Numerical Analysis

Operations Research in R

Stefan Feuerriegel

Today's Lecture

Objectives

1. Understanding how computers store and handle numbers
2. Repeating basic operations in linear algebra and their use in R
3. Recapitulating the concept of derivatives and the Taylor approximation
4. Formulating necessary and sufficient conditions for optimality

Outline

Outline

Positional Notation

- Method of representing numbers
- Same symbol for different orders of magnitude ($\neq$ Roman numerals)
- Format is $d_n d_{n-1} \dots d_2 d_1 = d_n \cdot b^{n-1} + d_{n-1} \cdot b^{n-2} + \dots + d_2 \cdot b + d_1$ with

b	base of the number
n	number of digits
d	digit in the i-th position of the number

- Example: 752 is $7_3 \cdot 10^2 + 5_2 \cdot 10 + 2_1$

Base Conversions

- Numbers can be converted between bases
  - Base 10 is default
  - Binary system with base 2 common for computers
- Example: 752 in base 10 equals 1 011 110 000 in base 2
- Conversion from base b into base 10 via

$$d_n \cdot b^{n-1} + d_{n-1} \cdot b^{n-2} + \ldots + d_2 \cdot b + d_1$$

- Example: 101 101 011 in base 2

$$\begin{aligned} & 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \\ = & 1 \cdot 256 + 0 \cdot 128 + 1 \cdot 64 + 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \\ = & 363 \text{ in base 10} \end{aligned}$$

Base Conversions

Question

- Convert the number 10 011 010 from base 2 into base 10
  - 262
  - 138
  - 154
- Visit webpage with course quiz.

Question

- Convert the number 723 from base 10 into base 2
  - 111 010 011
  - 10 011 010 011
  - 1 011 010 011
- Visit webpage with course quiz.

Base Conversions

Question

- Convert the number 10 011 010 from base 2 into base 10
  - 262
  - 138
  - 154
- Visit webpage with course quiz.

Question

- Convert the number 723 from base 10 into base 2
  - 111 010 011
  - 10 011 010 011
  - 1 011 010 011
- Visit webpage with course quiz.

Base Conversions

- Conversion scheme of number s_1 from base 10 into base b:

Start	Integer part of division	Remainder
s_1	$s_2 := \left\lfloor \frac{s_1}{b} \right\rfloor$	$r_1 := s_1 \mod b$
s_2	$s_3 := \left\lfloor \frac{s_2}{b} \right\rfloor$	$r_2 := s_2 \mod b$
$\dots$		
s_m	$\left\lfloor \frac{s_m}{b} \right\rfloor \overset{!}{=} \mathbf{0}$	$r_m := s_m \mod b$

- The result is $r_m \dots r_2 r_1$

Base Conversions

Example: convert 363 from base 10 into base 2

- Calculation steps:

Start	Integer division by 2	Remainder
363	181	1
181	90	1
90	45	0
45	22	1
22	11	0
11	5	1
5	2	1
2	1	0
1	0	1

- Result: 101 101 011 in base 2

Base Conversions in R

- Load necessary library `sfsmisc`

```
library(sfsmisc)
```

- Call function `digitsBase(s, base=b)` to convert *s* into base *b*

```
# convert the number 450 from base 10 into base 8
digitsBase(450, base=8)

## Class 'basedInt'(base = 8) [1:1]
##       [,1]
## [1,]     7
## [2,]     0
## [3,]     2
```

- Call `strtoi(d, base=b)` to convert *d* from base *b* into base 10

```
# convert the number 10101 from base 2 into base 10
strtoi(10101, base=2)

## [1] 21
```

Floating-Point Representation

- Floating point is the representation to approximate real numbers in computing

$$(-1)^{\text{sign}} \cdot \text{significand} \cdot \text{base}^{\text{exponent}}$$

- Significand and exponent have a fixed number of digits
- More digits for the significand (or mantissa) increase accuracy
- The exponent controls the range of numbers
- Examples

$$\begin{aligned} 256.78 &\quad\rightarrow\quad +2.5678 \cdot 10^2 \\ -256.78 &\quad\rightarrow\quad -2.5678 \cdot 10^2 \\ 0.00365 &\quad\rightarrow\quad +3.65 \cdot 10^{-3} \end{aligned}$$

- Very large and very small numbers are often written in scientific notation (also named E notation)
$\rightarrow$ e. g. $2.2\text{e}6 = 2.2 \cdot 10^6 = 2\,200\,000$, $3.4\text{e}{-2} = 0.034$

Limited Precision of Floating-Point Numbers

- The limited precision of a computer leads false results

```
x <- 10^30 + 10^(-20)
x - 10^30

## [1] 0

sin(pi) == 0

## [1] FALSE

3 - 2.9 == 0.1

## [1] FALSE
```

Limited Precision of Floating-Point Numbers

- Workaround is to use `round(x)` but this cuts all non-integer digits

```
round(sin(pi))

## [1] 0
```

- A better method is to use a tolerance for the comparison

```
a <- 3 - 2.9
b <- 0.1
tol <- 1e-10
abs(a - b) <= tol

## [1] TRUE
```

- Numbers that are too large can cause an overflow

```
2*10^900

## [1] Inf
```

Outline

Dot Product

- The dot product (or scalar product) takes two equal-size vectors and returns a scalar, as defined by

