1. Vectors

- notation
- examples
- vector operations
- linear functions
- complex vectors
- complexity of vector computations

Vector

- a vector is an ordered finite list of numbers
- we use two styles of notation:

$$\begin{bmatrix} -1.1 \\ 0.0 \\ 3.6 \\ 7.2 \end{bmatrix} = (-1.1,\ 0.0,\ 3.6,\ 7.2)$$

- numbers in the list are the *elements* (*entries*, *coefficients*, *components*)
- number of elements is the *size* (*length*, *dimension*) of the vector
- a vector of size n is called an n-vector
- set of n-vectors with real elements is denoted $\mathbf{R}^n$

Conventions

- we usually denote vectors by lowercase letters

$$a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = (a_1, a_2, \ldots, a_n)$$

- ith element of vector a is denoted a_i
- i is the *index* of the ith element a_i

Note

- several other conventions exist
- we'll make exceptions, *e.g.*, a_i can refer to ith vector in a collection of vectors

Block vectors, subvectors

Stacking

- vectors can be stacked (concatenated) to create larger vectors
- example: stacking vectors b, c, d of size m, n, p gives an $(m+n+p)$-vector

$$a = \begin{bmatrix} b \\ c \\ d \end{bmatrix} = \left(b_1, \ldots, b_m, c_1, \ldots, c_n, d_1, \ldots, d_p\right)$$

- other notation: $a = (b, c, d)$

Subvectors

- colon notation can be used to define subvectors (slices) of a vector
- example: if $a = (1, -1, 2, 0, 3)$, then $a_{2:4} = (-1, 2, 0)$

Special vectors

Zero vector and ones vector

$$0 = (0, 0, \ldots, 0), \qquad \mathbf{1} = (1, 1, \ldots, 1)$$

size follows from context (if not, we add a subscript and write 0_n, $\mathbf{1}_n$)

Unit vectors

- there are n unit vectors of size n, written $e_1, e_2, \ldots, e_n$
- ith unit vector is zero except its ith element which is 1; for $n = 3$,

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

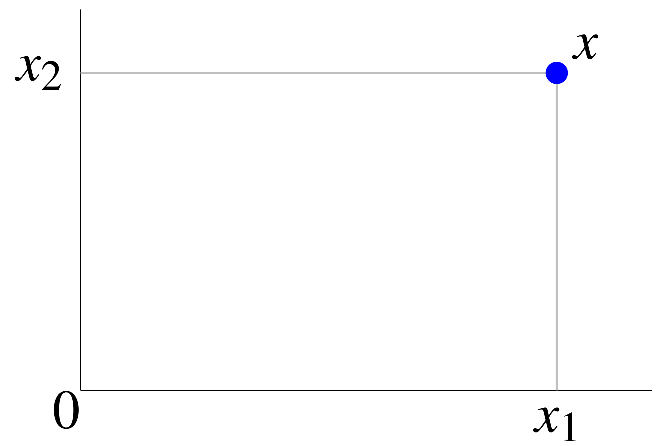
- size of e_i follows from context (or should be specified explicitly)

Outline

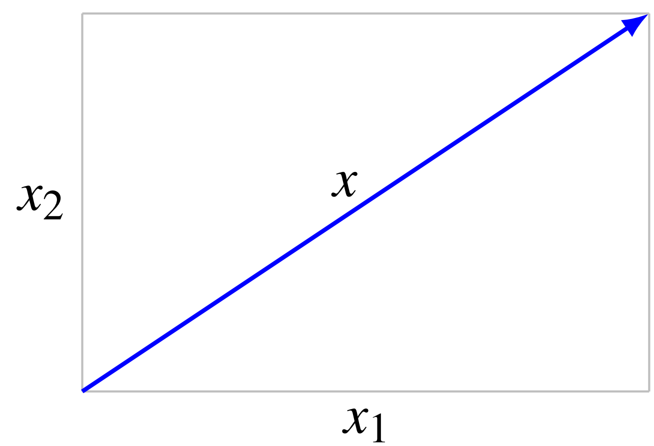
- notation
- **examples**
- vector operations
- linear functions
- complex vectors
- complexity of vector computations

