

Матрично-векторное дифференцирование





#### Градиент

Пусть  $f(x):\mathbb{R}^n o \mathbb{R}$ , тогда вектор, который содержит все первые частные производные:

$$\nabla f(x) = \frac{df}{dx} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$



#### Градиент

Пусть  $f(x):\mathbb{R}^n o \mathbb{R}$ , тогда вектор, который содержит все первые частные производные:

$$\nabla f(x) = \frac{df}{dx} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

называется градиентом функции f(x). Этот вектор указывает направление наискорейшего возрастания. Таким образом, вектор  $-\nabla f(x)$  указывает направление наискорейшего убывания функции в точке. Кроме того, вектор градиента всегда ортогонален линии уровня в точке.

#### i Example

Для функции  $f(x,y) = x^2 + y^2$  градиент равен:

$$\nabla f(x,y) = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

Он указывает направление наискорейшего возрастания функции.

#### **i** Question

Как связана норма градиента с крутизной функции?

4

#### Гессиан

Пусть  $f(x):\mathbb{R}^n o \mathbb{R}$ , тогда матрица, содержащая все вторые частные производные:

$$f''(x) = \nabla^2 f(x) = \frac{\partial^2 f}{\partial x_i \partial x_j} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_i \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_i \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix}$$



#### Гессиан

Пусть  $f(x):\mathbb{R}^n o \mathbb{R}$ , тогда матрица, содержащая все вторые частные производные:

$$f''(x) = \nabla^2 f(x) = \frac{\partial^2 f}{\partial x_i \partial x_j} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix} \qquad \text{Для фу}$$

Гессиан может быть тензором:  $(f(x):\mathbb{R}^n o \mathbb{R}^n)$  Таким образом, это просто трехмерный тензор, каждый срез которого это гессиан соответствующей скалярной функции  $(
abla^2 f_1(x),\dots,
abla^2 f_m(x))$ .

#### **i** Example

Для функции  $f(x,y)=x^2\!+\!y^2$  гессиан равен:

$$H_f(x,y) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Эта матрица содержит информацию о кривизне функции в разных направлениях.

#### i Question

Как можно использовать гессиан для определения выпуклости или вогнутости функции?

**⊕ ೧ ⊘** 

#### Теорема Шварца

Пусть  $f:\mathbb{R}^n \to \mathbb{R}$  - функция. Если смешанные частные производные  $\frac{\partial^2 f}{\partial x_i \partial x_j}$  и  $\frac{\partial^2 f}{\partial x_j \partial x_i}$  непрерывны на открытом множестве, содержащем точку a, то они равны в точке

$$\it a$$
. То есть,

 $\frac{\partial^2 f}{\partial x_i \partial x_j}(a) = \frac{\partial^2 f}{\partial x_j \partial x_i}(a)$ 

# Теорема Шварца

Пусть  $f:\mathbb{R}^n o \mathbb{R}$  - функция. Если смешанные частные производные  $\frac{\partial^2 f}{\partial x_i \partial x_j}$  и  $\frac{\partial^2 f}{\partial x_j \partial x_i}$  непрерывны на открытом множестве, содержащем точку a, то они равны в точке

$$a$$
. То есть, 
$$\frac{\partial^2 f}{\partial x_i \partial x_i}(a) = \frac{\partial^2 f}{\partial x_i \partial x_i}(a)$$

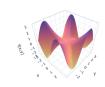
Согласно данной теореме, если смешанные частные производные непрерывны на открытом множестве, то гессиан симметричен. То есть,

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_i \partial x_j} \quad \nabla^2 f(x) = (\nabla^2 f(x))^T$$

Эта симметричность упрощает вычисления и анализ, связанные с гессианом в различных приложениях, особенно в оптимизации.

# і Контрпример Шварца

$$f(x,y) = \begin{cases} \frac{xy(x^2-y^2)}{x^2+y^2} & \text{ для } (x,\,y) \neq (0,\,0), \\ 0 & \text{ для } (x,y) = (0,0). \end{cases}$$





**⊕** ດ **ø** 

Можно проверить, что  $\frac{\partial^2 f}{\partial x \partial y}(0,0) \neq \frac{\partial^2 f}{\partial y \partial x}(0,0)$ , хотя смешанные частные производные существуют, и в каждой другой точке симметричность выполняется.

 $f \to \min_{x,y,z}$ 

Матрично-векторное дифференцирование

#### Якобиан

Обобщением понятия градиента на случай многомерной функции  $f(x):\mathbb{R}^n \to \mathbb{R}^m$  является следующая матрица:

$$J_f = f'(x) = \frac{df}{dx^T} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_n} & \frac{\partial f_2}{\partial x_n} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

Она содержит информацию о скорости изменения функции по отношению к ее входу.

#### Question

Можно ли связать эти три определения выше (градиент, якобиан, и гессиан) с помощью одного утверждения?

#### i Example

Для функции

$$f(x,y) = \begin{bmatrix} x+y \\ x-y \end{bmatrix},$$

Якобиан равен:

$$J_f(x,y) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

#### i Question

Как матрица Якоби связана с градиентом для скалярных функций?



#### Итог

$$f(x):X\to Y;\qquad \frac{\partial f(x)}{\partial x}\in G$$

X	Υ	G	Name
$\mathbb{R}$	$\mathbb{R}$	$\mathbb R$	f'(x) (производная)
$\mathbb{R}^n$	$\mathbb{R}$	$\mathbb{R}^n$	$rac{\partial f}{\partial x_i}$ (градиент)
$\mathbb{R}^n$	$\mathbb{R}^m$	$\mathbb{R}^{n  imes m}$	$rac{\partial f_i}{\partial x_i}$ (якобиан)
$\mathbb{R}^{m  imes n}$	$\mathbb{R}$	$\mathbb{R}^{m  imes n}$	$rac{\partial f}{\partial x_{ij}}$

**♥ ೧ Ø** 7

Аппроксимация Тейлора первого порядка, также известная как линейное приближение, строится вблизи некоторой точки  $x_0$ . Если  $f: \mathbb{R}^n \to \mathbb{R}$  дифференцируемая функция, то ее аппроксимация первого порядка задается следующим образом:

$$f_{x_0}^I(x) = f(x_0) + \nabla f(x_0)^T (x - x_0)$$

где:

•  $f(x_0)$  - значение функции в точке  $x_0$ .

Аппроксимация Тейлора первого порядка, также известная как линейное приближение, строится вблизи некоторой точки  $x_0$ . Если  $f: \mathbb{R}^n \to \mathbb{R}$  дифференцируемая функция, то ее аппроксимация первого порядка задается следующим образом:

$$f_{x_0}^I(x) = f(x_0) + \nabla f(x_0)^T (x - x_0)$$

где:

- $f(x_0)$  значение функции в точке  $x_0$ .
- $\nabla f(x_0)$  градиент функции в точке  $x_0$ .