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^T \boldsymbol{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n$$

  with $\boldsymbol{a} = [a_1, a_2, \ldots, a_n]^T \in \mathbb{R}^n$ and $\boldsymbol{b} = [b_1, b_2, \ldots, b_n]^T \in \mathbb{R}^n$
- Usage in R via the operator %*%

```
A <- c(1, 2, 3)
B <- c(4, 5, 6)
A %*% B

##      [,1]
## [1,]   32

# deletes dimensions which have only one value
drop(A %*% B)

## [1] 32
```

Properties of the Dot Product

- Cummutative

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{b} \cdot \boldsymbol{a}$$

- Distributive over vector addition

$$\boldsymbol{a} \cdot (\boldsymbol{b} + \boldsymbol{c}) = \boldsymbol{a} \cdot \boldsymbol{b} + \boldsymbol{a} \cdot \boldsymbol{c}$$

- Bilinear

$$\boldsymbol{a} \cdot (r\boldsymbol{b} + \boldsymbol{c}) = r(\boldsymbol{a} \cdot \boldsymbol{b}) + \boldsymbol{a} \cdot \boldsymbol{c} \quad \text{with } r \in \mathbb{R}$$

- Scalar multiplication

$$(r_1\boldsymbol{a}) \cdot (r_2\boldsymbol{b}) = r_1 r_2 (\boldsymbol{a} \cdot \boldsymbol{b}) \quad \text{with } r_1, r_2 \in \mathbb{R}$$

- Two non-zero vector $\boldsymbol{a}$ and $\boldsymbol{b}$ are orthogonal if and only if $\boldsymbol{a} \cdot \boldsymbol{b} = 0$

(Vector) Norm

- The norm is a real number which gives us information about the "length" or "magnitude" of a vector

- It is defined as $\|\cdot\| \mapsto \mathbb{R}^{\geq 0}$ such that

  1. $\|\boldsymbol{x}\| > 0$ if $\boldsymbol{x} \neq [0, \ldots, 0]^T$ and $\|\boldsymbol{x}\| = 0$ if and only if $\boldsymbol{x} = [0, \ldots, 0]^T$
  2. $\|r\boldsymbol{x}\| = \|r\| \, \|\boldsymbol{x}\|$ for any scalar $r \in \mathbb{R}$
  3. $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$

- This definition is highly abstract, many variants exist

- The so-called inner product $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ is a generalization to abstract vector spaces over a field of scalars (e. g. $\mathbb{C}$)

Common Variants of Vector Norms

- The absolute-value norm equals the absolute value, i. e.

$$\|x\| = |x| \quad \text{for } x \in \mathbb{R}$$

- The Euclidean norm (or L^2-norm) is the intuitive notion of length

$$\|\boldsymbol{x}\|_2 = \sqrt{\boldsymbol{x} \cdot \boldsymbol{x}} = \sqrt{x_1^2 + \ldots x_n^2}$$

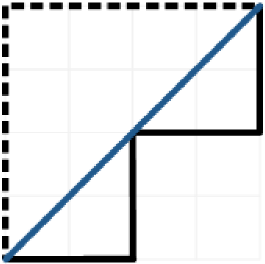
- The Manhattan norm (or L^1-norm) is the distance on a rectangular grid

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

- Their generalization is the p-norm

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \quad \text{for } p \geq 1$$

L^1- vs. L^2-Norm



Blue $\rightarrow$ Euclidean distance

Black $\rightarrow$ Manhattan distance

Question

- What is the distance $d =$ bottom left $\rightarrow$ top right in L^1- and L^2-norm?
  - $\|d\|_1 = 8$, $\|d\|_2 = 16$
  - $\|d\|_1 = 8$, $\|d\|_2 = \sqrt{32}$
  - $\|d\|_1 = \sqrt{32}$, $\|d\|_2 = 8$
- Visit webpage with course quiz.

Vector Norms in R

- No default built-in function, instead calculate the L^1- and L^2-norm manually

```
x <- c(1, 2, 3)
sum(abs(x)) # L1-norm
```

```
## [1] 6
```

```
sqrt(sum(x^2)) # L2-norm
```

```
## [1] 3.741657
```

- The p-norm needs to be computed as follows

```
(sum(abs(x)^3))^(1/3) # 3-norm
```

```
## [1] 3.301927
```

Scalar Multiplication

- Definition: $\lambda \boldsymbol{x} = \boldsymbol{x}\lambda = \begin{bmatrix} \lambda x_1 \\ \vdots \\ \lambda x_n \end{bmatrix}$ $\quad \lambda A = A\lambda = \begin{bmatrix} \lambda a_{11} & \cdots & \lambda a_{1m} \\ \vdots & \ddots & \vdots \\ \lambda a_{n1} & \cdots & a_{nm} \end{bmatrix}$
- Use the default multiplication operator $*$

```
5*c(1, 2, 3)

## [1]  5 10 15

m <- matrix(c(1,2, 3,4, 5,6), ncol=3)
m

##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6

5*m

##      [,1] [,2] [,3]
## [1,]    5   15   25
## [2,]   10   20   30
```

