2. Vectors

- vector notation
- vector operations
- inner product and linear functions
- complexity
- examples of vectors

Vector

a *vector* is a collection of elements denoted as

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

- a_i is the ith *element* (*entry, coefficient, component*) of vector a
- i is the *index* of the ith element a_i
- number of elements n is the *size* (*length, dimension*) of the vector
- a vector of size n is called an n*-vector*
- example of a 4-vector:

$$a = \begin{bmatrix} -1.1 \\ 0.0 \\ 3.6 \\ 7.2 \end{bmatrix} = (-1.1, 0.0, 3.6, 7.2), \qquad a_3 = 3.6$$

Notes and conventions

- $\mathbb{R}^n$ is set of n-vectors with real entries
- $a \in \mathbb{R}^n$ means a is n-vector with real entries
- two n-vectors a and b are equal, denoted as $a = b$, if $a_i = b_i$ for all i
- a_i can refer to an ith vector in a collection of vectors
  - in this case, we use $(a_i)_j$ to denote the jth entry of vector a_i
  - example: if $a_2 = (-1, 2, -5)$, then $(a_2)_3 = -5$

Conventions

- parentheses are also used instead of rectangular brackets to represent a vector
- other notations exist to distinguish vectors from numbers (*e.g.*, $\boldsymbol{a}, \vec{a}, \mathbf{a}$)
- conventions vary; be prepared to distinguish scalars from vectors

Row vector and transpose

an *row* vector b of size n with entries $b_1, \ldots, b_n$ has the form:

$$b = \begin{bmatrix} b_1 & b_2 & \ldots & b_n \end{bmatrix}$$

- all vectors are column vectors unless otherwise stated
- other notation exists, *e.g.*, $b = [b_1, b_2, \ldots, b_n]$ (we will not use)

Transpose: the *transpose* of an n-column vector a is the row vector a^T:

$$a^T = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}^T = \begin{bmatrix} a_1 & a_2 & \ldots & a_n \end{bmatrix}$$

- $(\cdot)^T$ is transpose operation
- $(a^T)^T = a$ (transpose of row vector is a column vector)

Block vectors, subvectors

Stacking

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

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

- we say that b, c, and d are *subvectors* or *slices* of a
- example: if $a = 1$, $b = (2, -1)$, $c = (4, 2, 7)$, then $(a, b, c) = (1, 2, -1, 4, 2, 7)$

Subvectors slicing

- colon (:) notation is used to define subvectors (slices) of a vector
- for vector a, we define $a_{r:s} = (a_r, \ldots, a_s)$
- example: if $a = (1, -1, 2, 0, 3)$, then $a_{2:4} = (-1, 2, 0)$

Special vectors

Zero vector and ones vector

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

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, denoted by $e_1, e_2, \ldots, e_n$

$$(e_i)_j = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases}$$

- the ith unit vector is zero except its ith element which is 1
- example: for $n = 3$,

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

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

Sparsity

- a vector is *sparse* if many of its entries are 0
- can be stored and manipulated efficiently on a computer
- $\mathbf{nnz}(x)$ is number of entries that are nonzero
- examples:
  - $x = 0$ with $\mathbf{nnz}(x) = 0$
  - $x = e_i$ (unit vectors), $\mathbf{nnz}(x) = 1$
  - $x = (0, 0, 1, 0, 0, 0, -2, 0, 5, 0, 0)$, $\mathbf{nnz}(x) = 3$
- sparse vectors arise in many applications

Outline

- vector notation
- **vector operations**
- inner product and linear functions
- complexity
- examples of vectors

Addition and subtraction

for n-vectors a and b,

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

Example

$$\begin{bmatrix} 0 \\ 7 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 9 \\ 3 \end{bmatrix}$$

Properties: for vectors a, b of equal size

- commutative: $a + b = b + a$
- associative: $a + (b + c) = (a + b) + c$

Scalar-vector multiplication

for scalar β and n-vector a,

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

example:

$$(-2) \begin{bmatrix} 1 \\ 9 \\ 6 \end{bmatrix} = \begin{bmatrix} -2 \\ -18 \\ -12 \end{bmatrix}$$

Properties: for vectors a, b of equal size, scalars β, γ

- commutative: $\beta a = a\beta$
- associative: $(\beta\gamma)a = \beta(\gamma a)$, we write as $\beta\gamma a$
- distributive with scalar addition: $(\beta + \gamma)a = \beta a + \gamma a$
- distributive with vector addition: $\beta(a + b) = \beta a + \beta b$

