Lecture slides for

Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares

Stephen Boyd   Lieven Vandenberghe

1. Vectors

Outline

Notation

Examples

Addition and scalar multiplication

Inner product

Complexity

Vectors

- a *vector* is an ordered list of numbers
- written as

$$\begin{bmatrix} -1.1 \\ 0.0 \\ 3.6 \\ -7.2 \end{bmatrix} \quad \text{or} \quad \begin{pmatrix} -1.1 \\ 0.0 \\ 3.6 \\ -7.2 \end{pmatrix}$$

  or $(-1.1, 0, 3.6, -7.2)$
- numbers in the list are the *elements* (*entries*, *coefficients*, *components*)
- number of elements is the *size* (*dimension*, *length*) of the vector
- vector above has dimension 4; its third entry is 3.6
- vector of size n is called an n-*vector*
- numbers are called *scalars*

Vectors via symbols

- we'll use symbols to denote vectors, *e.g.*, a, X, p, β, E^{aut}
- other conventions: $\mathbf{g}$, $\vec{a}$
- ith element of n-vector a is denoted a_i
- if a is vector above, $a_3 = 3.6$
- in a_i, i is the *index*
- for an n-vector, indexes run from $i = 1$ to $i = n$
- *warning:* sometimes a_i refers to the ith vector in a list of vectors
- two vectors a and b *of the same size* are *equal* if $a_i = b_i$ for all i
- we overload $=$ and write this as $a = b$

Block vectors

- suppose b, c, and d are vectors with sizes m, n, p
- the *stacked vector* or *concatenation* (of b, c, and d) is

$$a = \begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

- also called a *block vector*, with (block) entries b, c, d
- a has size $m + n + p$

$$a = (b_1, b_2, \ldots, b_m, c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_p)$$

Zero, ones, and unit vectors

- n-vector with all entries 0 is denoted 0_n or just 0
- n-vector with all entries 1 is denoted $\mathbf{1}_n$ or just $\mathbf{1}$
- a *unit vector* has one entry 1 and all others 0
- denoted e_i where i is entry that is 1
- unit vectors of length 3:

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

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: zero vectors, unit vectors