Transpose

- The transpose of a matrix A is another matrix A^T where the values in columns and rows are flipped

$$A^T := [a_{ji}]_{ij}$$

- Example: $\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
- Transpose via `t(A)`

```
m

##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6

t(m)

##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
## [3,]    5    6
```

Matrix-by-Vector Multiplication

- Definition:

$$A\boldsymbol{x} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_n \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nm}x_n \end{bmatrix}$$

  with $A \in \mathbb{R}^{n \times m}$, $\boldsymbol{x} \in \mathbb{R}^n$ and $A\boldsymbol{x} \in \mathbb{R}^n$
- Use operator %*% in R

```
m %*% x

##      [,1]
## [1,]   22
## [2,]   28
```

Element-Wise Matrix Multiplication

- For matrices $A \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{n \times m}$, it returns a matrix $C \in \mathbb{R}^{n \times m}$ of defined as

$$c_{ij} = a_{ij} b_{ij}$$

- The default multiplication operator $*$ performs an element-wise multiplication

```
m
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
m*m
```

```
##      [,1] [,2] [,3]
## [1,]    1    9   25
## [2,]    4   16   36
```

Matrix-by-Matrix Multiplication

- Given matrices $A \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{m \times l}$, then the matrix multiplication obtains $C = AB \in \mathbb{R}^{n \times l}$, defined by

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}$$

- It is implemented by the operator %*%

```
m

##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
t(m)

##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
## [3,]    5    6
```

```
m %*% t(m)

##      [,1] [,2]
## [1,]   35   44
## [2,]   44   56
```

Identity Matrix

- The identity matrix

  $$I_n = \text{diag}(1, 1, \ldots, 1) \in \mathbb{R}^{n \times n}$$

  is a square matrix with 1s on the diagonal and 0s elsewhere
- It fulfills

  $$I_n A = A I_m = A$$

  given a matrix $A \in \mathbb{R}^{n \times m}$
- The command `diag(n)` creates an identity matrix of size $n \times n$

```
diag(3)

##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

Matrix Inverse

- The inverse of a square matrix A is a matrix A^{-1} such that

$$AA^{-1} = I \qquad \text{(note that generally this is } \neq A^{-1}A\text{)}$$

- A square matrix has an inverse if and only if its determinant $\det A \neq 0$
- The direct calculation is numerically highly unstable, and thus one often rewrites the problem to solve a system of linear equations

Matrix Inverse in R

- `solve()` calculates the inverse A^{-1} of a square matrix A

```
sq.m <- matrix(c(1,2, 3,4), ncol=2)
sq.m

##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4

solve(sq.m)

##      [,1] [,2]
## [1,]   -2  1.5
## [2,]    1 -0.5

sq.m %*% solve(sq.m) - diag(2) # post check

##      [,1] [,2]
## [1,]    0    0
## [2,]    0    0
```

Pseudoinverse

- The pseudoinverse $A^+ \in \mathbb{R}^{m \times n}$ is a generalization of the inverse of a matrix $A \in \mathbb{R}^{n \times m}$; fulfilling among others

$$AA^+ = I$$

- `ginv(A)` inside the library `MASS` calculates the pseudoinverse

```
library(MASS)
```

```
ginv(m)

##            [,1]       [,2]
## [1,] -1.3333333  1.0833333
## [2,] -0.3333333  0.3333333
## [3,]  0.6666667 -0.4166667

m %*% ginv(m)

##              [,1] [,2]
## [1,] 1.000000e+00    0
## [2,] 2.664535e-15    1
```

- If AA^+ is invertible, it is given by

$$A^+ := A^T \left(AA^T\right)^{-1}$$

Determinant

- The determinant $\det A$ is a useful value for a square matrix A, relating to e. g. the region it spans
- A square matrix is also invertible if and only if $\det A \neq 0$

Calculation

- The determinant of of a 2×2 matrix A is defined by

$$\det A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11} a_{22} - a_{12} a_{21}$$

- A similar simple rule exists for matrices of size 3×3, for all others one usually utilizes the Leibniz or the Laplace formula
- Calculation in R is via `det(A)`

```
det(sq.m)

## [1] -2
```

Eigenvalues and Eigenvectors

- An eigenvector $\boldsymbol{v}$ of a square matrix A is a vector that does not change its direction under the linear transformation by $A \in \mathbb{R}^{n \times n}$
- This is given by

$$A\boldsymbol{v} = \lambda \boldsymbol{v} \qquad \text{for } \boldsymbol{v} \neq [0, \ldots 0]^T \in \mathbb{R}^n$$

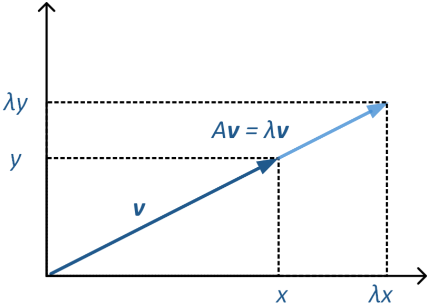
  where $\lambda \in \mathbb{R}$ is the eigenvalue associated with the eigenvector $\boldsymbol{v}$

- Example: the matrix $A = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 2 \end{bmatrix}$ has the following eigenvectors and eigenvalues

$$\lambda_1 = 1, \boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \quad \lambda_2 = 2, \boldsymbol{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \lambda_3 = 3, \boldsymbol{v}_3 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Eigenvalues and Eigenvectors

Geometric interpretation



Matrix A stretches the vector $\boldsymbol{v}$ but does not change its direction

$\rightarrow$ $\boldsymbol{v}$ is an eigenvector of A

Eigenvalues and Eigenvectors

Question

- Given $A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 2 & 3 & 3 \end{bmatrix}$
- Which of the following is not an eigenvector/eigenvalue pair?
  - $\lambda_1 = 1, \boldsymbol{v}_1 = \begin{bmatrix} \frac{2}{3} \\ -\frac{2}{3} \\ \frac{1}{3} \end{bmatrix}$
  - $\lambda_2 = 2, \boldsymbol{v}_2 = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$
  - $\lambda_3 = 3, \boldsymbol{v}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
- Visit webpage with course quiz.

Eigenvalues and Eigenvectors in R

- Eigenvalues and eigenvectors of a square matrix A via `eigen(A)`