Component-wise multiplication

for n-vectors a, b

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

Example

$$\begin{bmatrix} 1 \\ -3 \\ 0 \\ 8 \end{bmatrix} \circ \begin{bmatrix} -2 \\ 2 \\ 4 \\ 2 \end{bmatrix} = \begin{bmatrix} -2 \\ -6 \\ 0 \\ 16 \end{bmatrix}$$

Linear combination

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

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

- scalars $\beta_1, \ldots, \beta_m$ are the *coefficients* of the linear combination
- example: any n-vector b can be written as

$$b = b_1 e_1 + \cdots + b_n e_n$$

Special linear combinations

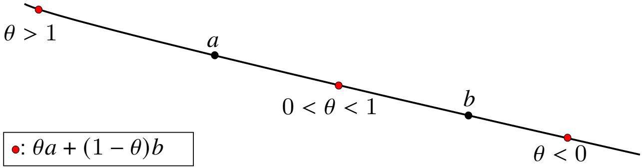
- *affine combination*: when $\beta_1 + \cdots + \beta_m = 1$
- *convex combination* or *weighted average*: when $\beta_1 + \cdots + \beta_m = 1$ and $\beta_i \geq 0$

Line segment

any point on the line passing through distinct a and b can be written as

$$c = \theta a + (1 - \theta)b$$

- θ is a scalar
- an affine combination
- for $0 \leq \theta \leq 1$, point c lie on the segment between a and b



Outline

- vector notation
- vector operations
- **inner product and linear functions**
- complexity
- examples of vectors

Inner product

the *inner product* (or *dot product*) of two n-vectors a, b is

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

- a scalar
- other notation exists: $\langle a, b \rangle$, $\langle a \mid b \rangle$, $a \cdot b$
- example:

$$\begin{bmatrix} -1 \\ 2 \\ 2 \end{bmatrix}^T \begin{bmatrix} 1 \\ 0 \\ -3 \end{bmatrix} = (-1)(1) + (2)(0) + (2)(-3) = -7$$

Properties of inner product

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

- nonnegativity: $a^T a \geq 0$, and $a^T a = 0$ if and 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: $(a + b)^T c = a^T c + b^T c$

Useful combination: for vectors a, b, c, d

$$(a + b)^T(c + d) = a^T c + a^T d + b^T c + b^T d$$

Block vectors: if vectors a, b are block vectors, and corresponding blocks $a_i, b_i \in \mathbb{R}^{n_i}$ have the same sizes (they conform),

$$a^T b = \begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix}^T \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix} = a_1^T b_1 + \cdots + a_k^T b_k$$

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

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

$$\text{avg}(x) = \frac{a_1 + a_2 + \cdots + a_n}{n} = \left(\frac{1}{n}\mathbf{1}\right)^T a$$

Linear functions

- $f : \mathbb{R}^n \to \mathbb{R}$ means f is a *function* mapping n-vectors to numbers
- example: $f(x) = x_1 + x_2 - x_4^2$ ($f : \mathbb{R}^4 \to \mathbb{R}$)

Linear functions: f is *linear* if it satisfies the superposition property

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

for all numbers α, β, and all n-vectors x, y

Extension: if f is linear, 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 $\alpha_1, \ldots, \alpha_m$

Inner product function

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

the inner product function is linear:

$$\begin{aligned} f(\alpha x + \beta y) &= a^T(\alpha x + \beta y) \\ &= a^T(\alpha x) + a^T(\beta y) \\ &= \alpha(a^T x) + \beta(a^T y) \\ &= \alpha f(x) + \beta f(y) \end{aligned}$$

All linear functions are inner products

- if $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is linear, then $f(x) = a^T x$ for some (unique) a
- this follows from

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

  with $a = (f(e_1), \ldots, f(e_n))$

Example

- mean or average value of an n-vector is linear

$$f(x) = \mathrm{avg}(x) = (x_1 + x_2 + \cdots + x_n)/n = a^T x$$

  where $a = (1/n, \ldots, 1/n) = (1/n)\mathbf{1}$ (sometimes denoted $\bar{x}$ or μ_x)

- maximum element func. $f(x) = \max\{x_1, \ldots, x_n\}$, is not linear (unless $n = 1$)
  - we can show this by a counterexample for $n = 2$
  - take $x = (1, -1), y = (-1, 1), \alpha = 1/2, \beta = 1/2$
  - then