Аппроксимация Тейлора первого порядка, также известная как линейное приближение, строится вблизи некоторой точки  $x_0$ . Если  $f: \mathbb{R}^n \to \mathbb{R}$  дифференцируемая функция, то ее аппроксимация первого порядка задается следующим образом:

$$f_{x_0}^I(x) = f(x_0) + \nabla f(x_0)^T (x - x_0)$$

где:

- $f(x_0)$  значение функции в точке  $x_0$ .
- $\nabla f(x_0)$  градиент функции в точке  $x_0$ .



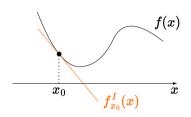
Аппроксимация Тейлора первого порядка, также известная как линейное приближение, строится вблизи некоторой точки  $x_0$ . Если  $f: \mathbb{R}^n \to \mathbb{R}$  дифференцируемая функция, то ее аппроксимация первого порядка задается следующим образом:

$$f_{x_0}^I(x) = f(x_0) + \nabla f(x_0)^T (x - x_0)$$

где:

- $f(x_0)$  значение функции в точке  $x_0$ .
- $\nabla f(x_0)$  градиент функции в точке  $x_0$ .

Часто для упрощения теоретического анализа в некоторых методах заменяют Рисунок 1: Аппроксимация Тейлора функцию вблизи некоторой точки на её аппроксимацию



первого порядка в окрестности точки  $x_0$ 

Аппроксимация Тейлора второго порядка, также известная как квадратичное приближение, использует информацию о кривизне функции. Для дважды дифференцируемой функции  $f:\mathbb{R}^n \to \mathbb{R}$ , ее аппроксимация второго порядка, строящаяся вблизи некоторой точки  $x_0$ , задается следующим образом:

$$f_{x_0}^{II}(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2} (x - x_0)^T \nabla^2 f(x_0) (x - x_0)$$

Где  $\nabla^2 f(x_0)$  - гессиан функции f в точке  $x_0$ .



Аппроксимация Тейлора второго порядка, также известная как квадратичное приближение, использует информацию о кривизне функции. Для дважды дифференцируемой функции  $f:\mathbb{R}^n o\mathbb{R}$ , ее аппроксимация второго порядка, строящаяся вблизи некоторой точки  $x_0$ , задается следующим образом:

$$f_{x_0}^{II}(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2} (x - x_0)^T \nabla^2 f(x_0) (x - x_0)$$

Где  $\nabla^2 f(x_0)$  - гессиан функции f в точке  $x_0$ .

Когда линейного приближения функции не достаточно, можно рассмотреть замену f(x) на  $f_{x_0}^{II}(x)$  в окрестности точки  $x_0$ . В общем, приближения Тейлора дают нам способ локально аппроксимировать функции.

Аппроксимация первого порядка определяется градиентом функции в точке, т.е. нормалью к касательной гиперплоскости. А аппроксимация второго порядка представляет из себя параболу. Эти приближения особенно полезны в оптимизации и численных методах, потому что они предоставляют простой

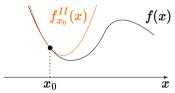


Рисунок 2: Аппроксимация Тейлора второго порядка в окрестности точки  $x_0$ 



способ работы со сложными функциями.

## Дифференциалы

#### **i** Theorem

Пусть  $x \in S$  - внутренняя точка множества S, и пусть  $D: U \to V$  - линейный оператор. Мы говорим, что функция f дифференцируема в точке x с производной D, если для всех достаточно малых  $h \in U$  выполняется следующее разложение:

$$f(x+h) = f(x) + D[h] + o(||h||)$$

Если для любого линейного оператора  $D:U\to V$  функция f не дифференцируема в точке x с производной D, то мы говорим, что f не дифференцируема в точке x.





# Дифференциалы

После получения дифференциальной записи df мы можем получить градиент, используя следующую формулу:

$$df(x) = \langle \nabla f(x), dx \rangle$$

### Дифференциалы

После получения дифференциальной записи df мы можем получить градиент, используя следующую формулу:

$$df(x) = \langle \nabla f(x), dx \rangle$$

Далее, если у нас есть дифференциал в такой форме и мы хотим вычислить вторую производную матричной/векторной функции, мы рассматриваем "старый" dx как константу  $dx_1$ , затем вычисляем  $d(df) = d^2 f(x)$ 

$$d^2f(x) = \langle \nabla^2 f(x) dx_1, dx \rangle = \langle H_f(x) dx_1, dx \rangle$$





Пусть A и B - постоянные матрицы, а X и Y - переменные (или матричные функции).

• dA = 0

- dA = 0
- $\bullet \ d(\alpha X) = \alpha(dX)$

- dA = 0
- $d(\alpha X) = \alpha(dX)$
- d(AXB) = A(dX)B



- dA = 0
- $d(\alpha X) = \alpha(dX)$
- d(AXB) = A(dX)B
- d(X+Y) = dX + dY

- dA = 0
- $d(\alpha X) = \alpha(dX)$
- d(AXB) = A(dX)B
- d(X+Y) = dX + dY•  $d(X^T) = (dX)^T$



- dA = 0
- $d(\alpha X) = \alpha(dX)$
- d(AXB) = A(dX)B
- d(X+Y) = dX + dY
- $d(X^T) = (dX)^T$
- $a(X^{-}) = (aX)^{-}$
- $\bullet \ d(XY) = (dX)Y + X(dY)$



- dA = 0
- $d(\alpha X) = \alpha(dX)$
- d(AXB) = A(dX)B
- d(X+Y) = dX + dY
- $d(X^T) = (dX)^T$
- d(XY) = (dX)Y + X(dY)
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$



Пусть A и B - постоянные матрицы, а X и Y - переменные (или матричные функции).

