Optimization and Data Science

Lecture 2: Mathematical Basics

Prof. Dr. Thomas Slawig

Kiel University - CAU Kiel
Dep. of Computer Science

Summer 2020

Contents

Contents

Data are numbers

- Computers store bits/bytes, i.e., single values or arrays/vectors of $\{0,1\}$.
- Unsigned Integers ($\subset \mathbb{N}$):

$$z = \sum_{i=0}^{\ell-1} z_i 2^i, z_i \in \{0,1\} \quad \to \quad (z_{\ell-1}, \dots, z_0), \quad z_i \in \{0,1\}. \tag{1}$$

- Signed integers ($\subset \mathbb{Z}$):
  - Non-negative: as in (1), but first bit $z_{\ell-1} = 0$.
  - Negative using **two's complement**:
    represent $|z|$ as in (1), then switch all bits and add binary $(0,\dots,0,1)$.

$\rightsquigarrow$ Example signed integers using two's complement ($\ell = 4$):

$$\begin{array}{lllll} (1000) \mathrel{\hat{=}} -8, & (1001) \mathrel{\hat{=}} -7, & (1010) \mathrel{\hat{=}} -6, & \dots, & (1111) \mathrel{\hat{=}} -1, \\ (0000) \mathrel{\hat{=}} 0, & (0001) \mathrel{\hat{=}} 1, & (0010) \mathrel{\hat{=}} 2 & \dots, & (0111) \mathrel{\hat{=}} 7, \end{array}$$

$\rightsquigarrow$ "Round-up effect": $7 + 1 \mathrel{\hat{=}} (0111) + (0001) = (1000) \mathrel{\hat{=}} -8$.

- Characters/strings are represented sequence of bytes.

Floating point numbers

- Real numbers in $\mathbb{R}$ as normalized floating point numbers.
- Example in the decimal system:

not normalized	normalized
± 12.34	$\pm 1.234 \times 10^1$
± 0.987	$\pm 9.87 \times 10^{-1}$

- ... in the binary system:

not normalized	normalized
± 10.01	$\pm 1.001 \times 2^1$
± 0.01	$\pm 1.0 \times 2^{-2}$

- General form of normalized floating point binary numbers:

$$z = (-1)^s \left(\sum_{i=0}^{\ell_m} m_i 2^{-i} \right) 2^e, \quad s, m_i \in \{0, 1\}, m_0 \neq 0 \text{ (for } z \neq 0).$$

Floating point numbers: IEEE standard

- Normalized floating point binary numbers always start with 1 in front of the floating point:

$$z = (-1)^s \left(\sum_{i=0}^{\ell_m} m_i 2^{-i} \right) 2^e, \quad s, m_i \in \{0,1\}, m_0 \neq 0 \text{ (for } z \neq 0).$$

- **Normalized floating point numbers**, IEEE standard:

$$z = (-1)^s \underbrace{\left(1 + \sum_{i=1}^{\ell_m} m_i 2^{-i} \right)}_{=(1.m_1 \cdots m_{\ell_m})_2} 2^e, \qquad \textbf{sign bit:} \quad s \in \{0,1\}$$

$$\textbf{mantissa:} \quad m = (m_1, \ldots, m_{\ell_m}), \qquad m_i \in \{0,1\}, \quad \ell_m : \text{mantissa length}$$

$$\textbf{exponent:} \quad e = \sum_{i=0}^{\ell_e - 1} e_i 2^i - (2^{\ell_e - 1} - 1), \quad e_i \in \{0,1\}, \quad \ell_e : \text{exponent length}$$

- Leading bit not stored $\rightsquigarrow$ special treatment of $z = 0$. Exponent shift avoids sign bit.

Floating point numbers: IEEE standard

- Normalized floating point numbers:

$$z = (-1)^s \left(1 + \sum_{i=1}^{\ell_m} m_i 2^{-i}\right) 2^e.$$

- Range of the exponent:

$$e_{min} := \underbrace{-(2^{\ell_e - 1} - 1)}_{\text{all } e_i = 0} \leq e = \sum_{i=0}^{\ell_e - 1} e_i 2^i - (2^{\ell_e - 1} - 1)$$

$$\leq \underbrace{\sum_{i=0}^{\ell_e - 1} 2^i - (2^{\ell_e - 1} - 1)}_{\text{all } e_i = 1} = 2_e^{\ell} - 1 - (2^{\ell_e - 1} - 1) = 2^{\ell_e} - 2^{\ell_e - 1} = 2^{\ell_e - 1} =: e_{max}.$$

- Lengths of mantissa and exponent ℓ_m, ℓ_e are fixed.