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

Affine functions

$f : \mathbb{R}^n \to \mathbb{R}$ is *affine* if

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

for all n-vectors and scalars $\alpha + \beta = 1$

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

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

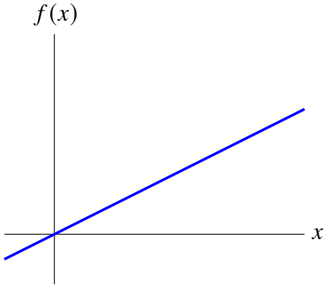
  $$\begin{aligned} a &= \big(f(e_1) - f(0), f(e_2) - f(0), \ldots, f(e_n) - f(0)\big) \\ b &= f(0) \end{aligned}$$

- an affine function is a linear function plus a constant

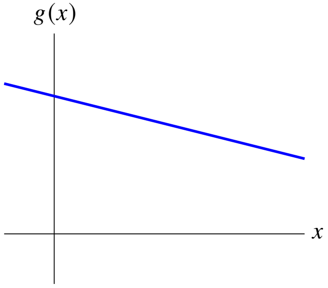
- often affine functions are called linear (which is mathematically not true)

Linear versus affine functions

f is linear



g is affine



Outline

- vector notation
- vector operations
- inner product and linear functions
- **complexity**
- examples of vectors

Floating point operation (FLOP)

Computer representation of numbers

- computers store (real) numbers in *floating-point format*
- number represented as 64 bits (0s and 1s), or 8 bytes (group of bits)
- each of 2^{64} sequences of bits corresponds to a specific number

Floating point operations

- 1 flop = one basic arithmetic operation ($+, -, *, /, \sqrt{}, \ldots$) in $\mathbb{R}$ (or complex $\mathbb{C}$)
- speed with which a computer can carry out flops is typically in 1-10 Gflop/s
- *complexity* of an operation is the number of flops required to carry it out
- flop is the unit of complexity when comparing algorithms; run time of the algorithm:

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

  this is a very crude and simplified model of complexity of algorithms

Dominant terms

- typically, complexity is highly simplified, dropping small or negligible terms
- 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

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

- order is useful in understanding how the time to execute the computation will scale when the size of the operands changes

Complexity of vector operations

for vectors of size n

- $x + y$ needs n additions, so n flops
- scalar multiplication: n flops
- componentwise multiplication: n flops
- inner product: $2n - 1 \approx 2n$ flops
  - we simplify this to $2n$ (or even n) flops
- these operations are all order n

Sparse vectors: when x and/or y is sparse

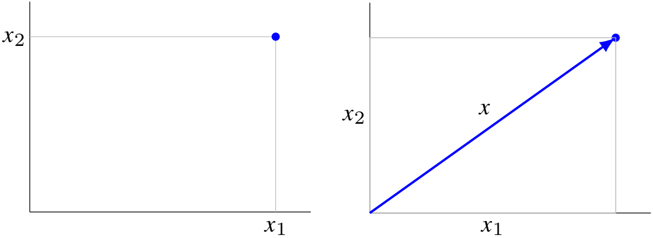
- ax requires $\textbf{nnz}(x)$ flops
- $x + y$ requires $\min\{\textbf{nnz}(x), \textbf{nnz}(y)\}$ flops
- if sparsity pattern do not overlap, $x + y$ requires zero flops
- $x^T y$ requires no more than $2\min\{\textbf{nnz}(x), \textbf{nnz}(y)\}$ flops

Outline

- vector notation
- vector operations
- inner product and linear functions
- complexity
- **examples of vectors**

Location and displacement

- location (position): coordinates of a point in 2-D (plane) or 3-D space
- displacement: vector represents the change in position from one point to another (shown as an arrow in plane or 3-D space)

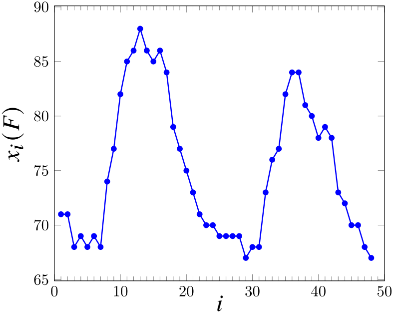


- other quantities that have direction and magnitude (velocity, force vector, ...)