- dA = 0
- $d(\alpha X) = \alpha(dX)$
- d(AXB) = A(dX)B
- d(X+Y) = dX + dY
- $d(X^T) = (dX)^T$
- $a(X^{\perp}) = (aX)^{\perp}$
- $\bullet \ d(XY) = (dX)Y + X(dY)$
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$

•  $d\left(\frac{X}{\phi}\right) = \frac{\phi dX - (d\phi)X}{\phi^2}$ 



- dA = 0
- $d(\alpha X) = \alpha(dX)$
- d(AXB) = A(dX)B
- $\bullet \ d(X+Y) = dX + dY$
- $d(X^T) = (dX)^T$
- $a(X^{\perp}) = (aX)^{\perp}$
- $\bullet \ d(XY) = (dX)Y + X(dY)$
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$

- $d\left(\frac{X}{\phi}\right) = \frac{\phi dX (d\phi)X}{\phi^2}$
- $d(\det X) = \det X\langle X^{-T}, dX \rangle$

- dA = 0
- $d(\alpha X) = \alpha(dX)$
- d(AXB) = A(dX)B
- d(X+Y) = dX + dY
- $d(X^T) = (dX)^T$
- $a(X^{\perp}) = (aX)^{\perp}$
- $\bullet \ d(XY) = (dX)Y + X(dY)$
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$

- $d\left(\frac{X}{\phi}\right) = \frac{\phi dX (d\phi)X}{\phi^2}$
- $d(\det X) = \det X\langle X^{-T}, dX \rangle$
- $d(\operatorname{tr} X) = \langle I, dX \rangle$

- dA = 0
- $d(\alpha X) = \alpha(dX)$
- d(AXB) = A(dX)B
- d(X+Y) = dX + dY
- $d(X^T) = (dX)^T$
- d(XY) = (dX)Y + X(dY)
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$

- $d\left(\frac{X}{\phi}\right) = \frac{\phi dX (d\phi)X}{\phi^2}$
- $d(\det X) = \det X \langle X^{-T}, dX \rangle$
- $d(\operatorname{tr} X) = \langle I, dX \rangle$
- $df(g(x)) = \frac{df}{dg} \cdot dg(x)$

- dA = 0
- $d(\alpha X) = \alpha(dX)$
- d(AXB) = A(dX)B
- d(X+Y) = dX + dY
- $d(X^T) = (dX)^T$
- d(XY) = (dX)Y + X(dY)
- a(XY) = (aX)Y + X(aY)
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$

- $d\left(\frac{X}{\phi}\right) = \frac{\phi dX (d\phi)X}{\phi^2}$
- $d(\det X) = \det X\langle X^{-T}, dX \rangle$
- $d(\operatorname{tr} X) = \langle I, dX \rangle$
- $df(g(x)) = \frac{df}{dg} \cdot dg(x)$
- $H = (J(\nabla f))^T$

Пусть A и B - постоянные матрицы, а X и Y - переменные (или матричные функции).

- dA = 0
- $d(\alpha X) = \alpha(dX)$
- d(AXB) = A(dX)B
- $\bullet$  d(X+Y)=dX+dY
- $d(X^T) = (dX)^T$
- d(XY) = (dX)Y + X(dY)
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$

- $d(\det X) = \det X\langle X^{-T}, dX\rangle$ •  $d(\operatorname{tr} X) = \langle I, dX \rangle$ 
  - $df(g(x)) = \frac{df}{da} \cdot dg(x)$
  - $H = (J(\nabla f))^T$
  - $d(X^{-1}) = -X^{-1}(dX)X^{-1}$

•  $d\left(\frac{X}{\phi}\right) = \frac{\phi dX - (d\phi)X}{\phi^2}$ 

#### **i** Example

Найти  $df, \nabla f(x)$ , если  $f(x) = \langle x, Ax \rangle - b^T x + c$ .

i Example

Найти  $df, \nabla f(x)$ , если  $f(x) = \ln \langle x, Ax \rangle.$ 

#### i Example

Найти  $df, \nabla f(x)$ , если  $f(x) = \ln \langle x, Ax \rangle$ .

1. Заметим, что A должна быть положительно определенной, потому что  $\langle x,Ax \rangle$  аргумент логарифма и для любого x формула должна быть положительной. Таким образом,  $A \in \mathbb{S}^n_{++}$  Давайте сначала найдем дифференциал:

$$\begin{split} df &= d\left(\ln\langle x,Ax\rangle\right) = \frac{d\left(\langle x,Ax\rangle\right)}{\langle x,Ax\rangle} = \frac{\langle dx,Ax\rangle + \langle x,d(Ax)\rangle}{\langle x,Ax\rangle} = \\ &= \frac{\langle Ax,dx\rangle + \langle x,Adx\rangle}{\langle x,Ax\rangle} = \frac{\langle Ax,dx\rangle + \langle A^Tx,dx\rangle}{\langle x,Ax\rangle} = \frac{\langle (A+A^T)x,dx\rangle}{\langle x,Ax\rangle} \end{split}$$

#### i Example

Найти df,  $\nabla f(x)$ , если  $f(x) = \ln \langle x, Ax \rangle$ .

1. Заметим, что A должна быть положительно определенной, потому что  $\langle x,Ax \rangle$  аргумент логарифма и для любого x формула должна быть положительной. Таким образом,  $A \in \mathbb{S}^n_{++}$  Давайте сначала найдем дифференциал:

$$df = d\left(\ln\langle x, Ax \rangle\right) = \frac{d\left(\langle x, Ax \rangle\right)}{\langle x, Ax \rangle} = \frac{\langle dx, Ax \rangle + \langle x, d(Ax) \rangle}{\langle x, Ax \rangle} =$$

$$= \frac{\langle Ax, dx \rangle + \langle x, Adx \rangle}{\langle x, Ax \rangle} = \frac{\langle Ax, dx \rangle + \langle A^Tx, dx \rangle}{\langle x, Ax \rangle} = \frac{\langle (A + A^T)x, dx \rangle}{\langle x, Ax \rangle}$$

2. Наша основная цель - получить форму  $df = \langle \cdot, dx \rangle$ 

$$df = \left\langle \frac{2Ax}{\langle x, Ax \rangle}, dx \right\rangle$$

Таким образом, градиент равен  $\nabla f(x) = \frac{2Ax}{\langle x, Ax \rangle}$ 

# Матричное дифференцирование. Пример 3

## **i** Example

Найти  $df, \nabla f(X)$ , если  $f(X) = \langle S, X \rangle - \log \det X$ .

### Линейный поиск





Предположим, у нас есть задача минимизации функции  $f(x):\mathbb{R} o \mathbb{R}$  скалярной переменной:

$$f(x)\to \min_{x\in\mathbb{R}}$$

nin u.z Линейный поиск

Предположим, у нас есть задача минимизации функции  $f(x):\mathbb{R} o \mathbb{R}$  скалярной переменной:

$$f(x) \to \min_{x \in \mathbb{R}}$$

Иногда мы рассматриваем похожую задачу поиска минимума функции на отрезке [a,b]:

$$f(x) \to \min_{x \in [a,b]}$$



Предположим, у нас есть задача минимизации функции  $f(x): \mathbb{R} \to \mathbb{R}$  скалярной переменной:

$$f(x) \to \min_{x \in \mathbb{R}}$$

Иногда мы рассматриваем похожую задачу поиска минимума функции на отрезке [a,b]:

$$f(x) \to \min_{x \in [a,b]}$$

#### i Example

Типичным примером задачи линейного поиска является выбор подходящего шага для алгоритма градиентного спуска:

$$\begin{aligned} x_{k+1} &= x_k - \alpha \nabla f(x_k) \\ \alpha &= \operatorname{argmin} \, f(x_{k+1}) \end{aligned}$$



Предположим, у нас есть задача минимизации функции  $f(x):\mathbb{R} o \mathbb{R}$  скалярной переменной:

$$f(x) \to \min_{x \in \mathbb{R}}$$

Иногда мы рассматриваем похожую задачу поиска минимума функции на отрезке [a,b]:

$$f(x) \to \min_{x \in [a,b]}$$

#### i Example

Типичным примером задачи линейного поиска является выбор подходящего шага для алгоритма градиентного спуска:

$$\begin{aligned} x_{k+1} &= x_k - \alpha \nabla f(x_k) \\ \alpha &= \operatorname{argmin} \, f(x_{k+1}) \end{aligned}$$

Линейный поиск является фундаментальной задачей оптимизации, использующийся для решения сложных задач. Для упрощения предположим, что f(x) унимодальна, то есть имеет единственный пик или впадину.