Outline

Notation

Examples

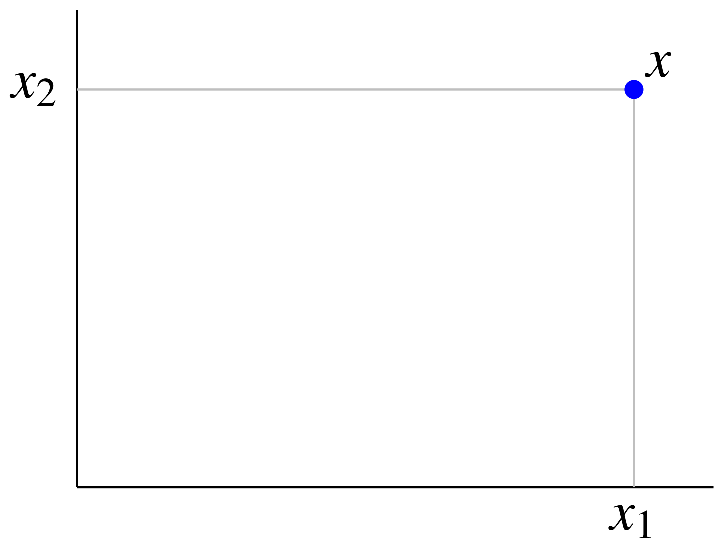
Addition and scalar multiplication

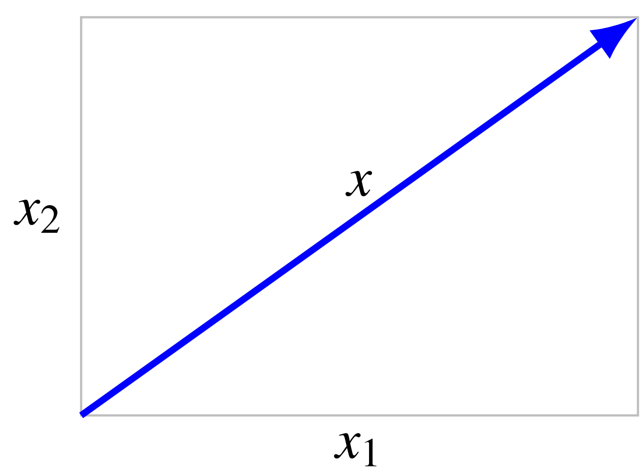
Inner product

Complexity

Location or displacement in 2-D or 3-D

2-vector (x_1, x_2) can represent a location or a displacement in 2-D





More examples

- color: $(\mathrm{R}, \mathrm{G}, \mathrm{B})$
- quantities of n different commodities (or resources), *e.g.*, bill of materials
- portfolio: entries give shares (or $ value or fraction) held in each of n assets, with negative meaning short positions
- cash flow: x_i is payment in period i to us
- audio: x_i is the acoustic pressure at sample time i (sample times are spaced 1/44100 seconds apart)
- features: x_i is the value of ith *feature* or *attribute* of an entity
- customer purchase: x_i is the total $ purchase of product i by a customer over some period
- word count: x_i is the number of times word i appears in a document

Word count vectors

- a short document:

> **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**.

- a small dictionary (left) and word count vector (right)

$$
\begin{array}{l}
\text{word} \\
\text{in} \\
\text{number} \\
\text{horse} \\
\text{the} \\
\text{document}
\end{array}
\qquad\qquad
\left[\begin{array}{c}
3 \\
2 \\
1 \\
0 \\
4 \\
2
\end{array}\right]
$$

- dictionaries used in practice are much larger

Outline

Vector addition

- n-vectors a and b can be added, with sum denoted $a + b$
- to get sum, add corresponding entries:

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

- subtraction is similar

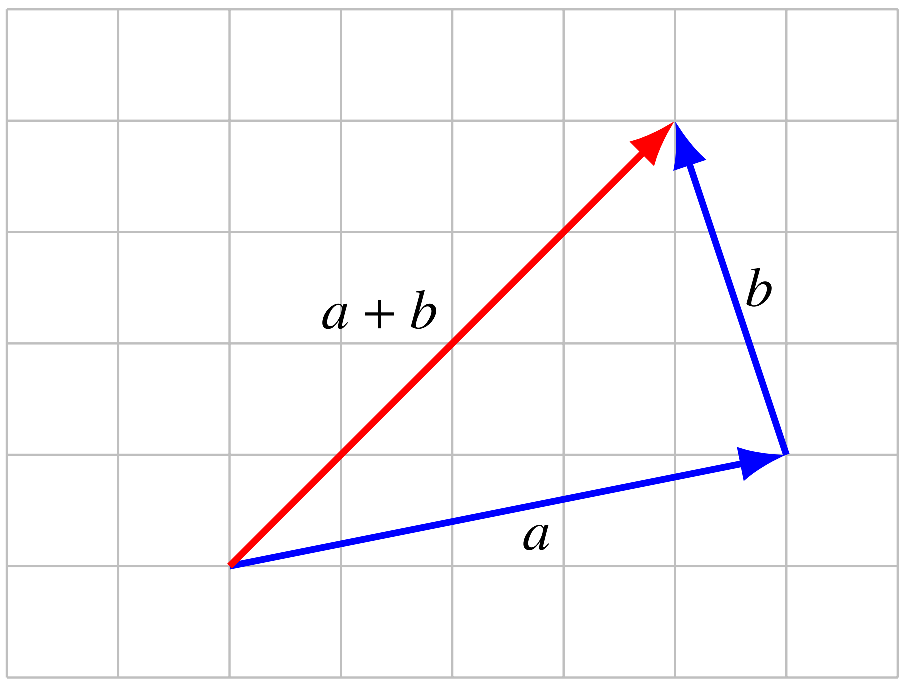
Properties of vector addition

- *commutative*: $a + b = b + a$
- *associative*: $(a + b) + c = a + (b + c)$
  (so we can write both as $a + b + c$)
- $a + 0 = 0 + a = a$
- $a - a = 0$