```
sq.m

##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4

e <- eigen(sq.m)
e$val # eigenvalues

## [1]  5.3722813 -0.3722813

e$vec # eigenvectors

##            [,1]       [,2]
## [1,] -0.5657675 -0.9093767
## [2,] -0.8245648  0.4159736
```

Definiteness of Matrices

- The definiteness of a matrix helps in determining the nature of optima
- Definitions
  - The symmetric matrix $A \in \mathbb{R}^{n\times n}$ is positive definite if

$$\boldsymbol{x}^T A\boldsymbol{x} > 0 \quad \text{for all } \boldsymbol{x} \neq [0, \ldots, 0]^T$$

  - The symmetric matrix $A \in \mathbb{R}^{n\times n}$ is positive semi-definite if

$$\boldsymbol{x}^T A\boldsymbol{x} \geq 0 \quad \text{for all } \boldsymbol{x} \neq [0, \ldots, 0]^T$$

Example

The identity matrix $I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is positive definite, since

$$\boldsymbol{x}^T I_2 \boldsymbol{x} = [x_1, x_2] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1^2 + x_2^2 > 0 \text{ for all } \boldsymbol{x} \neq [0, 0]^T$$

Positive Definiteness

- Tests for positive definiteness
  - Evaluating $\boldsymbol{x}^T A \boldsymbol{x}$ for all $\boldsymbol{x}$ is impractical
  - All eigenvalues λ_i of A are positive
  - Check if all upper-left sub-matrices have positive determinants (Sylvester's criterion)

Definiteness Tests in R

The library `matrixcalc` offers methods to test all variants of definiteness

```
library(matrixcalc)
```

```
I <- diag(3)
I
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

```
is.negative.definite(I)
```

```
## [1] FALSE
```

```
is.positive.definite(I)
```

```
## [1] TRUE
```

```
C <- matrix(c(-2,1,0, 1,-2,1, 0,1,-2),
            nrow=3, byrow=TRUE)
C
```

```
##      [,1] [,2] [,3]
## [1,]   -2    1    0
## [2,]    1   -2    1
## [3,]    0    1   -2
```

```
is.positive.semi.definite(C)
```

```
## [1] FALSE
```

```
is.negative.semi.definite(C)
```

```
## [1] TRUE
```

Outline

Differentiability

Definition

Let $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ be a function and $x_0 \in D$

- f is differentiable at the point x_0 if the following limit exists

$$f'(x_0) = \frac{\mathrm{d}f}{\mathrm{d}x}(x_0) = \lim_{\varepsilon \to 0} \frac{f(x_0 + \varepsilon) - f(x_0)}{\varepsilon}$$

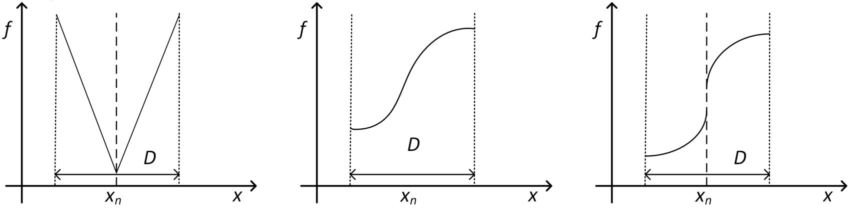
  the limit $f'(x_0)$ is called the derivative of f at the point x_0
- If it is differentiable for all $x \in D$, then f is differentiable with derivative f'

Remarks

- Similarly, the 2nd derivative f'' and, by induction, the n-th derivative $f^{(n)}$
- Geometrically, $f'(x_0)$ is the slope of the tangent to $f(x)$ at x_0

Differentiability

Examples



continuous
differentiable

continuous
not differentiable

discontinuous
not differentiable

Question

- What is correct for the function $f(x) = \frac{2x-1}{x+2}$?
  - Continuous and differentiable
  - Continuous but not differentiable
  - Discontinuous and not differentiable
- Visit webpage with course quiz.