# Унимодальная функция

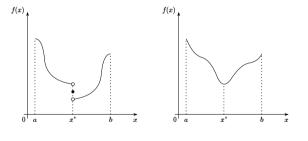
#### i Definition

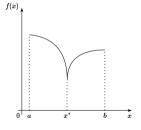
Функция f(x) называется **унимодальной** на отрезке [a,b], если существует  $x_* \in [a,b]$ , что  $f(x_1) > f(x_2) \quad \forall a \leq x_1 < x_2 < x_*$  и  $f(x_1) < f(x_2) \quad \forall x_* < x_1 < x_2 \leq b$ 

# Унимодальная функция

#### i Definition

Функция f(x) называется **унимодальной** на отрезке [a,b], если существует  $x_* \in [a,b]$ , что  $f(x_1) > f(x_2) \quad \forall a \leq x_1 < x_2 < x_*$  и  $f(x_1) < f(x_2) \quad \forall x_* < x_1 < x_2 \leq b$ 





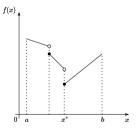


Рисунок 3: Примеры унимодальных функций

Пусть f(x) является унимодальной функцией на отрезке [a,b]. Тогда если  $x_1 < x_2 \in [a,b]$ , то:

 $\bullet \ \text{ if } f(x_1) \leq f(x_2) \to x_* \in [a,x_2]$ 



Пусть f(x) является унимодальной функцией на отрезке [a,b]. Тогда если  $x_1 < x_2 \in [a,b]$ , то:

- if  $f(x_1) \le f(x_2) \to x_* \in [a, x_2]$
- if  $f(x_1) \ge f(x_2) \to x_* \in [x_1, b]$

Пусть f(x) является унимодальной функцией на отрезке [a,b]. Тогда если  $x_1 < x_2 \in [a,b]$ , то:

- if  $f(x_1) \le f(x_2) \to x_* \in [a, x_2]$
- if  $f(x_1) \ge f(x_2) \to x_* \in [x_1, b]$

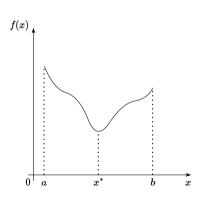
Пусть f(x) является унимодальной функцией на отрезке [a,b]. Тогда если  $x_1 < x_2 \in [a,b]$ , то:

- $\bullet \ \text{ if } f(x_1) \leq f(x_2) \rightarrow x_* \in [a,x_2]$
- if  $f(x_1) \ge f(x_2) \to x_* \in [x_1, b]$



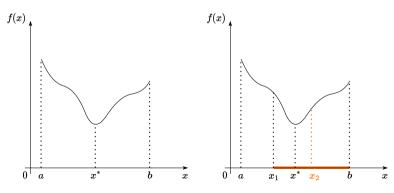
Пусть f(x) является унимодальной функцией на отрезке [a,b]. Тогда если  $x_1 < x_2 \in [a,b]$ , то:

- $\bullet \text{ if } f(x_1) \leq f(x_2) \to x_* \in [a,x_2]$
- $\bullet \ \text{ if } f(x_1) \geq f(x_2) \rightarrow x_* \in [x_1,b]$



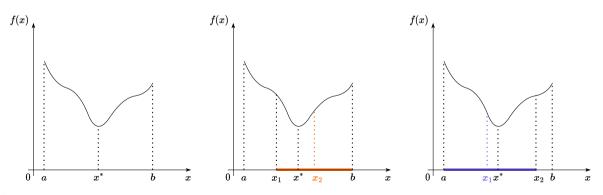
Пусть f(x) является унимодальной функцией на отрезке [a,b]. Тогда если  $x_1 < x_2 \in [a,b]$ , то:

- $\bullet \ \text{ if } f(x_1) \leq f(x_2) \rightarrow x_* \in [a,x_2]$
- $\bullet \ \text{ if } f(x_1) \geq f(x_2) \rightarrow x_* \in [x_1,b]$



Пусть f(x) является унимодальной функцией на отрезке [a,b]. Тогда если  $x_1 < x_2 \in [a,b]$ , то:

- if  $f(x_1) \le f(x_2) \to x_* \in [a, x_2]$
- if  $f(x_1) \ge f(x_2) \to x_* \in [x_1, b]$



Мы хотим решить следующую задачу:

$$f(x) \to \min_{x \in [a,b]}$$

Делим отрезок на две равные части и выбираем, основываясь на ключевом свойстве, описанном выше, ту, которая содержит решение задачи. Наша цель после одной итерации метода - локализовать решение в отрезке в два раза меньшей длины.

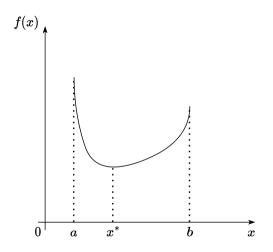


Рисунок 4: Метод дихотомии для унимодальной функции



Мы измеряем значение функции в середине отрезка

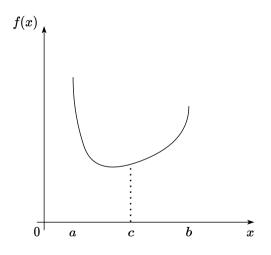


Рисунок 5: Dichotomy method for unimodal function

⊕ ი დ

Чтобы применить ключевое свойство, мы выполняем еще одно измерение.

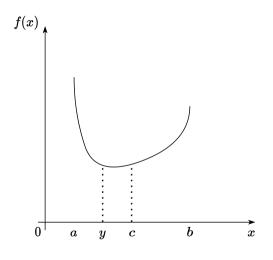


Рисунок 6: Метод дихотомии для унимодальной функции

⇔റ⊘

Выбираем целевой отрезок. В этом случае нас все устраивает, потому что уже разделили решение на две равные части. Но это не всегда так.

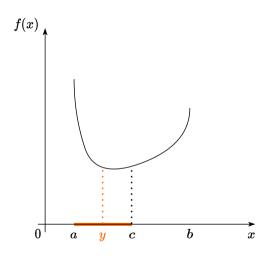


Рисунок 7: Метод дихотомии для унимодальной функции



Рассмотрим другую унимодальную функцию.