Time series or signal

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

- hourly temperature over a period of n hours



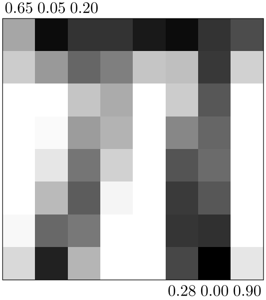
- audio signal: entries give the value of acoustic pressure at equally spaced times

Color, images, video

Color: 3-vector can represent a color, with RGB intensity values

Monochrome (black and white) image

grayscale values of $M \times N$ pixels stored as MN-vector (row-wise or column-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

Quantities, values, proportions

Quantities

- elements of n-vector represent quantities of n resources or products
- sign indicates whether quantity is held or owed, produced or consumed, ...
- example: *bill of materials* is the list of resources (items) that are required to build a product represented as a vector that gives the amounts of n resources required to create a product

Values across a population

- n-vector gives values of some quantity across population of individuals or entities
- example: an n-vector b can give the blood pressure of a collection of n patients, with b_i the blood pressure of patient i

Proportions

- vector w give fractions or proportions out of n choices, outcomes, or options
- w_i the fraction with choice or outcome i ($w_i \geq 0$ and $w_1 + \cdots + w_n = 1$)

Portfolio

Portfolio

- a collection of financial assets (investments) such as stocks, bonds, cash, commodities (*e.g.*, gold), real estate ...
- it refers to a group of investments that an investor uses in order to earn a profit while making sure that capital or assets are preserved

Vector representation

- n-vector s can represent stock portfolio (*e.g.*, investment in n assets)
- s_i is the number of shares of asset i held (or invested in asset i)
- elements can be the no. of shares, dollar values, fractions of total dollar amount
- shares you owe another party (short positions) are represented by negative values

Daily return and cash flow

Daily return

- daily fractional return of a stock for a period of n trading days
- example: return time series vector $(-0.022, +0.014, +0.004)$ means stock price
  - went down 2.2% on the first day
  - then up 1.4% the next day
  - and up again 0.4% on the third day

Cash flow

- cash flow: payments into and out of an entity over n periods
- example: vector $(1000, -10, -10, -10, -1010)$ represents
  - a one year loan of 1000
  - with 1% interest only payments made each period (*e.g.*, quarter)
  - and the principal and last interest payment at the end

Word count vectors

- vector represents a document
- size of vector is the number of words in a dictionary
- word *count vector:* entry i is the number of times word i occurs in document
- word *histogram:* entry i is frequency of word i in document (in percentage)

Example: *word count vectors are used in computer-based document analysis; each entry of the word count vector represents the number of times the associated dictionary word appears in the document*

$$\begin{array}{l} \text{word} \\ \text{in} \\ \text{number} \\ \text{horse} \\ \text{document} \end{array} \qquad \begin{bmatrix} 3 \\ 2 \\ 1 \\ 0 \\ 2 \end{bmatrix}$$

Features

Feature vector

- collects together n different quantities that relate to a single object
- entries are called the *features* or *attributes*

Examples

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

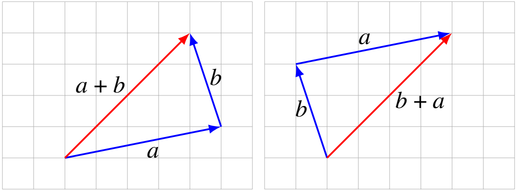
Notes

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

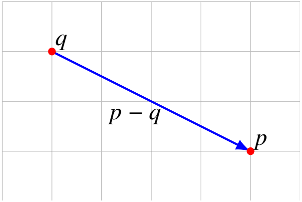
Addition and multiplication examples

Displacements addition

- if a and b are displacements, $a + b$ is the net displacement

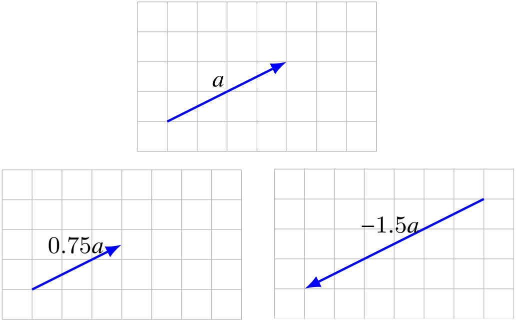


- displacement from point q to point p is $p - q$



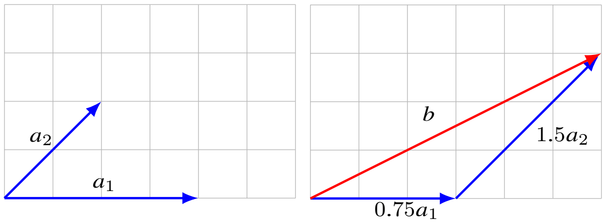
Displacement multiplication

- vector a represents a displacement
- for $\beta > 0$, βa is displacement in same direction of a, with magnitude scaled by β
- for $\beta < 0$, βa is displac. in the opposite direction of a, with mag. scaled by $|\beta|$