Location and displacement

Location: coordinates of a point in a plane or three-dimensional space



Displacement: shown as arrow in plane or 3-D space

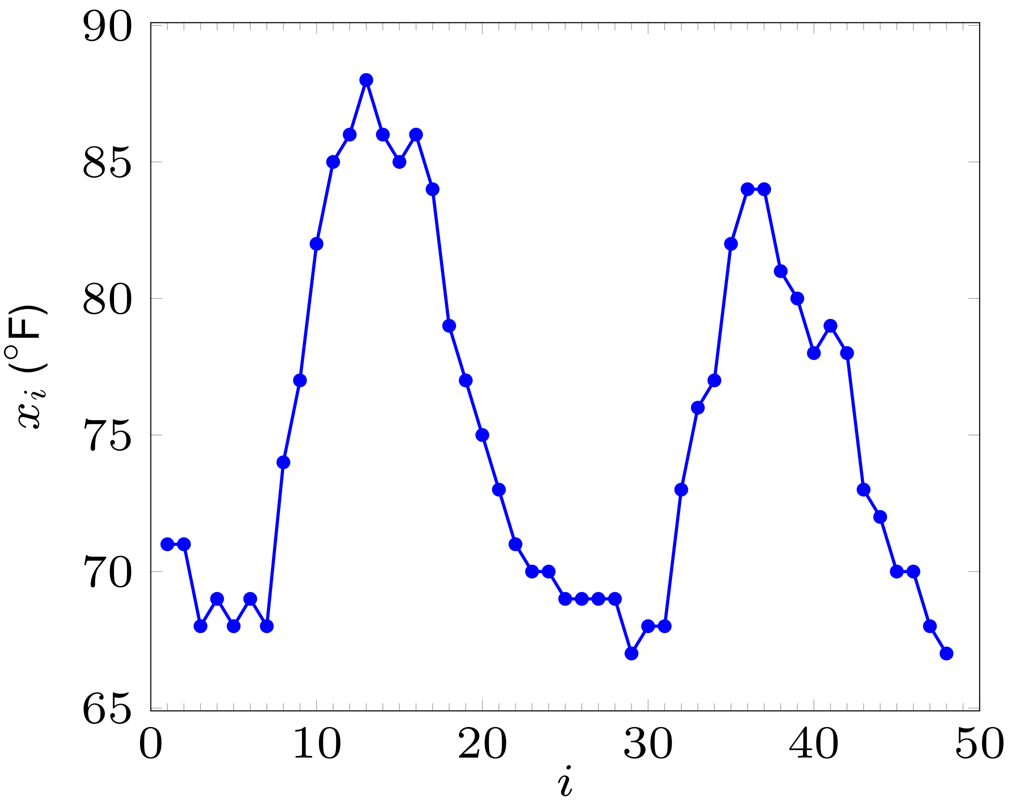


other quantities that have direction and magnitude, *e.g.*, force vector

Signal or time series

elements of n-vector are values of some quantity at n different times

- hourly temperature over period of n hours

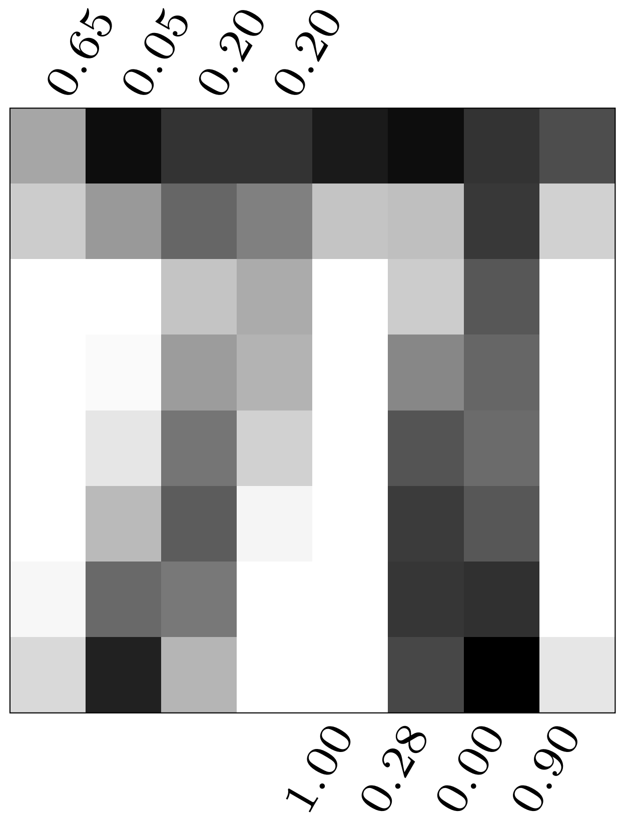


- daily return of a stock for period of n trading days
- cash flow: payments to an entity over n periods (*e.g.*, quarters)

Images, video

Monochrome (black and white) image

grayscale values of $M \times N$ pixels stored as MN-vector (*e.g.*, row-wise)



$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{62} \\ x_{63} \\ x_{64} \end{bmatrix} = \begin{bmatrix} 0.65 \\ 0.05 \\ 0.20 \\ \vdots \\ 0.28 \\ 0.00 \\ 0.90 \end{bmatrix}$$

Color image: $3MN$-vectors with R, G, B values of the MN pixels

Video: vector of size KMN represents K monochrome images of $M \times N$ pixels

Portfolio vector, resource vector

Portfolio

- n-vector represents stock portfolio or investment in n assets
- ith element is amount invested in asset i
- elements can be no. of shares, dollar values, or fractions of total dollar amount

Resource vector

- elements of n-vector represent quantities of n resources or commodities
- sign indicates whether quantity is held or owed, produced or consumed, ...
- example: bill of materials gives quantities needed to create a product

Word count vectors

- vector represents a document
- size of vector is number of words in a dictionary
- word count vector: element i is number of times word i occurs in the document
- word histogram: element i is frequency of word i in the document

Example

> **Word** count vectors are used **in** computer based **document** analysis. Each entry of the **word** count vector is the **number** of times the associated dictionary **word** appears **in** the **document**.

$$
\begin{array}{ll}
\text{word} & \\
\text{in} & \\
\text{number} & \\
\text{horse} & \\
\text{document} &
\end{array}
\qquad
\left[\begin{array}{c} 3 \\ 2 \\ 1 \\ 0 \\ 2 \end{array}\right]
$$

Feature vectors

contain values of variables or attributes that describe members of a set

Examples

- age, weight, blood pressure, gender, . . . , of patients
- square footage, #bedrooms, list price, . . . , of houses in an inventory

Note

- vector elements can represent very different quantities, in different units
- can contain categorical features (*e.g.*, $0/1$ for male/female)
- ordering has no particular meaning