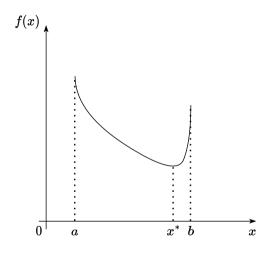


Рисунок 8: Метод дихотомии для унимодальной функции



Измеряем значение функции в середине отрезка.

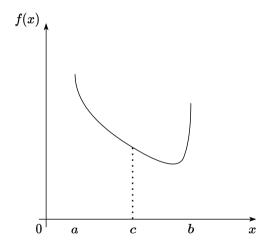


Рисунок 9: Метод дихотомии для унимодальной функции



Делаем еще одно измерение.

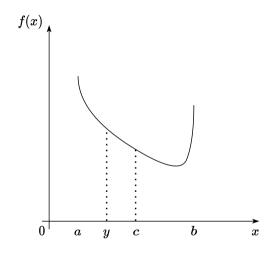


Рисунок 10: Метод дихотомии для унимодальной функции



Выбираем целевой отрезок. Мы можем видеть, что полученный отрезок не является половиной исходного. Он равен  $\frac{3}{4}(b-a)$ . Чтобы исправить это, нам нужен еще один шаг алгоритма.

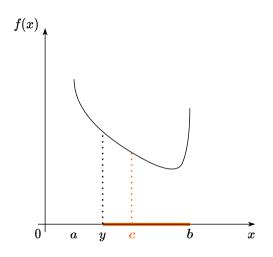


Рисунок 11: Метод дихотомии для унимодальной функции



После еще одного дополнительного измерения мы точно получим  $\frac{2}{3}\frac{3}{4}(b-a)=\frac{1}{2}(b-a)$ 

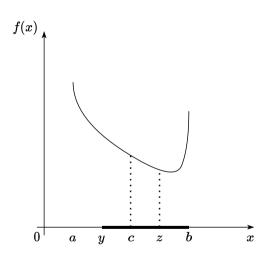


Рисунок 12: Метод дихотомии для унимодальной функции

⊕ ი

В итоге, каждая последующая итерация будет требовать не более двух измерений значений функции.

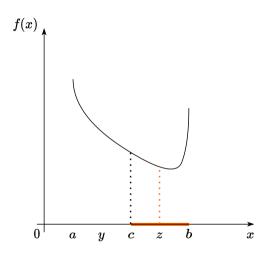


Рисунок 13: Метод дихотомии для унимодальной функции

Линейный поиск

## Метод дихотомии. Алгоритм

```
def binary_search(f, a, b, epsilon):
   c = (a + b) / 2
      while abs(b - a) > epsilon:
         y = (a + c) / 2.0
         if f(y) \ll f(c):
          b = c
            c = A
         else:
            z = (b + c) / 2.0
         if f(c) \le f(z):
            a = y
            b = z
         else:
            a = c
            c = z
      return c
```

 $f \to \min_{x,y,z}$  Линейный поиск

Длина отрезка на k-й итерации:

$$\Delta_k = b_k - a_k = \frac{1}{2^k}(b-a)$$



Длина отрезка на k-й итерации:

$$\Delta_k = b_k - a_k = \frac{1}{2^k}(b - a)$$

Для унимодальных функций это верно, если мы выбираем середину отрезка в качестве выхода итерации  $x_k$ :

$$|x_k - x_*| \leq \frac{\Delta_k}{2} \leq \frac{1}{2^{k+1}} (b-a) \leq (0.5)^k \cdot \frac{b-a}{2}$$



Длина отрезка на k-й итерации:

$$\Delta_k = b_k - a_k = \frac{1}{2^k}(b - a)$$

Для унимодальных функций это верно, если мы выбираем середину отрезка в качестве выхода итерации  $x_{\scriptscriptstyle L}$ :

$$|x_k - x_*| \le \frac{\Delta_k}{2} \le \frac{1}{2^{k+1}} (b - a) \le (0.5)^k \cdot \frac{b - a}{2}$$

Заметим, что на каждой итерации мы спрашиваем оракул не более двух раз, поэтому количество вызовов функции равно  $N = 2 \cdot k$ , что означает:

$$|x_k - x_*| \le (0.5)^{\frac{N}{2}} \cdot \frac{b-a}{2} \le (0.707)^N \frac{b-a}{2}$$

Длина отрезка на k-й итерации:

$$\Delta_k = b_k - a_k = \frac{1}{2^k}(b - a)$$

Для унимодальных функций это верно, если мы выбираем середину отрезка в качестве выхода итерации  $x_k$ :

$$|x_k - x_*| \le \frac{\Delta_k}{2} \le \frac{1}{2^{k+1}} (b - a) \le (0.5)^k \cdot \frac{b - a}{2}$$

Заметим, что на каждой итерации мы спрашиваем оракул не более двух раз, поэтому количество вызовов функции равно  $N=2\cdot k$ , что означает:

$$|x_k - x_*| \le (0.5)^{\frac{N}{2}} \cdot \frac{b-a}{2} \le (0.707)^N \frac{b-a}{2}$$

Помечая правую часть последнего неравенства за  $\varepsilon$ , мы получаем количество итераций метода, необходимое для достижения точности  $\varepsilon$ :

$$K = \left\lceil \log_2 \frac{b - a}{\varepsilon} - 1 \right\rceil$$

#### Метод золотого сечения

Идея очень похожа на метод дихотомии. На отрезке есть две точки - левая и правая точки золотого сечения и интуитивно понятно, что на следующей итерации одна из точек останется точкой золотого сечения.

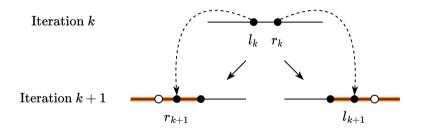


Рисунок 14: Идея, позволяющая уменьшить количество вызовов функции



# Метод золотого сечения. Алгоритм

```
def golden_search(f, a, b, epsilon):
   tau = (sqrt(5) + 1) / 2
   y = a + (b - a) / tau**2
   z = a + (b - a) / tau
   while b - a > epsilon:
      if f(y) \leq f(z):
         b = z
         z = y
         y = a + (b - a) / tau**2
      else:
         a = v
        v = z
         z = a + (b - a) / tau
   return (a + b) / 2
```

# Метод золотого сечения. Оценка

$$|x_k - x_*| \le \frac{b_k - a_k}{2} = \left(\frac{1}{\tau}\right)^N \frac{b - a}{2} \approx 0.618^k \frac{b - a}{2}$$

где 
$$au = rac{\sqrt{5}+1}{2}$$
 .