Chain Rule

Let $v(x)$ be a differentiable function, then the chain rule gives

$$\frac{\mathrm{d}u(v(x))}{\mathrm{d}x} = \frac{\mathrm{d}u}{\mathrm{d}v}\frac{\mathrm{d}v}{\mathrm{d}x}$$

Example Given $u(v(x)) = \sin(\pi x)$, then $u = \sin$, $v(x) = \pi x$ and

$$\frac{\mathrm{d}u(v(x))}{\mathrm{d}x} = \frac{\mathrm{d}\sin(\pi x)}{\mathrm{d}v}\frac{\mathrm{d}(\pi x)}{\mathrm{d}x} = \cos(\pi x)\pi$$

Question

- What is the derivative of $\log 4 - x$?
  - $\frac{1}{x-4}$
  - $\frac{4}{x}$
  - $\frac{1}{4-x}$
- Visit webpage with course quiz.

Partial Derivative

- The partial derivative with respect to x_i is given by

$$\frac{\partial f}{\partial x_i}(\boldsymbol{x}) := \lim_{\varepsilon \to 0} \frac{f(x_1, \ldots, x_i + \varepsilon, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n)}{\varepsilon}$$

- f is called partially differentiable, if f is differentiable at each point with respect to all variables
- Partial derivatives can be exchanged in their order

$$\frac{\partial}{\partial x_i}\left(\frac{\partial f}{\partial x_j}\right) = \frac{\partial}{\partial x_j}\left(\frac{\partial f}{\partial x_i}\right)$$

Derivatives in R

- The function `D(f, "x")` derives an expression *f* symbolically

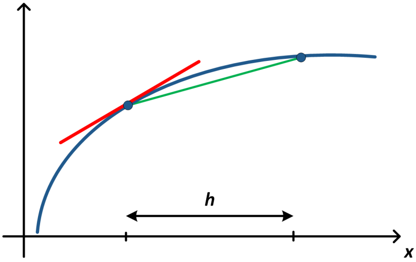
```
f <- expression(x^5 + 2*y^3 + sin(x) - exp(y))

D(f, "x")

## 5 * x^4 + cos(x)

D(D(f, "y"),"y")

## 2 * (3 * (2 * y)) - exp(y)

D(D(f, "x"),"y")

## [1] 0
```

- To compute the derivative at a specific point, we use `eval(expr)`

```
eval(D(f, "x"), list(x=2, y=1))

## [1] 79.58385
```

Finite Differences

- Numerical methods to approximate derivatives numerically
- Use a step size h, usually of order 10^{-6}



  - Forward differences

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

  - Backward differences

$$f'(x) = \frac{f(x) - f(x-h)}{h}$$

  - Centered differences

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

Higher-Order Differences

Use the previous formulae to derive 2nd order central differences

$$
\begin{aligned}
f''(x) &\approx \frac{f'(x+h) - f'(x)}{h} \\
&\approx \frac{\dfrac{f(x+h) - f(x)}{h} - \dfrac{f(x) - f(x-h)}{h}}{h} \\
&= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}
\end{aligned}
$$

Finite Differences in R

Question

- Given $f(x) = \sin x$
- Set `h <- 10e-6`
- How to calculate the derivative at $x = 2$ with centered differences in R?
  - `(sin(2+h) - sin(2-h)) / (2*h)`
  - `(sin(2+h) - sin(2-h)) / 2*h`
  - `(sin(2+h) - sin(2)) / (2*h)`
- Visit webpage with course quiz.

Gradient and Hessian Matrix

- Gradient of $f : \mathbb{R}^n \to \mathbb{R}^n$

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\boldsymbol{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\boldsymbol{x}) \end{bmatrix}$$

- The second derivatives of f are called the Hessian (matrix)

$$H(\boldsymbol{x}) = \nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(\boldsymbol{x}) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\boldsymbol{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\boldsymbol{x}) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(\boldsymbol{x}) \end{bmatrix}$$

- Since the order of derivatives can be exchanged, the Hessian $H(\boldsymbol{x})$ is symmetric, i. e. $H(\boldsymbol{x}) = (H(\boldsymbol{x}))^T$

Hessian Matrix in R

- `optimHess(x, f, ...)` approximates the Hessian matrix of f

```
f <- function(x) (x[1]^3*x[2]^2 - x[2]^2 + x[1])
optimHess(c(3, 2), f, control=(ndeps=0.0001))

##      [,1] [,2]
## [1,]   72  108
## [2,]  108   52
```

- Above example: forward differences to approximate the Hessian Matrix of $f(x_1, x_2)$ at a given point $(x_1, x_2) = (3, 2)$ with a given step size $h = 0.0001$

Outline

Taylor Series

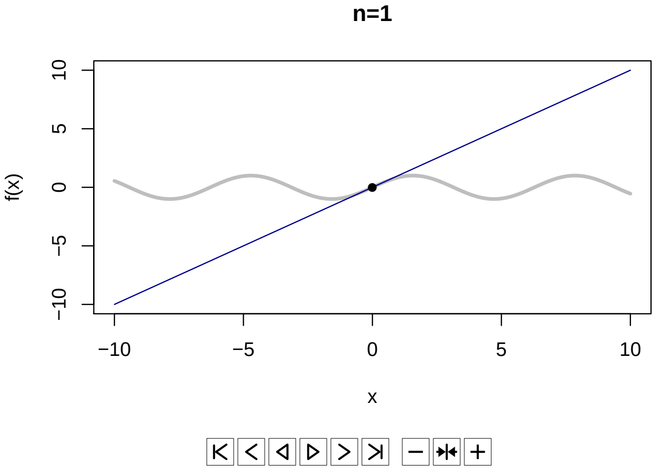
- Simple polynomial approximation to almost arbitrary functions
- Taylor series approximates f around a point x_0 as a power series