these are easy and boring to verify

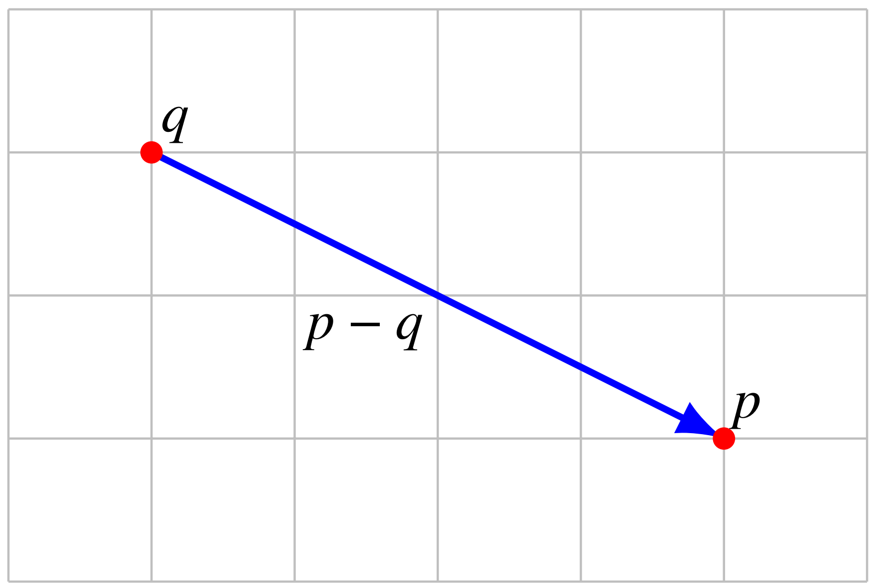
Adding displacements

if 3-vectors a and b are displacements, $a + b$ is the sum displacement



Displacement from one point to another

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



Scalar-vector multiplication

- scalar β and n-vector a can be multiplied

$$\beta a = (\beta a_1, \ldots, \beta a_n)$$

- also denoted $a\beta$
- example:

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

Properties of scalar-vector multiplication

- associative: $(\beta\gamma)a = \beta(\gamma a)$
- left distributive: $(\beta + \gamma)a = \beta a + \gamma a$
- right distributive: $\beta(a + b) = \beta a + \beta b$

these equations look innocent, but be sure you understand them perfectly

Linear combinations

- for vectors $a_1, \ldots, a_m$ and scalars $\beta_1, \ldots, \beta_m$,

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

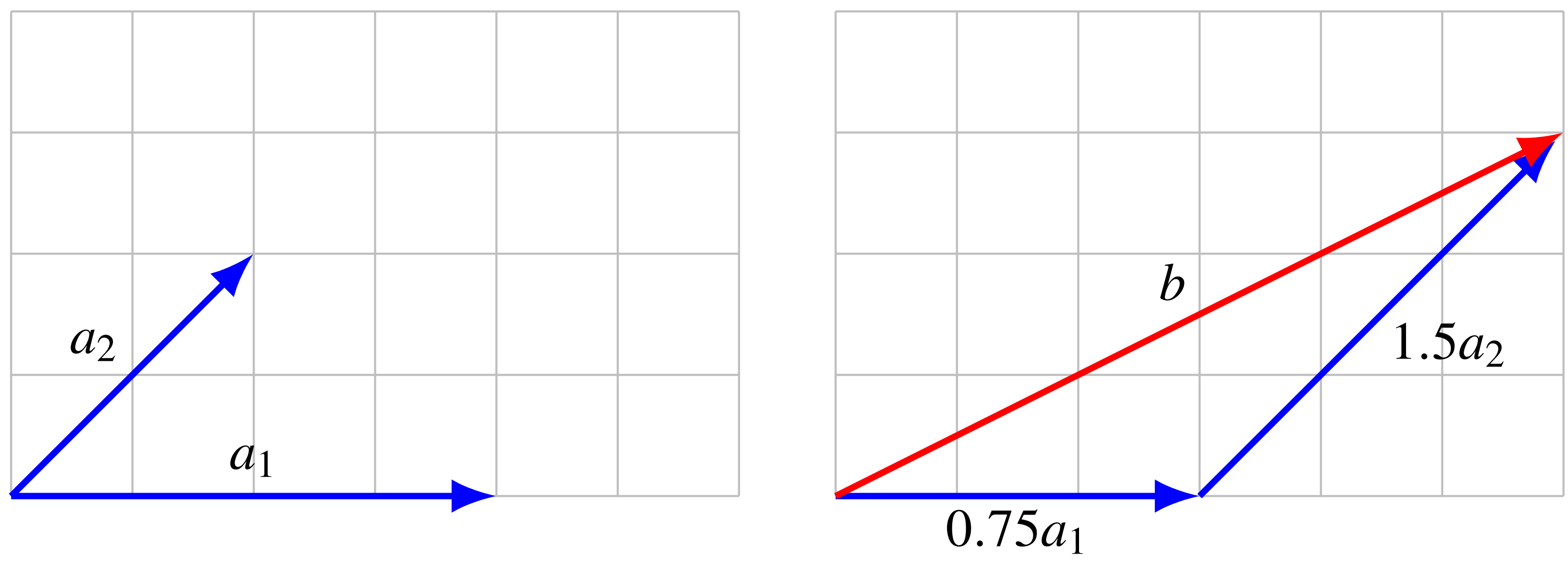
  is a *linear combination* of the vectors