Polynomials and generalized polynomials

a polynomial of degree $n-1$ or less

$$f(t) = c_1 + c_2 t + c_3 t^2 + \cdots + c_n t^{n-1}$$

can be represented by an n-vector $(c_1, c_2, \ldots, c_n)$

Extensions

- n basis functions $f_1(t), \ldots, f_n(t)$
- n-vector c represents the function $f(t) = c_1 f_1(t) + \cdots + c_n f_n(t)$
- example: the cosine polynomial

$$f(t) = c_1 + c_2 \cos t + c_3 \cos(2t) + \cdots + c_n \cos((n-1)t)$$

can be represented by an n-vector $(c_1, c_2, \ldots, c_n)$

Outline

- notation
- examples
- **vector operations**
- linear functions
- complex vectors
- complexity of vector computations

Addition and subtraction

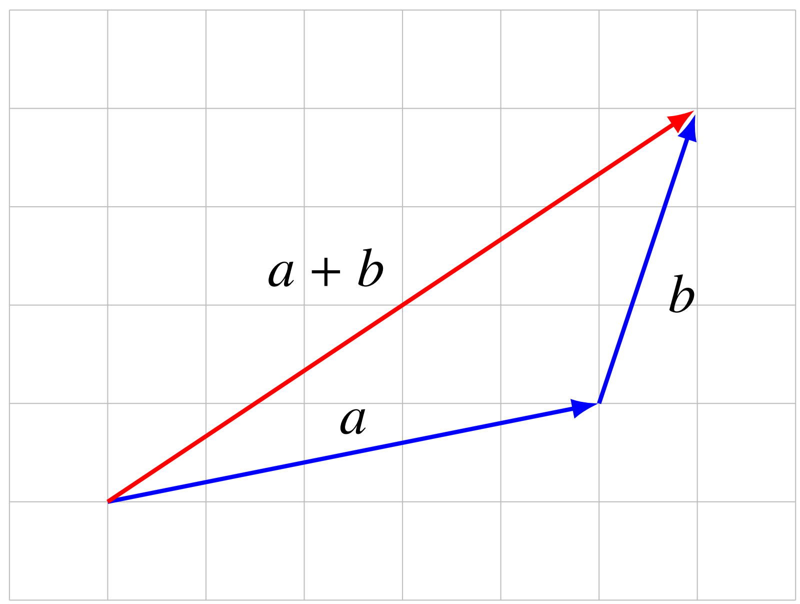
$$a + b = \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_n + b_n \end{bmatrix}, \qquad a - b = \begin{bmatrix} a_1 - b_1 \\ a_2 - b_2 \\ \vdots \\ a_n - b_n \end{bmatrix}$$

- commutative

$$a + b = b + a$$

- associative

$$a + (b + c) = (a + b) + c$$



Scalar-vector and componentwise multiplication

Scalar-vector multiplication: for scalar β and n-vector a,

$$\beta \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \beta a_1 \\ \beta a_2 \\ \vdots \\ \beta a_n \end{bmatrix}$$

Component-wise multiplication: for n-vectors a, b

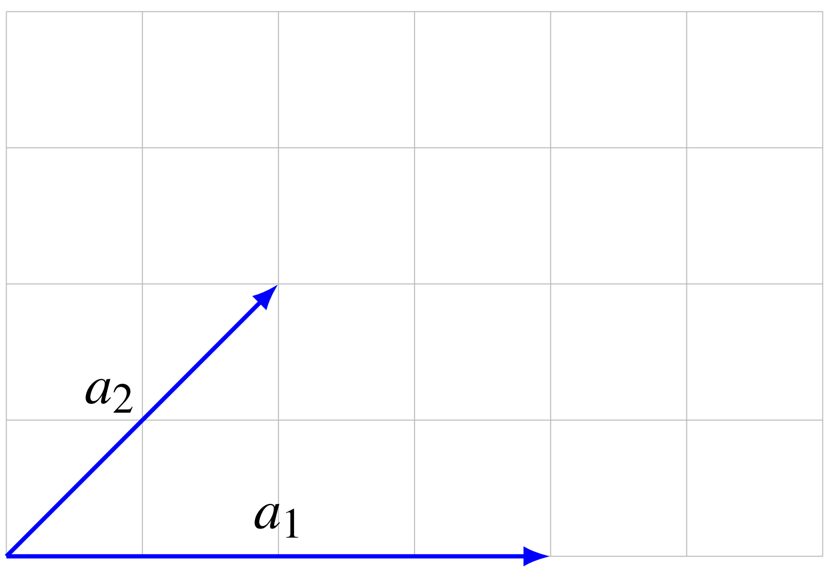
$$a \circ b = \begin{bmatrix} a_1 b_1 \\ a_2 b_2 \\ \vdots \\ a_n b_n \end{bmatrix}$$