• Знаменатель геометрической прогрессии для метода золотого сечения **больше**, чем для метода дихотомии: 0.618 больше, чем 0.5.



### Метод золотого сечения. Оценка

$$|x_k - x_*| \le \frac{b_k - a_k}{2} = \left(\frac{1}{\tau}\right)^N \frac{b - a}{2} \approx 0.618^k \frac{b - a}{2}$$

где 
$$au = rac{\sqrt{5}+1}{2}$$
 .

- Знаменатель геометрической прогрессии для метода золотого сечения больше, чем для метода дихотомии: 0.618 больше, чем 0.5.
- Количество вызовов функции меньше для метода золотого сечения, чем для метода дихотомии: 0.707 больше (значит медленнее), чем  $0.618.\;$  Для каждой итерации метода дихотомии (кроме первой), функция вызывается не более двух раз, в то время как для метода золотого сечения, она вызывается не более одного раза за итерацию.



## Метод параболической интерполяции

Три точки, не лежащие на одной прямой, однозначно определяют параболу, проходящую через них. Идея метода — аппроксимировать функцию такой параболой и в качестве следующего приближения взять точку её минимума. Предположим, у нас есть 3 точки  $x_1 < x_2 < x_3$  такие, что отрезок  $[x_1, x_3]$  содержит минимум функции f(x). Тогда мы должны решить следующую систему уравнений:

## Метод параболической интерполяции

Три точки, не лежащие на одной прямой, однозначно определяют параболу, проходящую через них. Идея метода — аппроксимировать функцию такой параболой и в качестве следующего приближения взять точку её минимума. Предположим, у нас есть 3 точки  $x_1 < x_2 < x_3$  такие, что отрезок  $[x_1, x_3]$  содержит минимум функции f(x). Тогда мы должны решить следующую систему уравнений:

$$ax_i^2 + bx_i + c = f_i = f(x_i), i = 1, 2, 3$$

Заметим, что эта система линейна, мы должны решить ее относительно a,b,c. Минимум этой параболы вычисляется по формуле:



# Метод параболической интерполяции

Три точки, не лежащие на одной прямой, однозначно определяют параболу, проходящую через них. Идея метода — аппроксимировать функцию такой параболой и в качестве следующего приближения взять точку её минимума. Предположим, у нас есть 3 точки  $x_1 < x_2 < x_3$  такие, что отрезок  $[x_1, x_3]$  содержит минимум функции f(x). Тогда мы должны решить следующую систему уравнений:

$$ax_i^2 + bx_i + c = f_i = f(x_i), i = 1, 2, 3$$

Заметим, что эта система линейна, мы должны решить ее относительно a,b,c. Минимум этой параболы вычисляется по формуле:

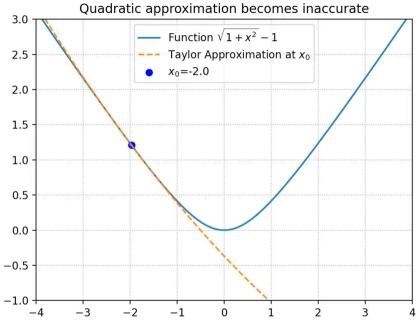
$$u = -\frac{b}{2a} = x_2 - \frac{(x_2 - x_1)^2 (f_2 - f_3) - (x_2 - x_3)^2 (f_2 - f_1)}{2 \left[ (x_2 - x_1)(f_2 - f_3) - (x_2 - x_3)(f_2 - f_1) \right]}$$

Заметим, что если  $f_2 < f_1, f_2 < f_3$ , то u будет лежать в  $[x_1, x_3]$ 

# Метод параболической интерполяции. Алгоритм <sup>1</sup>

```
def parabola_search(f, x1, x2, x3, epsilon):
  f1, f2, f3 = f(x1), f(x2), f(x3)
  while x3 - x1 > epsilon:
     u = x2 - ((x2 - x1)**2*(f2 - f3) - (x2 - x3)**2*(f2 - f1))/(2*((x2 - x1)*(f2 - f3) - (x2 - x3)*(f2 - f1)))
     fu = f(u)
     if x2 <= u:
        if f2 <= fu:
           x1, x2, x3 = x1, x2, u
           f1, f2, f3 = f1, f2, fu
        else:
           x1, x2, x3 = x2, u, x3
           f1. f2. f3 = f2. fu. f3
     else:
        if fu <= f2:
           x1. x2. x3 = x1. u. x2
           f1. f2. f3 = f1. fu. f2
        else:
           x1. x2. x3 = u. x2. x3
           f1. f2. f3 = fu. f2. f3
  return (x1 + x3)/2
```

<sup>&</sup>lt;sup>1</sup>Скорость сходимости этого метода суперлинейна, но локальна, что означает, что мы можем получить выгоду от использования этого метода только вблизи некоторой окрестности оптимума. Здесь доказательство суперлинейной сходимости порядка 1.32.



### Неточный линейный поиск

Нам не всегда нужно точно решать задачу минимизации. Иногда, достаточно найти приближенное решение. Такое часто встречается в задаче выбора шага метода оптимизации

$$\begin{aligned} x_{k+1} &= x_k - \alpha \nabla f(x_k) \\ \alpha &= \operatorname{argmin} \, f(x_{k+1}) \end{aligned}$$



### Неточный линейный поиск

Нам не всегда нужно точно решать задачу минимизации. Иногда, достаточно найти приближенное решение. Такое часто встречается в задаче выбора шага метода оптимизации

$$\begin{aligned} x_{k+1} &= x_k - \alpha \nabla f(x_k) \\ \alpha &= \operatorname{argmin} \, f(x_{k+1}) \end{aligned}$$

Рассмотрим скалярную функцию  $\phi(\alpha)$  в точке  $x_{\iota}$ :

$$\phi(\alpha) = f(x_k - \alpha \nabla f(x_k)), \alpha \geq 0$$



# Неточный линейный поиск

Нам не всегда нужно точно решать задачу минимизации. Иногда, достаточно найти приближенное решение. Такое часто встречается в задаче выбора шага метода оптимизации

$$\begin{aligned} x_{k+1} &= x_k - \alpha \nabla f(x_k) \\ \alpha &= \operatorname{argmin} \, f(x_{k+1}) \end{aligned}$$