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(x_0)}{3!}(x - x_0)^3 + \dots$$
$$= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n.$$

- f must be infinitely differentiable
- If $x_0 = 0$ the series is also called Maclaurin series
- To obtain an approximation of f, cut off after order n

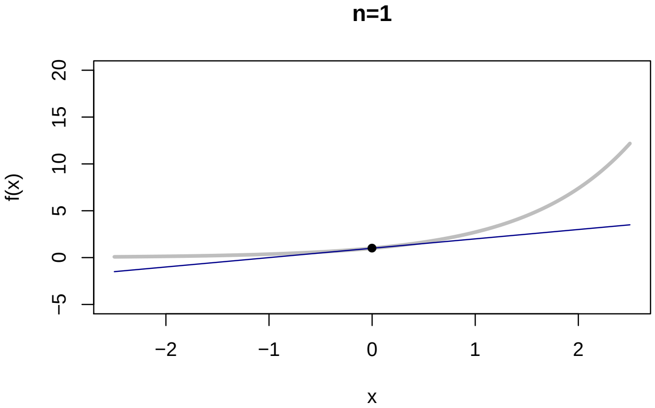
Taylor Approximation

Approximation of order n (blue) around $x_0 = 0$ for $f(x) = \sin x$ (in gray)



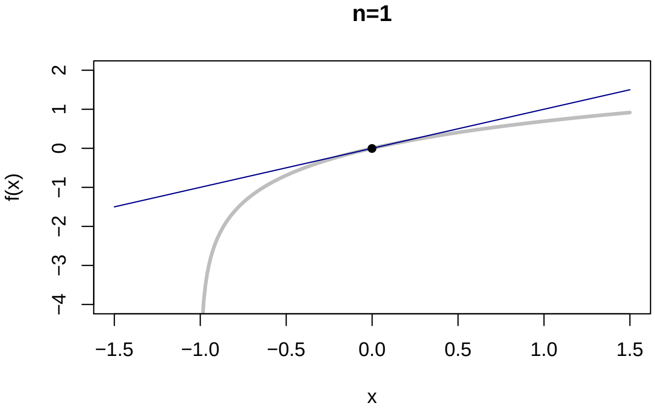
Taylor Approximation

Approximation of order n (blue) around $x_0 = 0$ for $f(x) = e^x$ (in gray)



Taylor Approximation

Approximation of order n (blue) around $x_0 = 0$ for $f(x) = \log x + 1$ (in gray)



Taylor Series

Question

- What is the Taylor series for $f(x) = \dfrac{1}{1-x}$ with $x_0 = 0$?
  - $f(x) = \frac{1}{x} + 1 + x + x^2 + x^3 + \dots$
  - $f(x) = 1 + x + x^2 + x^3 + \dots$
  - $f(x) = x + x^2 + x^3 + \dots$
- Visit webpage with course quiz.

Question

- What is the Taylor series for $f(x) = e^x$ with $x_0 = 0$?
  - $f(x) = \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$
  - $f(x) = 1 + \frac{x}{1} + \frac{x^2}{2} + \frac{x^3}{3} + \dots$
  - $f(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$
- Visit webpage with course quiz.

Taylor Series

Question

- What is the Taylor series for $f(x) = \dfrac{1}{1-x}$ with $x_0 = 0$?
  - $f(x) = \frac{1}{x} + 1 + x + x^2 + x^3 + \ldots$
  - $f(x) = 1 + x + x^2 + x^3 + \ldots$
  - $f(x) = x + x^2 + x^3 + \ldots$
- Visit webpage with course quiz.

Question

- What is the Taylor series for $f(x) = e^x$ with $x_0 = 0$?
  - $f(x) = \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$
  - $f(x) = 1 + \frac{x}{1} + \frac{x^2}{2} + \frac{x^3}{3} + \ldots$
  - $f(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$
- Visit webpage with course quiz.

Taylor Approximation with R

- Load library `pracma`

```
library(pracma)
```

- Calculate approximation up to degree 4 with `taylor(f, x0, n)`

```
f <- function(x) cos(x)
taylor.poly <- taylor(f, x0=0, n=4)
taylor.poly

## [1]  0.04166733  0.00000000 -0.50000000  0.00000000  1.00000000
```

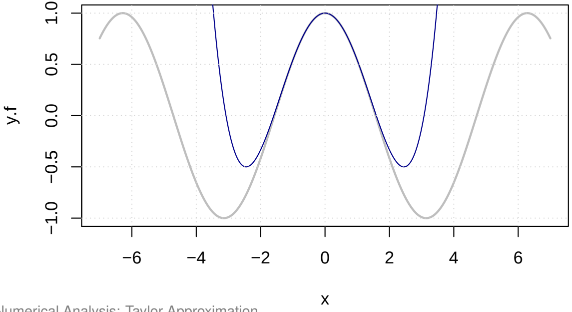
- Evaluate Taylor approximation `p` at x with `polyval(p, x)`

```
polyval(taylor.poly, 0.1) # x = 0.1

## [1] 0.9950042

cos(0.1) # for comparison

## [1] 0.9950042

polyval(taylor.poly, 0.5) - cos(0.5)

## [1] 2.164622e-05
```