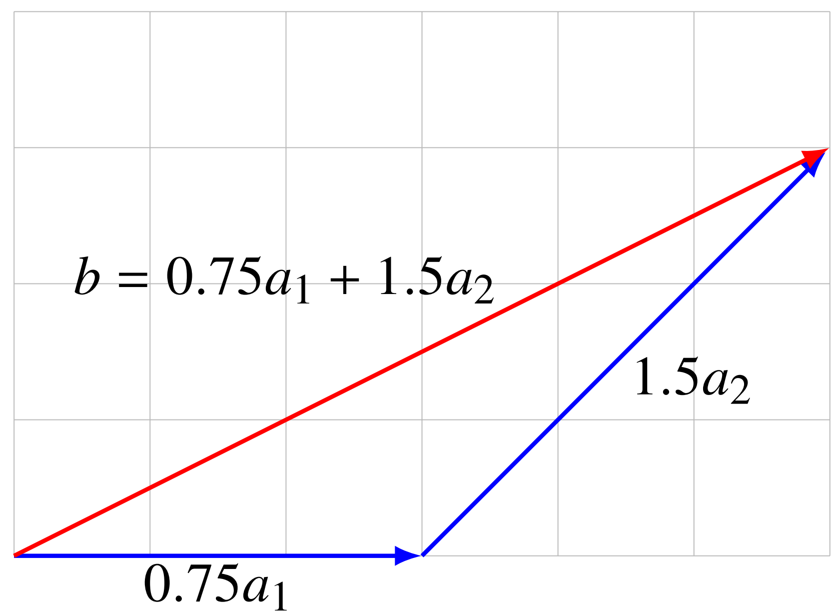
Linear combination

a *linear combination* of vectors $a_1, \ldots, a_m$ is a sum of scalar-vector products

$$\beta_1 a_1 + \beta_2 a_2 + \cdots + \beta_m a_m$$

the scalars $\beta_1, \ldots, \beta_m$ are the *coefficients* of the linear combination





Inner product

the inner product of two n-vectors a, b is defined as

$$a^T b = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

- a scalar
- meaning of superscript ^{T} will be explained when we discuss matrices
- other notation: $\langle a, b \rangle$, $(a \mid b)$, $\ldots$

Properties

for vectors a, b, c of equal length, scalar γ

- $a^T a = a_1^2 + a_2^2 + \cdots + a_n^2 \geq 0$
- $a^T a = 0$ only if $a = 0$
- commutative:
$$a^T b = b^T a$$
- associative with scalar multiplication:
$$(\gamma a)^T b = \gamma(a^T b)$$
- distributive with vector addition:
$$(a + b)^T c = a^T c + b^T c$$

Simple examples

Inner product with unit vector

$$e_i^T a = a_i$$

Differencing

$$(e_i - e_j)^T a = a_i - a_j$$

Sum and average

$$\mathbf{1}^T a = a_1 + a_2 + \cdots + a_n$$

$$(\frac{1}{n}\mathbf{1})^T a = \frac{a_1 + a_2 + \cdots + a_n}{n}$$

Examples

Weighted sum

- f is vector of features
- w is vector of weights
- inner product $w^T f = w_1 f_1 + w_2 f_2 + \cdots + w_n f_n$ is total score

Cost

- p is vector of prices of n goods
- q is vector of quantities purchased
- inner product $p^T q = p_1 q_1 + p_2 q_2 + \cdots + p_n q_n$ is total cost

Examples

Discounted total

- c is a cash flow over n periods

- d is vector of discount factors assuming interest rate $r \geq 0$:

$$d = (1, \ \frac{1}{1+r}, \ \frac{1}{(1+r)^2}, \ \ldots, \ \frac{1}{(1+r)^{n-1}})$$

- inner product $d^T c$ is discounted total or *net present value* of cash flow

$$d^T c = c_1 + \frac{c_2}{1+r} + \frac{c_3}{(1+r)^2} + \cdots + \frac{c_n}{(1+r)^{n-1}}$$

Examples

Portfolio return

- h is portfolio vector, with h_i the dollar value of asset i held
- r is vector of fractional returns over the investment period:

$$r_i = \frac{p_i^{\text{final}} - p_i^{\text{init}}}{p_i^{\text{init}}}, \quad i = 1, \ldots, n$$

  p_i^{init} and p_i^{final} are the prices of asset i at the beginning and end of the period

- $r^T h = r_1 h_1 + \cdots + r_n h_n$ is the total return, in dollars, over the period

Outline

- notation
- examples
- vector operations
- **linear functions**
- complex vectors
- complexity of vector computations

Linear function

a function $f : \mathbf{R}^n \to \mathbf{R}$ is **linear** if the superposition property

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y) \tag{1}$$

holds for all n-vectors x, y and all scalars α, β

Extension: if f is linear, superposition holds for any linear combination:

$$f(\alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_m u_m) = \alpha_1 f(u_1) + \alpha_2 f(u_2) + \cdots + \alpha_m f(u_m)$$

for all scalars $\alpha_1, \ldots, \alpha_m$ and all n-vectors $u_1, \ldots, u_m$

(this follows by applying (1) repeatedly)

Inner product function

for fixed $a \in \mathbf{R}^n$, define a function $f : \mathbf{R}^n \to \mathbf{R}$ as

$$f(x) = a^T x = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

- any function of this type is linear:

$$a^T(\alpha x + \beta y) = \alpha(a^T x) + \beta(a^T y)$$

  holds for all scalars α, β and all n-vectors x, y

- every linear function can be written as an inner-product function:

$$\begin{aligned} f(x) &= f(x_1 e_1 + x_2 e_2 + \cdots + x_n e_n) \\ &= x_1 f(e_1) + x_2 f(e_2) + \cdots + x_n f(e_n) \end{aligned}$$

  line 2 follows from superposition

Examples in $\mathbf{R}^3$

- $f(x) = \frac{1}{3}(x_1 + x_2 + x_3)$ is linear: $f(x) = a^T x$ with $a = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$

- $f(x) = -x_1$ is linear: $f(x) = a^T x$ with $a = (-1, 0, 0)$

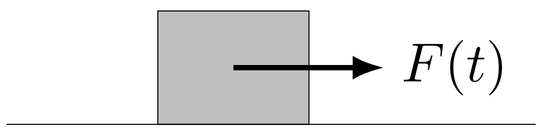
- $f(x) = \max\{x_1, x_2, x_3\}$ is not linear: superposition does not hold for