Рассмотрим скалярную функцию  $\phi(\alpha)$  в точке  $x_k$ :

$$\phi(\alpha) = f(x_k - \alpha \nabla f(x_k)), \alpha \geq 0$$

Первое приближение  $\phi(\alpha)$  в окрестности  $\alpha=0$  равно:

$$\phi(\alpha) \approx f(x_k) - \alpha \nabla f(x_k)^T \nabla f(x_k)$$

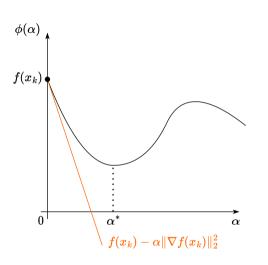


Рисунок 15: Иллюстрация аппроксимации Тейлора  $\phi_0^I(\alpha)$ 

Линейный поиск

# Неточный линейный поиск. Условие достаточного убывания

Условие неточного линейного поиска, известное как условие Армихо, утверждает, что  $\alpha$  должно обеспечить достаточное убывание функции f, удовлетворяющее:

$$f(x_k - \alpha \nabla f(x_k)) \leq f(x_k) - c_1 \cdot \alpha \nabla f(x_k)^T \nabla f(x_k)$$

# Неточный линейный поиск. Условие достаточного убывания

Условие неточного линейного поиска, известное как условие Армихо, утверждает, что  $\alpha$  должно обеспечить достаточное убывание функции f, удовлетворяющее:

$$f(x_k - \alpha \nabla f(x_k)) \leq f(x_k) - c_1 \cdot \alpha \nabla f(x_k)^T \nabla f(x_k)$$

для некоторой постоянной  $c_1 \in (0,1)$ . Заметим, что установка  $c_1 = 1$  соответствует первому приближению Тейлора  $\phi(\alpha)$ . Однако это условие может принимать очень малые значения  $\alpha$ , потенциально замедляя процесс решения. Обычно на практике используется  $c_1 \approx 10^{-4}$ .



# Неточный линейный поиск. Условие достаточного убывания

функцию

Условие неточного линейного поиска, известное как условие Армихо, утверждает, что  $\alpha$  должно обеспечить достаточное убывание функции f, удовлетворяющее:

$$f(x_k - \alpha \nabla f(x_k)) \leq f(x_k) - c_1 \cdot \alpha \nabla f(x_k)^T \nabla f(x_k)$$

установка  $c_1 = 1$  соответствует первому приближению Тейлора  $\phi(\alpha)$ . Однако это условие может принимать очень малые значения  $\alpha$ , потенциально замедляя процесс решения. Обычно на практике используется  $c_1 \approx 10^{-4}$ .

для некоторой постоянной  $c_1 \in (0,1)$ . Заметим, что

# i Example

Если

представляет собой стоимости в задаче оптимизации, важен выбор подходящего значения  $c_1$ . Например, обучении моделей ML неправильное значение  $c_1$ может привести к очень медленной сходимости или пропуску минимума.

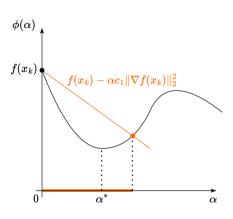


Рисунок 16: Иллюстрация условия достаточного убывания с коэффициентом  $c_1$ 

f(x)

# Неточный линейный поиск. Условия Гольдштейна

Рассмотрим две линейные скалярные функции  $\phi_1(\alpha)$  и  $\phi_2(\alpha)$ :

$$\phi_1(\alpha) = f(x_k) - c_1 \alpha \|\nabla f(x_k)\|^2$$

$$\phi_2(\alpha) = f(x_k) - c_2 \alpha \|\nabla f(x_k)\|^2$$

## Неточный линейный поиск. Условия Гольдштейна

Рассмотрим две линейные скалярные функции  $\phi_1(\alpha)$  и  $\phi_2(\alpha)$ :

$$\phi_1(\alpha) = f(x_k) - c_1 \alpha \|\nabla f(x_k)\|^2$$

$$\phi_2(\alpha) = f(x_k) - c_2 \alpha \|\nabla f(x_k)\|^2$$

Условия Гольдштейна-Армихо находят функцию  $\phi(\alpha)$  между  $\phi_1(\alpha)$  и  $\phi_2(\alpha)$ . Обычно  $c_1=\rho$  и  $c_2=1-\rho$ , с  $\rho\in(0,0.5)$ .

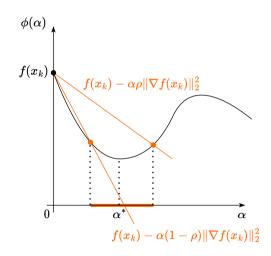


Рисунок 17: Иллюстрация условий Гольдштейна

♥ O •

# Неточный линейный поиск. Условие ограничения на кривизну

Чтобы избежать слишком коротких шагов, мы вводим второй критерий:

$$-\nabla f(x_k - \alpha \nabla f(x_k))^T \nabla f(x_k) \geq c_2 \nabla f(x_k)^T (-\nabla f(x_k))$$



## Неточный линейный поиск. Условие ограничения на кривизну

Чтобы избежать слишком коротких шагов, мы вводим второй критерий:

$$-\nabla f(x_k - \alpha \nabla f(x_k))^T \nabla f(x_k) \geq c_2 \nabla f(x_k)^T (-\nabla f(x_k))$$

для некоторого  $c_2 \in (c_1,1).$  Здесь  $c_1$  из условия Армихо.

Левая часть является производной  $\nabla_{\alpha}\phi(\alpha)$ , гарантирующей, что наклон  $\phi(\alpha)$  в целевой точке не менее чем в  $c_2$  раз больше начального наклона  $\nabla_{\alpha}\phi(\alpha)(0)$ .

Обычно для методов Ньютона и квазиньютоновских методов используется  $c_2 \approx 0.9$ . В объединении условие достаточного убывания и ограничение на кривизну образуют условия Вульфа.

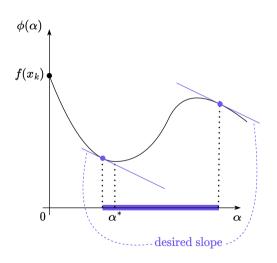


Рисунок 18: Иллюстрация условия ограничения на кривизну

**⊕ი** (

# Неточный линейный поиск. Условия Вульфа

$$-\nabla f(x_k - \alpha \nabla f(x_k))^T \nabla f(x_k) \geq c_2 \nabla f(x_k)^T (-\nabla f(x_k))$$

Вместе, условие достаточного убывания и ограничение на кривизну образуют условия Вульфа.

### **i** Theorem

Пусть  $f:\mathbb{R}^n \to \mathbb{R}$  непрерывно дифференцируема, и пусть  $\phi(\alpha) = f(x_k - \alpha \nabla f(x_k))$ . Предположим, что  $\nabla f(x_k)^T p_k < 0$ , где  $p_k = -\nabla f(x_k)$ , делая  $p_k$  направлением спуска. Также предположим, что f ограничена снизу вдоль луча  $\{x_k + \alpha p_k \mid \alpha > 0\}$ . Мы хотим показать, что для  $0 < c_1 < c_2 < 1$ , существуют интервалы шагов, удовлетворяющие условиям Вульфа.