- $\beta_1, \ldots, \beta_m$ are the *coefficients*
- a *very* important concept
- a simple identity: for any n-vector b,

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

Example

two vectors a_1 and a_2, and linear combination $b = 0.75a_1 + 1.5a_2$



Replicating a cash flow

- $c_1 = (1, -1.1, 0)$ is a \$1 loan from period 1 to 2 with 10% interest
- $c_2 = (0, 1, -1.1)$ is a \$1 loan from period 2 to 3 with 10% interest
- linear combination

  $$d = c_1 + 1.1c_2 = (1, 0, -1.21)$$

  is a two period loan with 10% compounded interest rate
- we have *replicated* a two period loan from two one period loans

Outline

Notation

Examples

Addition and scalar multiplication

Inner product

Complexity

Inner product

- *inner product* (or *dot product*) of n-vectors a and b is

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

- other notation used: $\langle a, b \rangle$, $\langle a | b \rangle$, (a, b), $a \cdot b$

- 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

- $a^T b = b^T a$
- $(\gamma a)^T b = \gamma(a^T b)$
- $(a + b)^T c = a^T c + b^T c$

can combine these to get, for example,

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

General examples

- $e_i^T a = a_i$ (picks out ith entry)
- $\mathbf{1}^T a = a_1 + \cdots + a_n$ (sum of entries)
- $a^T a = a_1^2 + \cdots + a_n^2$ (sum of squares of entries)

Examples

- w is weight vector, f is feature vector; $w^T f$ is score
- p is vector of prices, q is vector of quantities; $p^T q$ is total cost
- c is cash flow, d is discount vector (with interest rate r):

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

  $d^T c$ is net present value (NPV) of cash flow
- s gives portfolio holdings (in shares), p gives asset prices; $p^T s$ is total portfolio value

Outline

Notation

Examples

Addition and scalar multiplication

Inner product

Complexity

Flop counts

- computers store (real) numbers in *floating-point format*
- basic arithmetic operations (addition, multiplication, ...) are called *floating point operations* or flops
- complexity of an algorithm or operation: total number of flops needed, as function of the input dimension(s)
- this can be *very grossly approximated*
- crude approximation of time to execute: (flops needed)/(computer speed)
- current computers are around 1Gflop/sec (10^9 flops/sec)
- but this can vary by factor of 100

Complexity of vector addition, inner product

- $x + y$ needs n additions, so: n flops
- $x^T y$ needs n multiplications, $n - 1$ additions so: $2n - 1$ flops
- we simplify this to $2n$ (or even n) flops for $x^T y$
- and much less when x or y is sparse

2. Linear functions

Outline

Linear and affine functions

Taylor approximation

Regression model

Superposition and linear functions

- $f : \mathbf{R}^n \to \mathbf{R}$ means f is a function mapping n-vectors to numbers
- f satisfies the *superposition property* if

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

  holds for all numbers α, β, and all n-vectors x, y
- be sure to parse this very carefully!
- a function that satisfies superposition is called *linear*