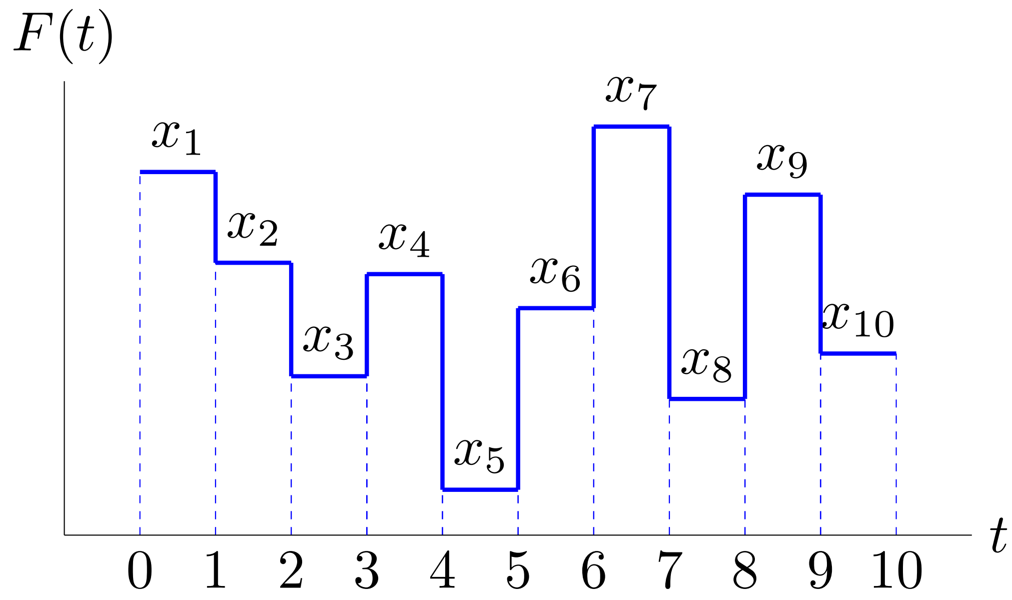
$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad y = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \alpha = -1, \qquad \beta = 1$$

  we have $f(x) = 1$, $f(y) = 0$,

$$f(\alpha x + \beta y) = 0 \neq \alpha f(x) + \beta f(y) = -1$$

Exercise





- unit mass with zero initial position and velocity
- apply piecewise-constant force $F(t)$ during interval $[0, 10)$:

$$F(t) = x_j \quad \text{for } t \in [j-1, j), \quad j = 1, \ldots, 10$$

- define $f(x)$ as position at $t = 10$, $g(x)$ as velocity at $t = 10$

are f and g linear functions of x?

Solution

- from Newton's law $s''(t) = F(t)$ where $s(t)$ is the position at time t
- integrate twice to get final velocity and position

$$\begin{aligned} s'(10) &= \int_0^{10} F(t)\,dt \\ &= x_1 + x_2 + \cdots + x_{10} \\ s(10) &= \int_0^{10} s'(t)\,dt \\ &= \frac{19}{2}x_1 + \frac{17}{2}x_2 + \frac{15}{2}x_3 + \cdots + \frac{1}{2}x_{10} \end{aligned}$$

the two functions are linear: $f(x) = a^T x$ and $g(x) = b^T x$ with

$$a = (\frac{19}{2}, \frac{17}{2}, \ldots, \frac{3}{2}, \frac{1}{2}), \qquad b = (1, 1, \ldots, 1)$$

Affine function

a function $f : \mathbf{R}^n \to \mathbf{R}$ is **affine** if it satisfies

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

for all n-vectors x, y and all scalars α, β with $\alpha + \beta = 1$

Extension: if f is affine, then

$$f(\alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_m u_m) = \alpha_1 f(u_1) + \alpha_2 f(u_2) + \cdots + \alpha_m f(u_m)$$

for all n-vectors $u_1, \ldots, u_m$ and all scalars $\alpha_1, \ldots, \alpha_m$ with

$$\alpha_1 + \alpha_2 + \cdots + \alpha_m = 1$$

Affine functions and inner products

for fixed $a \in \mathbf{R}^n$, $b \in \mathbf{R}$, define a function $f : \mathbf{R}^n \longrightarrow \mathbf{R}$ by

$$f(x) = a^T x + b = a_1x_1 + a_2x_2 + \cdots + a_nx_n + b$$

i.e., an inner-product function plus a constant (offset)

- any function of this type is affine: if $\alpha + \beta = 1$ then

$$a^T(\alpha x + \beta y) + b = \alpha(a^T x + b) + \beta(a^T y + b)$$

- every affine function can be written as $f(x) = a^T x + b$ with:

$$\begin{aligned} a &= (f(e_1) - f(0),\ f(e_2) - f(0),\ \ldots,\ f(e_n) - f(0)) \\ b &= f(0) \end{aligned}$$

Affine approximation

first-order Taylor approximation of differentiable $f : \mathbf{R}^n \to \mathbf{R}$ around z:

$$\hat{f}(x) = f(z) + \frac{\partial f}{\partial x_1}(z)(x_1 - z_1) + \cdots + \frac{\partial f}{\partial x_n}(z)(x_n - z_n)$$