Taylor Approximation in R

Visualizing Taylor approximation

```
x <- seq(-7.0, 7.0, by=0.01)
y.f <- f(x)
y.taylor <- polyval(taylor.poly, x)
plot(x, y.f, type="l", col="gray", lwd=2, ylim=c(-1, +1))
lines(x, y.taylor, col="darkblue")
grid()
```



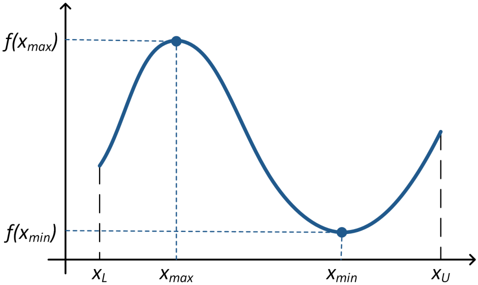
Outline

Extreme Value Theorem

Theorem

- Given: real-valued function f
- f continuous in the closed and bounded interval $[x_L, x_U]$
- Then f must attain a maximum and minimum at least once
- I. e. there exists $x_{\max}, x_{\min} \in [x_L, x_U]$ such that

$$f(x_{\max}) \geq f(x) \geq f(x_{\min}) \qquad \text{for all } x \in [x_L, x_U]$$



Optimum

Definitions

- x^* is a local minimum if $x^* \in D$ and if there is a neighborhood $N(x^*)$, such that

$$f(x^*) \leq f(x) \qquad \text{for all } x \in N(x^*) \subseteq D$$

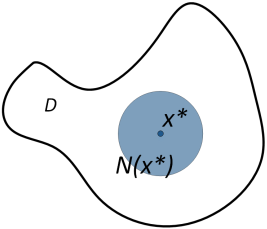
- x^* is a strict local minimum if $x^* \in D$ and if there is a neighborhood $N(x^*)$, such that

$$f(x^*) < f(x) \qquad \text{for all } x \in N(x^*) \subseteq D$$

- x^* is a global minimum if $x^* \in D$ and

$$f(x^*) \leq f(x) \qquad \text{for all } x \in D$$

$\rightarrow$ What conditions need to be fulfilled for a minimum?



Optimality Condition

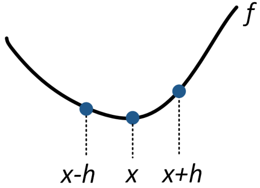
Conditions for a minimum x^*

1st order condition	$f'(x^*) = 0$	$\rightarrow$ *necessary*
2nd order condition	$f''(x^*) > 0$	$\rightarrow$ *sufficient*

Interpretation through Taylor series

$$f(x+h) = f(x) + f'(x)h + O\left(h^2\right)$$

Then



$$\left.\begin{array}{l} f(x+h) - f(x) \geq 0 \\ f(x-h) - f(x) \geq 0 \end{array}\right\} \Rightarrow f'(x) = 0$$

$$\left.\begin{array}{ll} f(x+h) - f(x) = \frac{1}{2} f''(x)h^2 + O\left(h^3\right) & > 0 \\ f(x-h) - f(x) = \frac{1}{2} f''(x)h^2 + O\left(h^3\right) & > 0 \end{array}\right\} \Rightarrow f''(x) > 0$$

Optimality Condition

Theorem (sufficient optimality condition)
Let f be twice continuously differentiable and let $\boldsymbol{x}^* \in \mathbb{R}^n$, if

1. First order condition

$$\nabla f(\boldsymbol{x}^*) = [0, \ldots, 0]^T$$

2. Second order condition

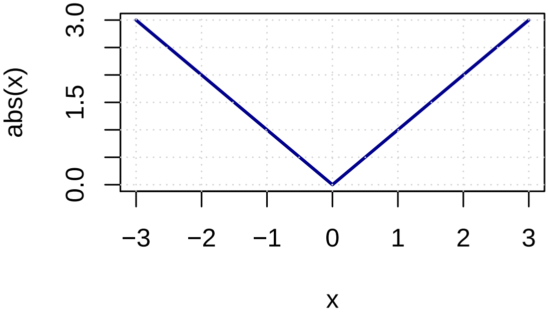
$$\nabla^2 f(\boldsymbol{x}^*) \text{ is positive definite}$$

then $\boldsymbol{x}^*$ is a strict local minimizer

Optimality Conditions

The previous theory does not cover all cases

- Imagine $f(x) = |x|$



- $f(x)$ has a global minimum at $x^* = 0$
- Since f is not differentiable, the optimality conditions do not apply

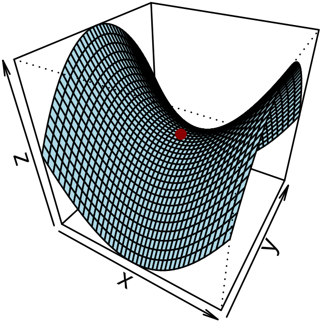
Stationarity

Definition

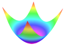
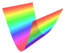
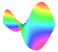
- Let f be continuously differentiable. A point $\boldsymbol{x}^* \in \mathbb{R}^n$ is stationary if
$$\nabla f(\boldsymbol{x}^*) = 0$$
- $\boldsymbol{x}^*$ is called a saddle point if it is neither a local minimum or maximum