The inner product function

- with a an n-vector, the function

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

  is the *inner product function*

- $f(x)$ is a weighted sum of the entries of x
- the inner product function is linear:

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

…and all linear functions are inner products

- suppose $f : \mathbf{R}^n \to \mathbf{R}$ is linear
- then it can be expressed as $f(x) = a^T x$ for some a
- specifically: $a_i = f(e_i)$
- follows from

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

Affine functions

- a function that is linear plus a constant is called *affine*
- general form is $f(x) = a^T x + b$, with a an n-vector and b a scalar
- a function $f : \mathbf{R}^n \to \mathbf{R}$ is affine if and only if

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

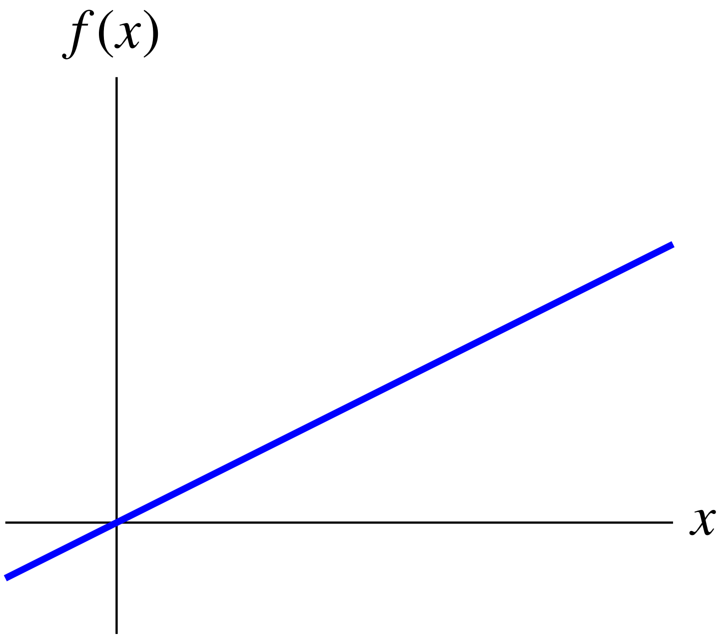
  holds for all α, β with $\alpha + \beta = 1$, and all n-vectors x, y

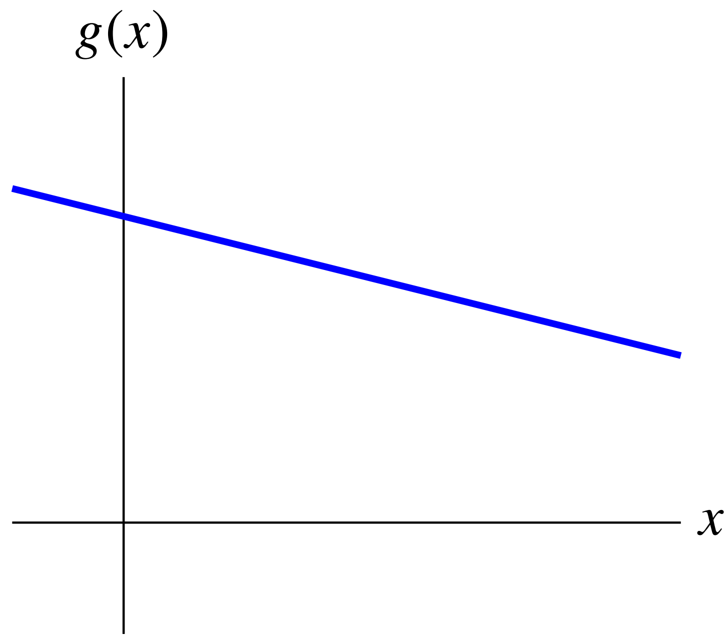
- sometimes (ignorant) people refer to affine functions as linear

Linear versus affine functions

f is linear

g is affine, not linear





Outline

Linear and affine functions

Taylor approximation

Regression model

First-order Taylor approximation

- suppose $f : \mathbf{R}^n \to \mathbf{R}$
- *first-order Taylor approximation* of f, near point z:

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

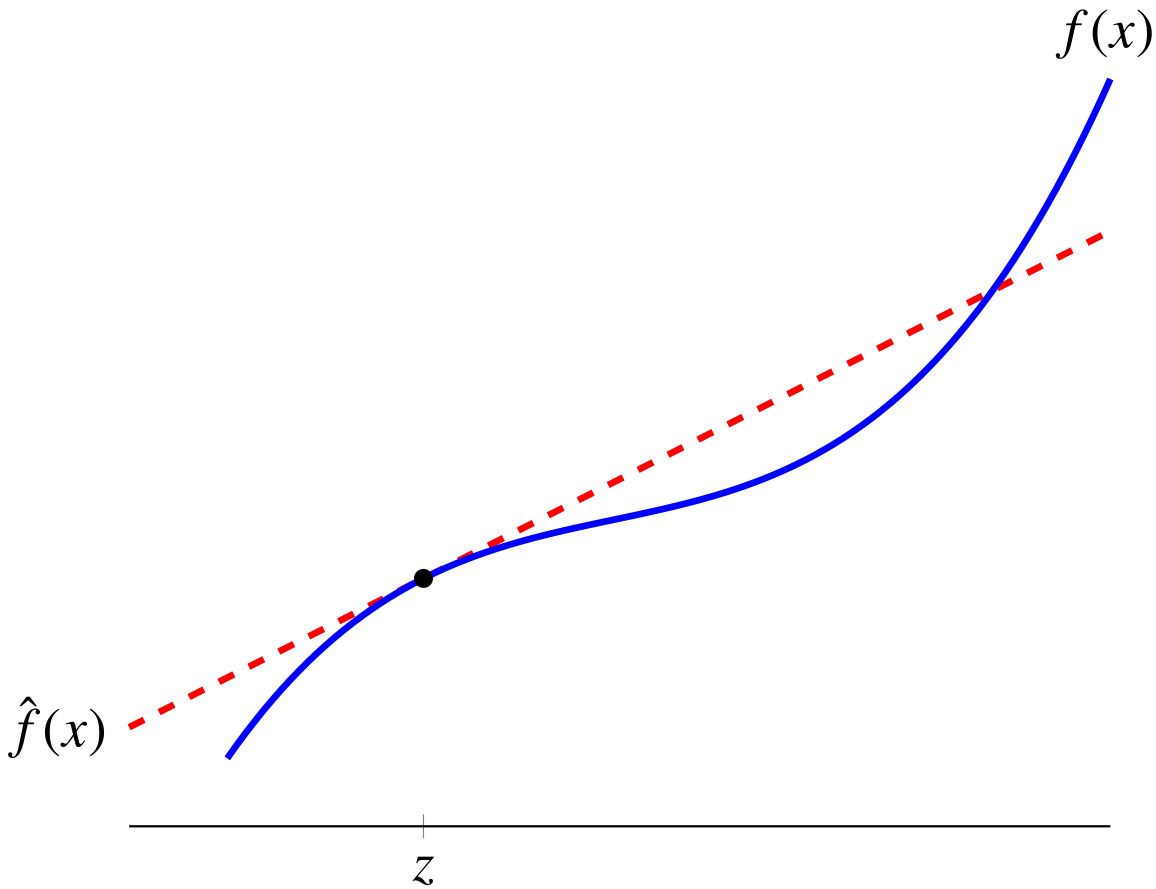
- $\hat{f}(x)$ is *very* close to $f(x)$ when x_i are all near z_i
- $\hat{f}$ is an affine function of x
- can write using inner product as

$$\hat{f}(x) = f(z) + \nabla f(z)^T (x - z)$$

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

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

Example



Outline

Linear and affine functions

Taylor approximation

Regression model

Regression model

- *regression model* is (the affine function of x)

$$\hat{y} = x^T \beta + v$$

- x is a feature vector; its elements x_i are called *regressors*
- n-vector β is the *weight vector*
- scalar v is the *offset*
- scalar $\hat{y}$ is the *prediction*
  (of some actual outcome or *dependent variable*, denoted y)

