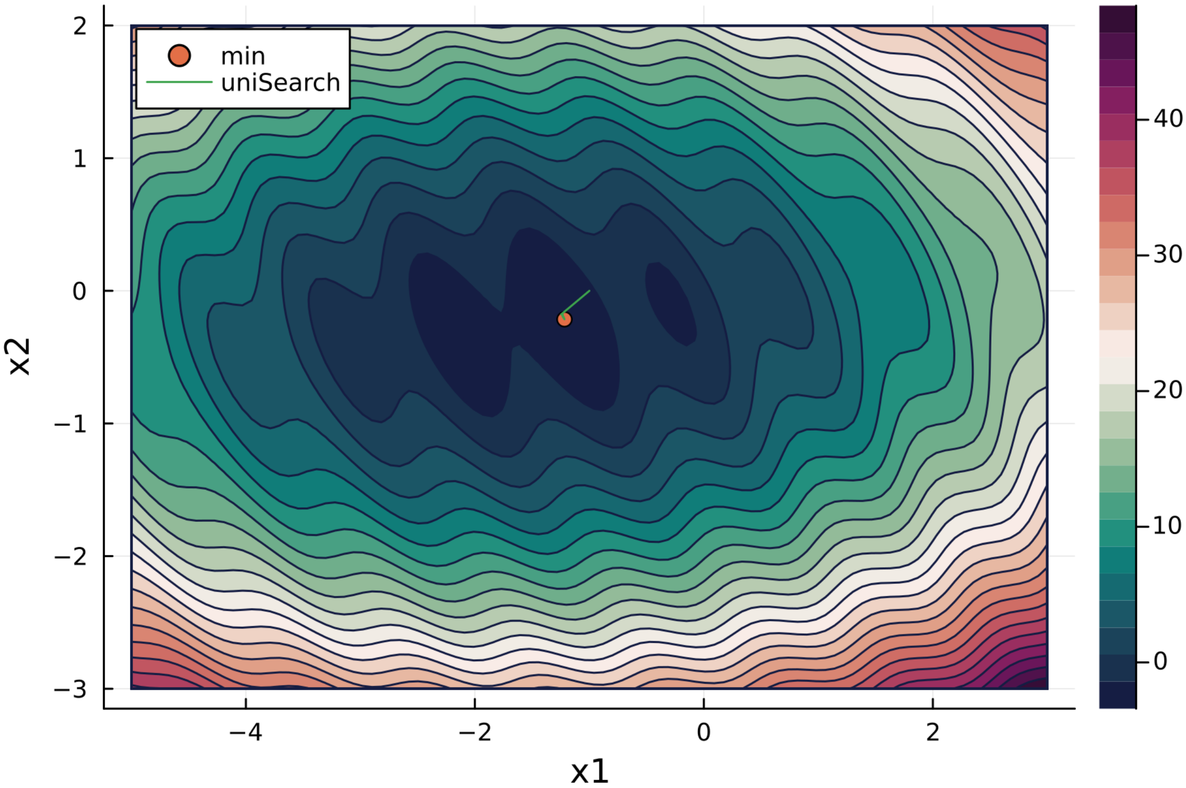
$$k : b_k - a_k \leqslant \epsilon$$

$$b_k - a_k = (b_0 - a_0) \cdot (1 - \lambda)^k$$

$$\epsilon \geqslant (b_0 - a_0) \cdot (1 - \lambda)^k \implies k \geqslant \log_{\frac{1}{1-\lambda}}(\frac{b_0 - a_0}{\epsilon})$$

На первом шаге функция вызывается 2 раза, на последующих 1. Т.о. всего $k+1$ вызовов функции.

2.2 Равномерный поиск



Рассмотрим k-й шаг алгоритма. Интервал $[a_k, b_k]$. Разобьём его на подинтервалы точками $x_i = a + ih, h = \frac{b-a}{n}, i = \overline{0, n}$.

$$\min_i f(x_i) = f(x_j) \implies f(x_{j-1}) > f(x_j), f(x_{j+1}) > f(x_j)$$

Функция унимодальна, значит $x_* \in [x_{j-1}, x_{j+1}]$. $a_{k+1} = x_{j-1}, b_{k+1} = x_{j+1}$. Продолжаем, пока не будет выполнено условие неопределённости: $b - a = \epsilon$

2.2.1 Оценка количества вызовов

$[a_0, b_0]$ — начальный интервал. ϵ — заданная точность. Интервал разбивается на n подинтервалов. Рассмотрим k шагов, $i \in \mathbb{N}, i \leqslant k$

$$k : b_k - a_k \leqslant \epsilon$$

$$b_{i+1} - a_{i+1} = \frac{2}{n}(b_i - a_i)$$

$$b_k - a_k = \left(\frac{2}{n}\right)^k (b_0 - a_0)$$

$$\epsilon \geqslant b_k - a_k \implies k \geqslant \log_{\frac{n}{2}} \left(\frac{b_0 - a_0}{\epsilon}\right)$$

На каждом шаге функция вызывается $n-1$ раз, по количеству внутренних точек. Таким образом, общее количество вызовов $k \cdot (n-1)$