Floating point numbers: IEEE standard

- Range of the exponent:

$$e_{min} := -(2^{\ell_e - 1} - 1) \leq e \leq 2^{\ell_e - 1} =: e_{max}.$$

- Special cases:

$\rightarrow$ All $e_i = 0, e = e_{min}$: leading bit of mantissa omitted $\rightsquigarrow$ **subnormal numbers**:

$$z = (-1)^s \left(\sum_{i=1}^{\ell_m} m_i 2^{-i} \right) 2^{e_{min}+1}.$$

$\rightarrow$ All $e_i = 1, e = e_{max}$: representation of
  - $\pm\infty$ (numbers too big to represent)
  - NaN (not-a-number): undefined operations $\frac{0}{0}, \infty - \infty$ etc.

- Storing:

$$(s, e_{\ell_e - 1}, \dots, e_0, m_{\ell_m}, \dots, m_1).$$

Floating point numbers: IEEE standard

- IEEE standard: half/single/double (2/4/8 byte): $\ell_m = 10/23/52, \ell_e = 5/8/11$.
- Example: $\ell_m = \ell_e = 2$, gives

$$e = \sum_{i=0}^{1} e_i 2^i - (2^1 - 1) = e_1 \cdot 2 + e_0 - 1 \quad \Rightarrow \quad e_{min} = -1, e_{max} = 2.$$

- Representable machine numbers:

$(e_0, e_1) = \;\Rightarrow$	$(01)_2 = 1$ $e = -1 + 1 = 0$	$(10)_2 = 2$ $e = -1 + 2 = 1$	$(00) = e_{min}$ $e := e_{min} + 1 = 0$	$(11) = e_{max}$
$m = (00)$	$(1.00)_2 \times 2^0 = 1$	$(1.00)_2 \times 2^1 = 2$	$(0.00)_2 = 0$	∞
(01)	$(1.01)_2 \times 2^0 = 1.25$	$(1.01)_2 \times 2^1 = 2.5$	$(0.01)_2 \times 2^0 = 0.25$	NaN
(10)	$(1.10)_2 \times 2^0 = 1.5$	$(1.10)_2 \times 2^1 = 3$	$(0.10)_2 \times 2^0 = 0.5$	NaN
(11)	$(1.11)_2 \times 2^0 = 1.75$	$(1.11)_2 \times 2^1 = 3.5$	$(0.11)_2 \times 2^0 = 0.75$	NaN
	normalized numbers		subnormal numbers	special values

Smallest positive (subnormal) number, biggest representable number

Round-off errors using floating point numbers

- The finiteness of floating point machine numbers introduces **round-off errors**.
- Example: periodic binary number that has to be rounded on the computer:

$$(0.1)_{10} = (0.0\overline{0011})_2.$$

- For *normalized* floating point numbers, the relative round-off error is given by

$$\frac{|x - x_{\mathbb{M}}|}{|x|} \leq eps, \quad x \in \mathbb{R},$$

where: $\mathbb{M}$ is the set of floating point numbers representable on the computer,
$x_{\mathbb{M}}$ the representation of $x \in \mathbb{R}$ as machine number,
the **machine precision**

$$eps := \min\{x_{\mathbb{M}} \in \mathbb{M} : 1 +_{\mathbb{M}} x_{\mathbb{M}} >_{\mathbb{M}} 1 \text{ in machine arithmetic}\}$$

$+_{\mathbb{M}}, >_{\mathbb{M}}$ the operations $+, >$ in machine arithmetic.

- Machine precision, IEEE standards half, single, double: $eps \approx 10^{-4}, 10^{-8}, 10^{-16}$
- ... differs from smallest positive machine number.