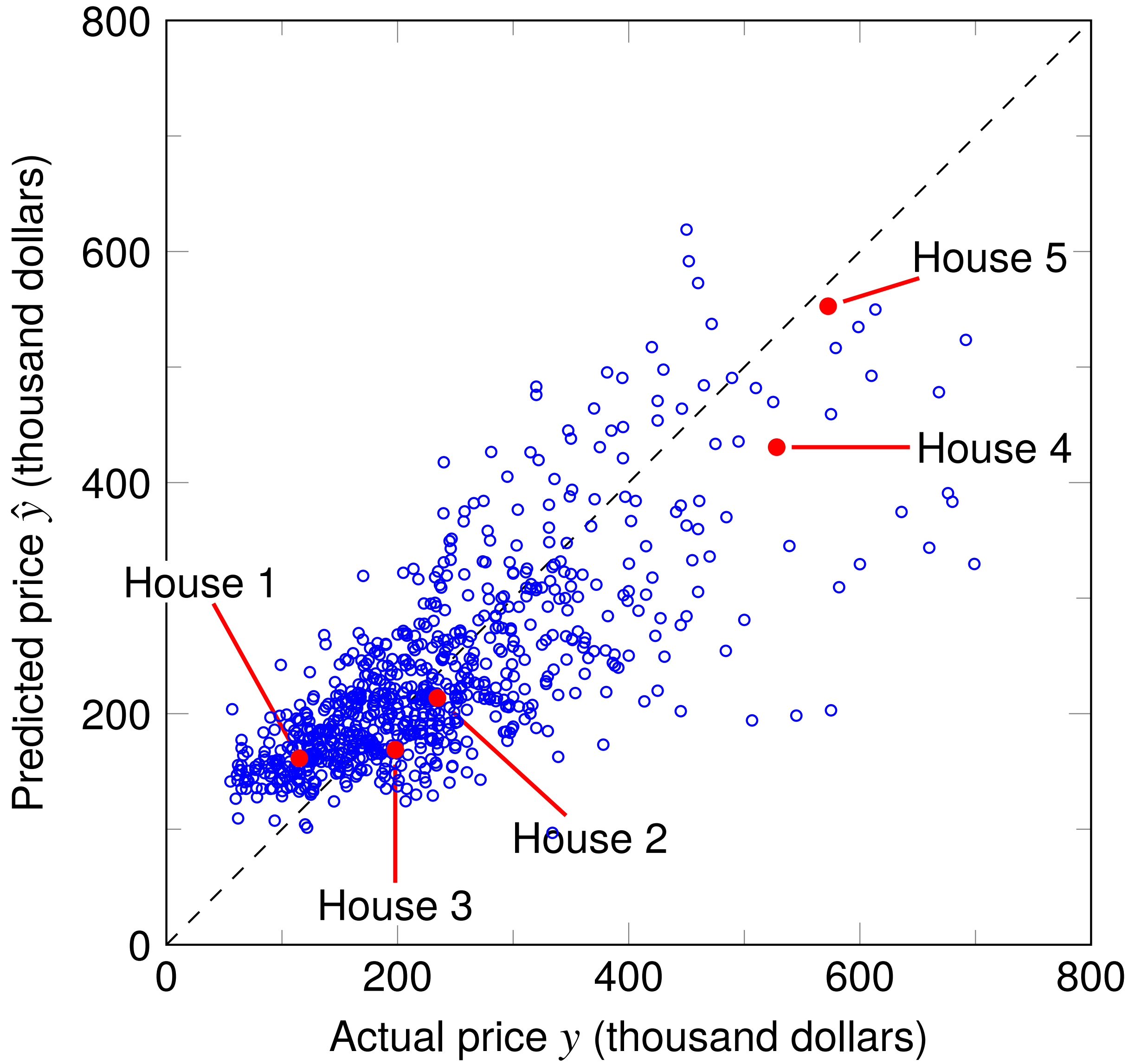
Example

- y is selling price of house in $\$1000$ (in some location, over some period)
- regressor is
$$x = (\text{house area, \# bedrooms})$$
(house area in 1000 sq.ft.)
- regression model weight vector and offset are
$$\beta = (148.73, -18.85), \qquad v = 54.40$$
- we'll see later how to guess β and v from sales data