Linear combination of displacements

$$b = 0.75a_1 + 1.5a_2$$



Word count

- a and b are word count vectors (using the same dictionary) for two documents
- $a + b$ is the word count vector of the document combining the original two
- $a - b$ how many more times each word appears in 1st document compared to 2nd

Audio mixing

- $a_1, \ldots, a_m$ are vectors representing audio signals over the same period of time
- βa_i is the same audio signal, but changed in volume (loudness) by the factor $|\beta_i|$
- linear combination $\beta_1 a_1 + \cdots + \beta_m a_m$ is a mixture of the audio tracks

Portfolio trading

- s is n-vector giving no. of shares of n assets in a portfolio
- b is n-vector giving no. of shares of assets that we buy ($b_i > 0$) or sell ($b_i < 0$)
- after trading, our portfolio is $s + b$, which is called the *trade vector* or *trade list*

Inner product examples

Weights, features, scores

- vectors of features f and weights w
- $w^T f = w_1 f_1 + w_2 f_2 + \cdots + w_n f_n$ is the total score
- example: features are associated with a loan applicant (e.g., age, income, . . .)
  - we can interpret $s = w^T f$ as a credit score
  - we can interpret w_i as the weight given to feature i in forming the score

Price quantity (cost)

- vectors of prices p and quantities q of n goods
- $p^T q = p_1 q_1 + p_2 q_2 + \cdots + p_n q_n$ is the total cost

Speed time

- vehicle travels over n segments with constant speed in each segment
- n-vector s gives the speed in the segments
- n-vector t gives the times taken to traverse the segments
- $s^T t$ is the total distance traveled

Polynomial evaluation

- n-vector c represents the coefficients of a polynomial p of degree $n-1$ or less:

$$p(x) = c_1 + c_2 x + \cdots + c_{n-1} x^{n-2} + c_n x^{n-1}$$

- t is number, and let $z = (1, t, t^2, \ldots, t^{n-1})$ be the n-vector of powers of t
- $c^T z = p(t)$ is the value of the polynomial p at the point t

Discounted total

- cash flow vector c where c_i is value at period i
- r is interest rate and $d = (1, 1/(1+r), \ldots, 1/(1+r)^{n-1})$
- $d^T c = c_1 + c_2/(1+r) + \ldots, c_n/(1+r)^{n-1}$ is the discounted total of cash flow
- called *net present value* (NPV) with interest rate r

Portfolio value

- s is an n-vector of holdings in shares of a portfolio of n assets
- p is an n-vector for the prices of the assets
- $p^T s$ is the total (or net) value of the portfolio

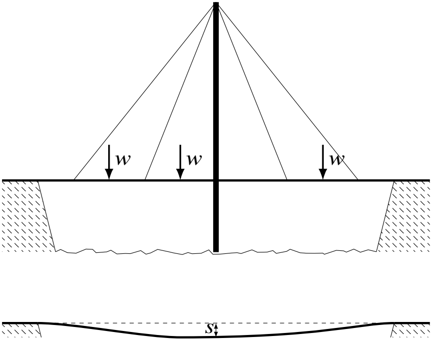
Portfolio return

- portfolio vector x with x_i representing dollar value of asset i
- r_i is rate (fraction) of return of asset i over the investment period:

$$p_i^{\mathrm{final}} = (1 + r_i) p_i^{\mathrm{init}}, \qquad r_i = \frac{p_i^{\mathrm{final}} - p_i^{\mathrm{init}}}{p_i^{\mathrm{init}}}$$

  p_i^{init} and p_i^{final} are the prices of asset i at the beginning and end of the period
- $r^T x = r_1 x_1 + \cdots + r_n x_n$ is total return in dollars over the period
- if w is the fractional (dollar) holdings of our portfolio, then $r^T w$ is rate of return
  - example: if $r^T w = 0.09$, then our portfolio return is 9%; if we had invested 10000 initially, we would have earned $\$900$

Sag of a bridge



- w gives the weight of the load on the bridge in n locations in metric tons
- s denote the distance that a specific point on the bridge sags, in millimeters
- $s \approx c^T w$ for some vector c
- coefficients c_i are called *compliances*, and give the sensitivity of the sag with respect to loads applied at the n locations
- vector c can be computed by (numerically) solving a partial differential equation

References and further readings

- S. Boyd and L. Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares,* Cambridge University Press, 2018.

- L. Vandenberghe. *EE133A lecture notes,* University of California, Los Angeles. (`http://www.seas.ucla.edu/~vandenbe/ee133a.html`)