Contents

Vectors and matrices

- Convention:

$$x = (x_i)_{i=1}^n = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

  is a column vector.
- The **transposed** vector

$$x^\top = (x_1, \ldots, x_n)$$

  is a row vector.
- Matrix:

$$A = (A_{ij})_{i=1,\ldots,m,j=1,\ldots,n} = \begin{pmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & & \vdots \\ A_{m1} & \cdots & A_{mn} \end{pmatrix} \in \mathbb{R}^{m\times n}$$

Matrix-vector multiplication

- For a matrix and a vector

$$A = (A_{ij})_{i=1,\dots,m,j=1,\dots,n} = \begin{pmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & & \vdots \\ A_{m1} & \cdots & A_{mn} \end{pmatrix} \in \mathbb{R}^{m\times n}, \quad x = (x_i)_{i=1}^n = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

  we define the **matrix-vector product**

$$Ax := \left(\sum_{j=1}^{n} A_{ij} x_j \right)_{i=1}^{m} \in \mathbb{R}^m$$

- The mapping $x \mapsto Ax$ is a linear function from $\mathbb{R}^n$ to $\mathbb{R}^m$.
- Special case: **inner** or **scalar product**: $x^\top y := \sum_{i=1}^{n} x_i y_i \in \mathbb{R}$ for $x, y \in \mathbb{R}^n$.

Matrix-matrix multiplication

- For two matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, we define the **matrix product** $AB \in \mathbb{R}^{m\times p}$ by

$$(AB)_{ik} := \sum_{j=1}^{n} A_{ij} B_{jk}, \quad i = 1, \dots, m, k = 1, \dots, p.$$

- (Inner) matrix dimension n has to match!
- Matrix product does **not commute**, i.e., in general

$$AB \neq BA.$$

- Special matrices: **diagonal matrices**:

$$D = (D_{ij})_{i,j=1}^{n} = \mathrm{diag}(d_1, \dots, d_n) \text{ with } D_{ij} = \left\{ \begin{array}{ll} d_i, & i = j \\ 0, & i \neq j \end{array} \right\}$$

- $DA \rightsquigarrow$ multiplication of i-th **row** of A by D_{ii}.
- $AD \rightsquigarrow$ multiplication of i-th **column** of A by D_{ii}.

Contents

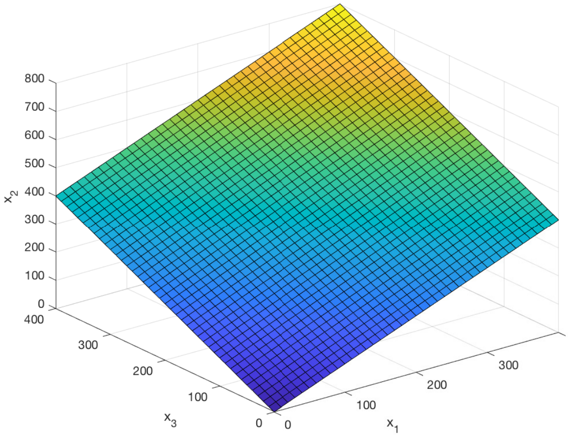
Elementary Functions

- Linear functions

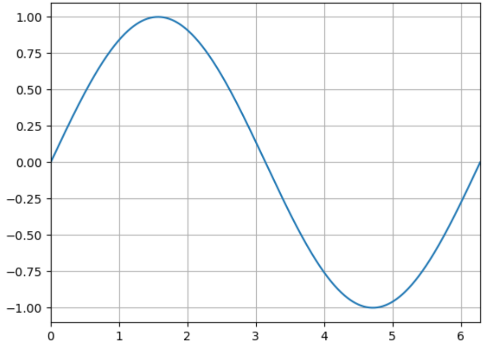
$$
\begin{aligned}
f : \mathbb{R} \to \mathbb{R} : \quad & f(x) = ax, \quad a \in \mathbb{R} \\
f : \mathbb{R}^n \to \mathbb{R}^m : \quad & f(x) = Ax, \quad A \in \mathbb{R}^{m \times n}
\end{aligned}
$$

- **Linearity** is defined as:

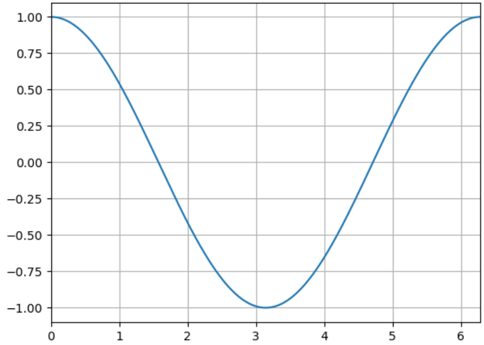
$$
\begin{aligned}
f(x + y) &= f(x) + f(y), \quad && x, y \in \mathbb{R}^m \\
f(\alpha x) &= \alpha f(x), && x \in \mathbb{R}^m, \alpha \in \mathbb{R}.
\end{aligned}
$$