- generalizes first-order Taylor approximation of function of one variable

$$\hat{f}(x) = f(z) + f'(z)(x - z)$$

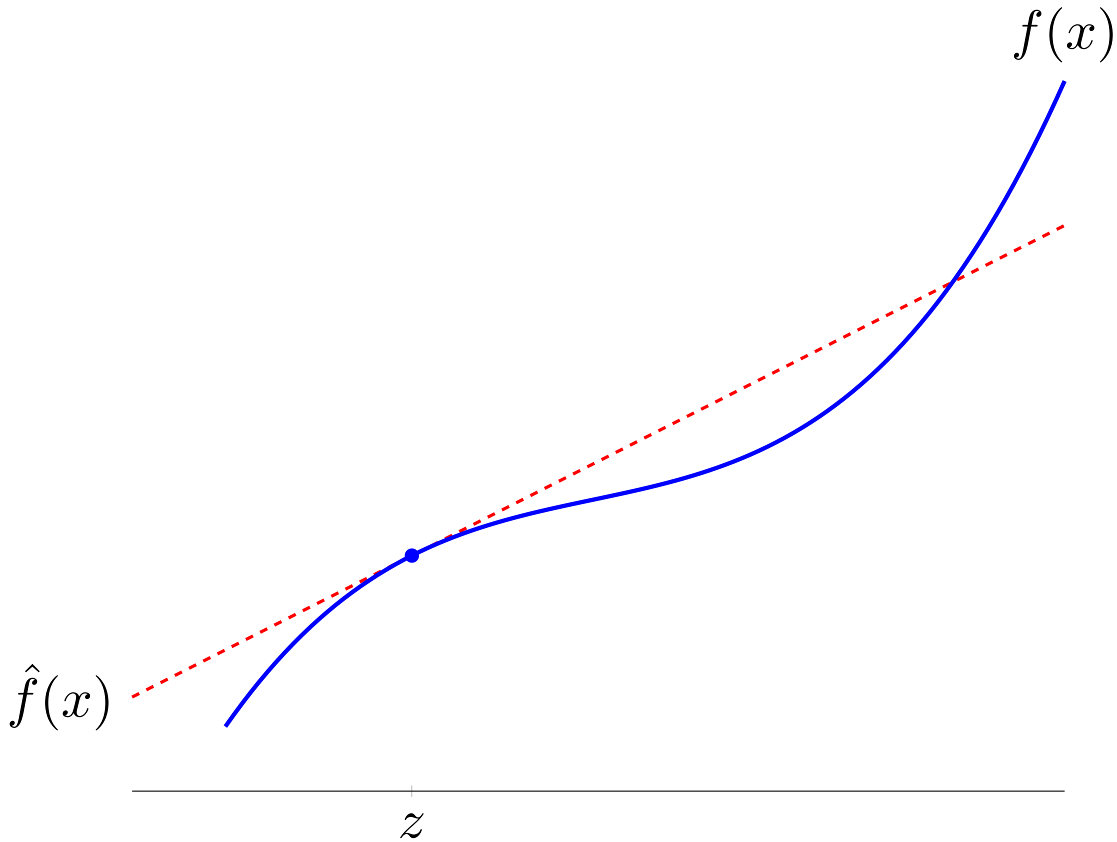
- $\hat{f}$ is a local affine approximation of f around z

- in vector notation: $\hat{f}(x) = f(z) + \nabla f(z)^T(x - z)$ where

$$\nabla f(z) = \left(\frac{\partial f}{\partial x_1}(z),\ \frac{\partial f}{\partial x_2}(z),\ \ldots,\ \frac{\partial f}{\partial x_n}(z)\right)$$

  the n-vector $\nabla f(z)$ is called the *gradient* of f at z

Example with one variable



$$\hat{f}(x) = f(z) + f'(z)(x - z)$$

Example with two variables

$$f(x_1, x_2) = x_1 - 3x_2 + e^{2x_1+x_2-1}$$

Gradient

$$\nabla f(x) = \begin{bmatrix} 1 + 2e^{2x_1+x_2-1} \\ -3 + e^{2x_1+x_2-1} \end{bmatrix}$$

First-order Taylor approximation around $z = 0$

$$\begin{aligned} \hat{f}(x) &= f(0) + \nabla f(0)^T(x - 0) \\ &= e^{-1} + (1 + 2e^{-1})x_1 + (-3 + e^{-1})x_2 \end{aligned}$$

Regression model

$$\hat{y} = x^T \beta + v = \beta_1 x_1 + \cdots + \beta_p x_p + v$$

- x is feature vector
- elements x_i are *regressors*, *independent variables*, or *inputs*
- $\beta = (\beta_1, \ldots, \beta_p)$ is vector of *weights* or *coefficients*
- v is *offset* or *intercept*
- coefficients $\beta_1, \ldots, \beta_p, v$ are the *parameters* of the regression model
- $\hat{y}$ is *prediction* (or *outcome*, *dependent variable*)
- regression model expresses $\hat{y}$ as an affine function of x