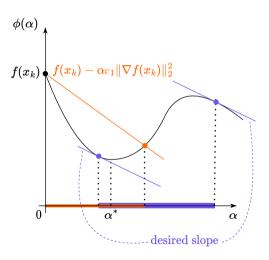


Рисунок 19: Иллюстрация условий Вульфа



Бэктрекинг - это техника для нахождения шага, удовлетворяющего условию Армихо, условиям Гольдштейна или другим критериям неточного линейного поиска. Она начинает с относительно большого шага и итеративно уменьшает его до тех пор, пока не будет выполнено условие.

Бэктрекинг - это техника для нахождения шага, удовлетворяющего условию Армихо, условиям Гольдштейна или другим критериям неточного линейного поиска. Она начинает с относительно большого шага и итеративно уменьшает его до тех пор, пока не будет выполнено условие.

#### Алгоритм:

1. Выберите начальный шаг,  $\alpha_0$ , и параметры  $\beta \in (0,1)$  и  $c_1 \in (0,1)$ .





Бэктрекинг - это техника для нахождения шага, удовлетворяющего условию Армихо, условиям Гольдштейна или другим критериям неточного линейного поиска. Она начинает с относительно большого шага и итеративно уменьшает его до тех пор, пока не будет выполнено условие.

#### Алгоритм:

- 1. Выберите начальный шаг,  $\alpha_0$ , и параметры  $\beta \in (0,1)$  и  $c_1 \in (0,1)$ .
- 2. Проверьте, удовлетворяет ли выбранный шаг выбранному условию (например, условию Армихо).



Бэктрекинг - это техника для нахождения шага, удовлетворяющего условию Армихо, условиям Гольдштейна или другим критериям неточного линейного поиска. Она начинает с относительно большого шага и итеративно уменьшает его до тех пор, пока не будет выполнено условие.

#### Алгоритм:

- 1. Выберите начальный шаг,  $\alpha_0$ , и параметры  $\beta \in (0,1)$  и  $c_1 \in (0,1)$ .
- 2. Проверьте, удовлетворяет ли выбранный шаг выбранному условию (например, условию Армихо).
- 3. Если условие выполнено, остановитесь; в противном случае, установите  $\alpha:=\beta\alpha$  и повторите шаг 2.



Бэктрекинг - это техника для нахождения шага, удовлетворяющего условию Армихо, условиям Гольдштейна или другим критериям неточного линейного поиска. Она начинает с относительно большого шага и итеративно уменьшает его до тех пор, пока не будет выполнено условие.

#### Алгоритм:

- 1. Выберите начальный шаг,  $\alpha_0$ , и параметры  $\beta \in (0,1)$  и  $c_1 \in (0,1)$ .
- 2. Проверьте, удовлетворяет ли выбранный шаг выбранному условию (например, условию Армихо).
- 3. Если условие выполнено, остановитесь; в противном случае, установите  $\alpha:=\beta\alpha$  и повторите шаг 2.



Бэктрекинг - это техника для нахождения шага, удовлетворяющего условию Армихо, условиям Гольдштейна или другим критериям неточного линейного поиска. Она начинает с относительно большого шага и итеративно уменьшает его до тех пор, пока не будет выполнено условие.

#### Алгоритм:

- 1. Выберите начальный шаг,  $\alpha_0$ , и параметры  $\beta \in (0,1)$  и  $c_1 \in (0,1).$
- 2. Проверьте, удовлетворяет ли выбранный шаг выбранному условию (например, условию Армихо).
- 3. Если условие выполнено, остановитесь; в противном случае, установите  $\alpha := \beta \alpha$  и повторите шаг 2. Шаг  $\alpha$  обновляется как

$$\alpha_{k+1} := \beta \alpha_k$$

в каждой итерации до тех пор, пока выбранное условие не будет выполнено.

## i Example

В обучении моделей машинного обучения линейный поиск с возвратом может использоваться для регулировки скорости обучения. Если потеря не уменьшается достаточно, скорость обучения уменьшается мультипликативно до тех пор, пока не будет выполнено условие Армихо.



# Численная иллюстрация

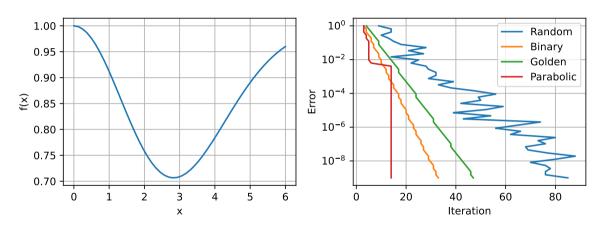


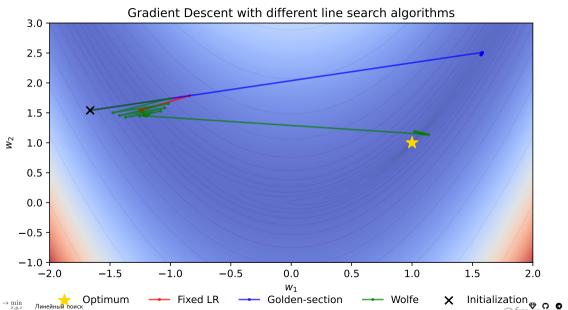
Рисунок 20: Сравнение различных алгоритмов линейного поиска

Открыть в Colab 🦂





# Градиентный спуск с линейным поиском



## Итоги



Итоги



#### Итоги

# Определения

- 1. Унимодальная функция.
- 2. Метод дихотомии.
- 3. Метод золотого сечения.
- 4. Метод параболической интерполяции.
- 5. Условие достаточного убывания для неточного линейного поиска.
- 6. Условия Гольдштейна для неточного линейного поиска.
- 7. Условие ограничения на кривизну для неточного линейного поиска.
- 8. Градиент функции  $f(x): \mathbb{R}^n \to \mathbb{R}$ .
- 9. Гессиан функции  $f(x): \mathbb{R}^n \to \mathbb{R}$ .
- 10. Якобиан функции  $f(x): \mathbb{R}^n \to \mathbb{R}^m$ .
- 11. Формула для аппроксимации Тейлора первого порядка  $f_{x_0}^I(x)$  функции  $f(x): \mathbb{R}^n \to \mathbb{R}$  в точке  $x_0$ .
- 12. Формула для аппроксимации Тейлора второго порядка  $f_{x_0}^{II}(x)$  функции  $f(x):\mathbb{R}^n \to \mathbb{R}$  в точке  $x_0$ .

- 13. Связь дифференциала функции df и градиента  $\nabla f$  для функции  $f(x): \mathbb{R}^n \to \mathbb{R}$ .
- 14. Связь второго дифференциала функции  $d^2f$  и гессиана  $\nabla^2 f$  для функции  $f(x): \mathbb{R}^n \to \mathbb{R}$ .

### Теоремы

1. Метод дихотомии и золотого сечения для унимодальных функций. Скорость сходимости.