Trigonometric functions

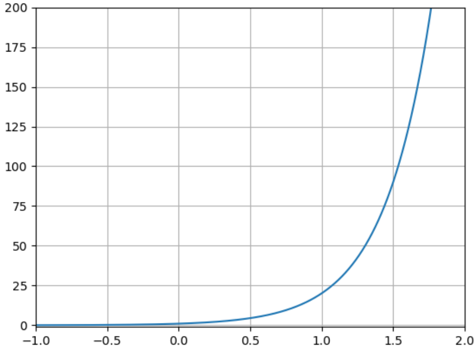


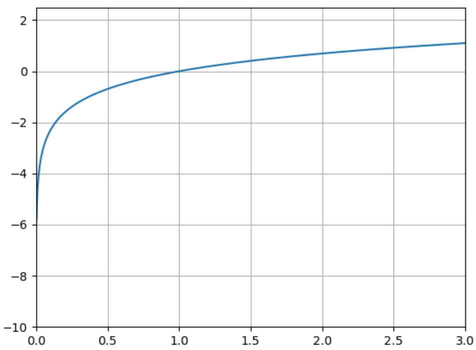
sine function $f(x) = \sin x$



cosine function $f(x) = \cos x$

Exponential and logarithm





Exponential function $f(x) = \exp(x) = e^x$

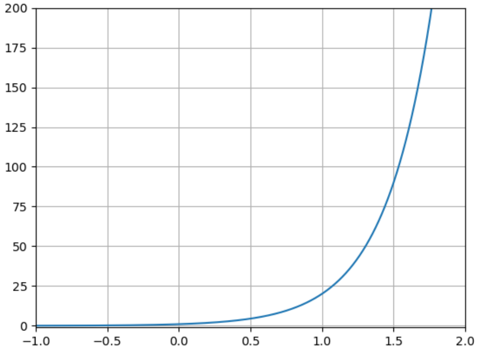
logarithmic function $f(x) = \log x$

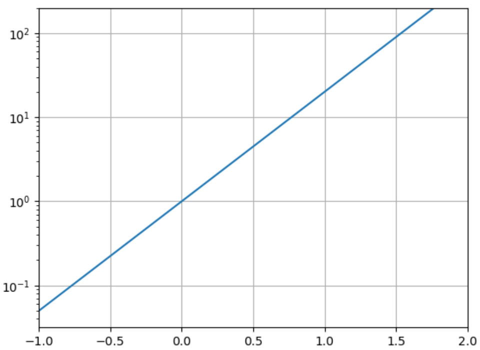
Exponential rules: $x^a x^b = x^{a+b}, (x^a)^b = x^{ab}$,

logarithmic rule: $\log(x^a) = a \log x$.

Describing exponential growth using a logarithmic plot

- Picture on the left: exponential growth $f(x) = x^a$ with $a \in \mathbb{R}$ unknown.
- Using a logarithmic scale on the vertical axis allows to visualize a, since $\log(x^a) = a \log x$. (picture on the right).





Contents

Global minimizers

Definition

- A point x^* is called a **global minimizer** (of f in X_{ad}) if

$$f(x^*) \leq f(x) \text{ for all } x \in X_{ad}. \tag{2}$$

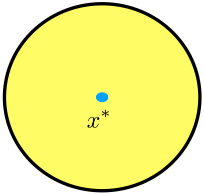
- Then, the value $f(x^*)$ is called the global **minimum**.
- A minimizer/minimum is called **strict**, if the inequality is strict in (2).