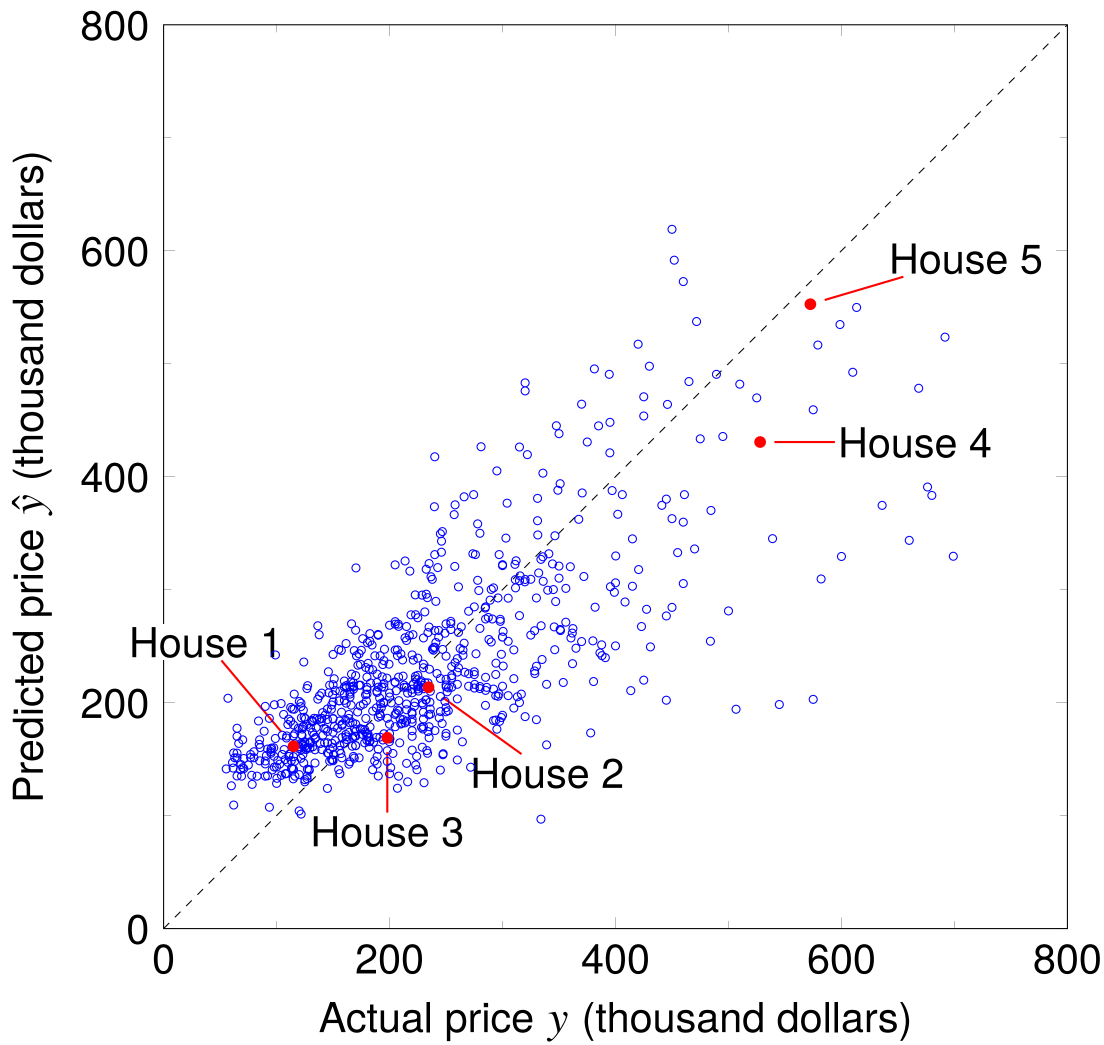
Example: house price regression model

$$\hat{y} = 54.4 + 148.73x_1 - 18.85x_2$$

- $\hat{y}$ is predicted selling price in thousands of dollars
- x_1 is area (1000 square feet); x_2 is number of bedrooms

House	x_1 (area)	x_2 (beds)	y (price)	$\hat{y}$ (prediction)
1	0.846	1	115.00	161.37
2	1.324	2	234.50	213.61
3	1.150	3	198.00	168.88
4	3.037	4	528.00	430.67
5	3.984	5	572.50	552.66

Example



scatter plot shows sale prices for 774 houses in Sacramento

Outline

- notation
- examples
- vector operations
- linear functions
- **complex vectors**
- complexity of vector computations

Complex numbers

Complex number: $x = \alpha + \mathrm{j}\beta$ with α, β real scalars

- $\mathrm{j} = \sqrt{-1}$ (more common notation is i or j)
- α is the *real part* of x, denoted $\operatorname{Re} x$
- β is the *imaginary* part, denoted $\operatorname{Im} x$

set of complex numbers is denoted $\mathbf{C}$

Modulus and conjugate

- modulus (absolute value, magnitude): $|x| = \sqrt{(\operatorname{Re} x)^2 + (\operatorname{Im} x)^2}$
- conjugate: $\bar{x} = \operatorname{Re} x - \mathrm{j} \operatorname{Im} x$
- useful formulas:

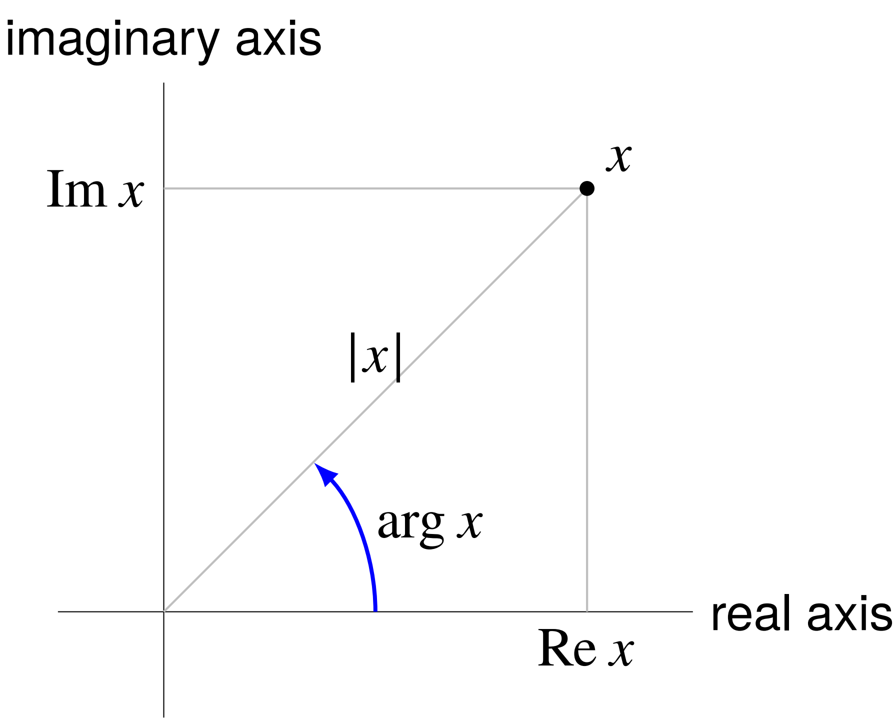
$$\operatorname{Re} x = \frac{x + \bar{x}}{2}, \qquad \operatorname{Im} x = \frac{x - \bar{x}}{2\mathrm{j}}, \qquad |x|^2 = \bar{x}x$$

Polar representation

nonzero complex number $x = \operatorname{Re} x + \mathrm{j} \operatorname{Im} x$ can be written as

$$x = |x| (\cos\theta + \mathrm{j} \sin\theta) = |x| e^{\mathrm{j}\theta}$$

- $\theta \in [0, 2\pi)$ is the *argument* (*phase angle*) of x (notation: $\arg x$)
- $e^{\mathrm{j}\theta}$ is complex exponential: $e^{\mathrm{j}\theta} = \cos\theta + \mathrm{j} \sin\theta$



Complex vector

- vector with complex elements: $a = \alpha + \mathrm{j}\beta$ with α, β real vectors
- real and imaginary part, conjugate are defined componentwise:

$$
\begin{aligned}
\operatorname{Re} a &= (\operatorname{Re} a_1, \operatorname{Re} a_2, \ldots, \operatorname{Re} a_n) \\
\operatorname{Im} a &= (\operatorname{Im} a_1, \operatorname{Im} a_2, \ldots, \operatorname{Im} a_n) \\
\bar{a} &= \operatorname{Re} a - \mathrm{j} \operatorname{Im} a
\end{aligned}
$$

- set of complex n-vectors is denoted $\mathbf{C}^n$
- addition, scalar/componentwise multiplication defined as in $\mathbf{R}^n$:

$$
a + b = \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_n + b_n \end{bmatrix}, \qquad \gamma a = \begin{bmatrix} \gamma a_1 \\ \gamma a_2 \\ \vdots \\ \gamma a_n \end{bmatrix}, \qquad a \circ b = \begin{bmatrix} a_1 b_1 \\ a_2 b_2 \\ \vdots \\ a_n b_n \end{bmatrix}
$$

Complex inner product

the inner product of complex n-vectors a, b is defined as

$$b^H a = \bar{b}_1 a_1 + \bar{b}_2 a_2 + \cdots + \bar{b}_n a_n$$

- a complex scalar
- meaning of superscript ^{H} will be explained when we discuss matrices
- other notation: $\langle a, b \rangle$, $(a \mid b)$, ...
- for real vectors, reduces to real inner product $b^T a$

Properties

for complex n-vectors a, b, c and complex scalars γ

- $a^H a \geq 0$: follows from

$$\begin{aligned} a^H a &= \bar{a}_1 a_1 + \bar{a}_2 a_2 + \cdots + \bar{a}_n a_n \\ &= |a_1|^2 + |a_2|^2 + \cdots + |a_n|^2 \end{aligned}$$

- $a^H a = 0$ only if $a = 0$
- $b^H a = \overline{a^H b}$
- $b^H(\gamma a) = \gamma(b^H a)$
- $(\gamma b)^H a = \bar{\gamma}(b^H a)$
- $(b + c)^H a = b^H a + c^H a$
- $b^H(a + c) = b^H a + b^H c$

Outline

- notation
- examples
- vector operations
- linear functions
- complex vectors
- **complexity of vector computations**

Floating point operation

Floating point operation (flop)

- the unit of complexity when comparing vector and matrix algorithms
- 1 flop = one basic arithmetic operation ($+$, $-$, $*$, $/$, $\sqrt{}$, $\ldots$) in $\mathbf{R}$ or $\mathbf{C}$

Comments: this is a very simplified model of complexity of algorithms

- we don't distinguish between the different types of arithmetic operations
- we don't distinguish between real and complex arithmetic
- we ignore integer operations (indexing, loop counters, $\ldots$)
- we ignore cost of memory access

Complexity

Operation count (flop count)

- total number of operations in an algorithm
- in linear algebra, typically a polynomial of the dimensions in the problem
- a crude predictor of run time of the algorithm:

$$\text{run time} \approx \frac{\text{number of operations (flops)}}{\text{computer speed (flops per second)}}$$

Dominant term: the highest-order term in the flop count

$$\frac{1}{3}n^3 + 100n^2 + 10n + 5 \approx \frac{1}{3}n^3$$

Order: the power in the dominant term

$$\frac{1}{3}n^3 + 10n^2 + 100 = \text{order } n^3$$

Examples

complexity of vector operations in this lecture (for vectors of size n)

- addition, subtraction: n flops
- scalar multiplication: n flops
- componentwise multiplication: n flops
- inner product: $2n - 1 \approx 2n$ flops

these operations are all order n