Examples

- $f(x) = -x^2$ has only one stationary $x^* = 0$, since $\nabla f(x^*) = -2x^* = 0$
- $f(x) = x^3$ has a saddle point at $x^* = 0$
- $f(x_1, x_2) = x_1^2 - x_2^2$ has a saddle point $\boldsymbol{x}^* = [0, 0]^T$



Stationary Points

Nature of $\boldsymbol{x}^*$	Definiteness of H	$\boldsymbol{x}^T H \boldsymbol{x}$	All λ_i	Illustration
Minimum	positive definite	> 0	> 0	
Valley	positive semi-definite	≥ 0	≥ 0	
Saddle point	indefinite	$\neq 0$	$\neq 0$	
Ridge	negative semi-definite	≤ 0	≤ 0	
Maximum	negative definite	< 0	< 0	

Convexity

Definitions

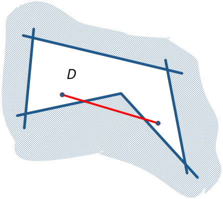
- A domain $D \subseteq \mathbb{R}^n$ is convex if

$$\forall x_1, x_2 \in D \, \forall \alpha \in [0,1] \quad \alpha x_1 + (1-\alpha) x_2 \in D$$

- A function $f : D \to \mathbb{R}$ is convex if

$$\forall x_1, x_2 \in D \, \forall \alpha \in [0,1] \quad f(\alpha x_1 + (1-\alpha) x_2) \leq \alpha x_1 + (1-\alpha) x_2$$

convex



non-convex / concave

Global Optimum

- Convexity gives information about the curvature, thus stationary points
- Constraints of an optimization define the feasible set

$$D = \{\boldsymbol{x} \in D \,\|\, g(\boldsymbol{x}) \leq 0, h(\boldsymbol{x}) = 0\}$$

  which can be either convex or concave
- Global minima are usually difficult to find numerically, except for cases of convex optimization

Definition
An optimization problem is convex if both the objective function f and its feasible set are convex

Theorem
The solution of a convex optimization is also its global solution

Outline

1. Number Representations
2. Linear Algebra
3. Differentiation
4. Taylor Approximation
5. Optimality Conditions
6. Wrap-Up

Summary: Linear Algebra

Dot product	$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^T \boldsymbol{b}$
Norm	$\\|\boldsymbol{x}\\|$
Transpose	$A^T = [a_{ji}]_{ij}$
Identity matrix	$I_n = \mathrm{diag}(1, \ldots, 1) \in \mathbb{R}^{n \times n}$
Inverse	$A^{-1} \in R^{n \times n}$ such that $AA^{-1} = I$
Pseudoinverse	$A^{+} \in R^{m \times n}$ such that $AA^{+} = I$
Determinant	$\det A$
Eigenvalue, -vector	$A\boldsymbol{v} = \lambda \boldsymbol{v}$ for $\boldsymbol{v} \neq 0$
Positive definite	$\boldsymbol{x}^T A \boldsymbol{x} > 0$ for $\boldsymbol{x} \neq 0$

Summary: Numerical Analysis

Partial derivative	$\frac{\mathrm{d}f}{\mathrm{d}x_i}(\boldsymbol{x})$
Finite differences	Numerical approximations to derivatives
Gradient	$\nabla f(\boldsymbol{x})$
Hessian	$H(\boldsymbol{x}) = \nabla^2 f(\boldsymbol{x})$
Taylor series	$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$

Summary: Optimality Conditions

- Local minimum x^* if $f(x^*) \leq f(x)$ for all $x \in N(x^*) \subseteq D$
- Global minimum if $f(x^*) \leq f(x)$ for all $x \in D$
- Sufficient conditions for a strict local optimizer
  1. $\nabla f(\boldsymbol{x}^*) = 0$ (stationarity)
  2. $\nabla^2 f(\boldsymbol{x}^*)$ is positive definite
- Convex optimization has a convex objective and a convex feasible set
- The minimum in convex optimization is always a global minimum

Summary: R Commands

`digitsBase(...)`	Convert number from base 10 to another base
`strtoi(...)`	Convert a number from any base to base 10
%*%	Dot product, matrix multiplication
`drop(A)`	Deletes dimensions in A with only one value
`t(A)`	Transpose a matrix A
`diag(n)`	Identity matrix of size $n \times n$
`solve(A),ginv(A)`	Inverse or pseudoinverse of a matrix A
`det(A)`	Determinant of A if existent
`eigen(A)`	Eigenvalues and eigenvectors of a matrix
`is.positive.definite(A),...`	Tests if matrix A is positive definite, ...
`D(f, x)`	Derivative of a function f regarding x
`eval(f, ...)`	Evaluates an expression f at a specific point
`optimHess(...)`	Approximate to Hessian matrix
`taylor(...),polyval(...)`	Taylor approximation

Outlook

Additional Materials

Further exercises in homework sheets 3 and 4

Future Exercises

R will be used to implement optimization algorithms