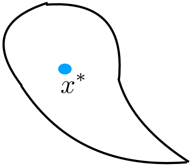
Local minimizers

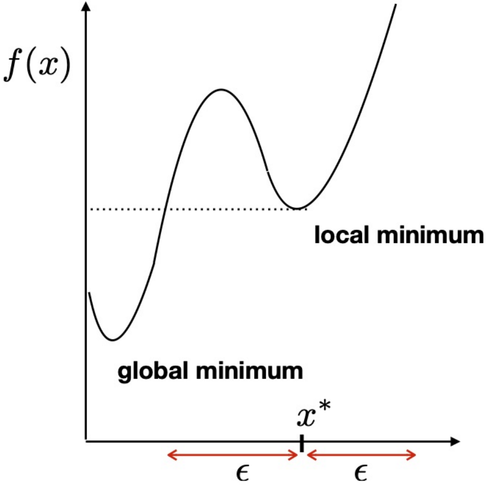
- Often, we can only compute local minima and minimizers.
- Then, the inequality

$$f(x^*) \leq f(x)$$

is only valid for all x in a "neighborhood" of x^*.

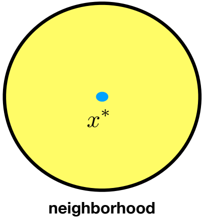






Distance measure: metric

- To define a neighborhood, we need a distance measure in X.
- Such kind of distance measure is called metric.

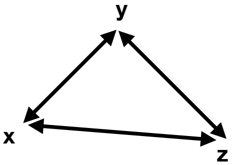


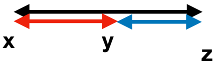
Definition

- A mapping $d : X \times X \to \mathbb{R}_{\geq 0}$ is called **metric** on X, if

$$\left.\begin{array}{rcl} d(x,y) & = & d(y,x) \\ d(x,z) & \leq & d(x,y) + d(y,z) \\ d(x,y) = 0 & \Leftrightarrow & x = y \end{array}\right\} \text{ for all } x,y,z \in X.$$

- We then call the pair (X, d) a **metric space**.





Neighborhood using a metric

- Let X, d be a metric space. We call

$$B_\epsilon(x^*) := \{x \in X : d(x, x^*) < \epsilon\}$$

the **open ball** around x^* with radius $\epsilon > 0$.

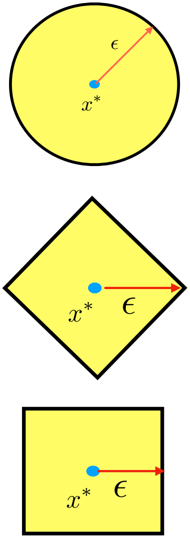
- Metrics in $X = \mathbb{R}^n$:

$$d(x, y) := \|x - y\|_2 := \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \text{ \textbf{(Euclidean norm)}}$$

$$d(x, y) := \|x - y\|_1 := \sum_{i=1}^{n} |x_i - y_i|$$

$$d(x, y) := \|x - y\|_\infty := \max_{i=1,\dots,n} |x_i - y_i|.$$

- Picture: balls $B_\epsilon(x^*)$ for these three metrics.



Local minimizers and minima

Now we define local minimizers/minima using a metric:

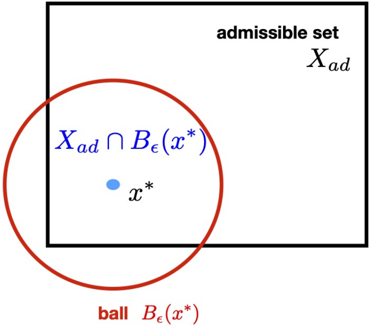
Definition

Let (X, d) be a metric space and $X_{ad} \subset X$.

- A point x^* is called a **local minimizer** (of f in X_{ad}) if

$$f(x^*) \leq f(x) \text{ for all } x \in X_{ad} \cap B_\epsilon(x^*) \text{ for some } \epsilon > 0.$$

- We also write $x^* = \arg\min_{x \in X_{ad}} f(x)$ for a local minimizer.



Mathematical basics: What is important

- Data on the computer are numbers $\rightsquigarrow$ helpful to know how they are stored,
- ... especially for floating point numbers.
- Important to know the error in the representation.
- We need some elementary functions
- ... and basics from linear algebra: vectors, matrices and their basic operations.
- For optimization problems, we need to define minimizers and minima.
- We distinguish between local and global ones.
- To define locality, we need to have a measure of distance between different points (i.e., vectors) in the multi-dimensional parameter space.
- Such kind of measure is called metric.
- A metric can be constructed by using different measures of vector lengths (that we call norms).