Example

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



800
600
400
200
0
0
200
400
600
800
Predicted price $\hat{y}$ (thousand dollars)
Actual price y (thousand dollars)
House 1
House 2
House 3
House 4
House 5

3. Norm and distance

Outline

Norm

Distance

Standard deviation

Angle

Norm

- the *Euclidean norm* (or just *norm*) of an n-vector x is

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{x^T x}$$

- used to measure the size of a vector

- reduces to absolute value for $n = 1$

Properties

for any n-vectors x and y, and any scalar β

- *homogeneity:* $\|\beta x\| = |\beta|\|x\|$
- *triangle inequality:* $\|x + y\| \leq \|x\| + \|y\|$
- *nonnegativity:* $\|x\| \geq 0$
- *definiteness:* $\|x\| = 0$ only if $x = 0$

easy to show except triangle inequality, which we show later

RMS value

- *mean-square value* of n-vector x is

$$\frac{x_1^2 + \cdots + x_n^2}{n} = \frac{\|x\|^2}{n}$$

- *root-mean-square value* (RMS value) is

$$\mathbf{rms}(x) = \sqrt{\frac{x_1^2 + \cdots + x_n^2}{n}} = \frac{\|x\|}{\sqrt{n}}$$

- $\mathbf{rms}(x)$ gives 'typical' value of $|x_i|$
- *e.g.*, $\mathbf{rms}(\mathbf{1}) = 1$ (independent of n)
- RMS value useful for comparing sizes of vectors of different lengths

Norm of block vectors

- suppose a, b, c are vectors
- $\|(a, b, c)\|^2 = a^T a + b^T b + c^T c = \|a\|^2 + \|b\|^2 + \|c\|^2$
- so we have

$$\|(a, b, c)\| = \sqrt{\|a\|^2 + \|b\|^2 + \|c\|^2} = \|(\|a\|, \|b\|, \|c\|)\|$$

  (parse RHS very carefully!)

- we'll use these ideas later

Chebyshev inequality

- suppose that k of the numbers $|x_1|, \ldots, |x_n|$ are $\geq a$
- then k of the numbers $x_1^2, \ldots, x_n^2$ are $\geq a^2$
- so $\|x\|^2 = x_1^2 + \cdots + x_n^2 \geq ka^2$
- so we have $k \leq \|x\|^2/a^2$
- number of x_i with $|x_i| \geq a$ is no more than $\|x\|^2/a^2$
- this is the *Chebyshev inequality*
- in terms of RMS value:

$$\text{fraction of entries with } |x_i| \geq a \text{ is no more than } \left(\frac{\mathbf{rms}(x)}{a}\right)^2$$

- example: no more than 4% of entries can satisfy $|x_i| \geq 5\ \mathbf{rms}(x)$

Outline

Norm

Distance

Standard deviation

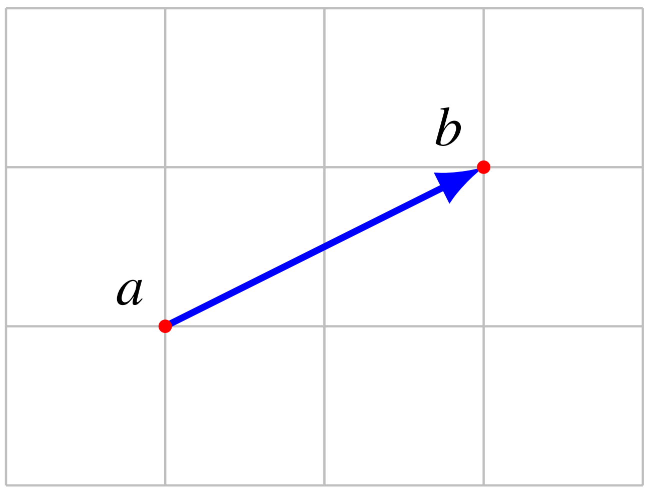
Angle

Distance

- (Euclidean) *distance* between n-vectors a and b is

$$\mathbf{dist}(a, b) = \|a - b\|$$

- agrees with ordinary distance for $n = 1, 2, 3$



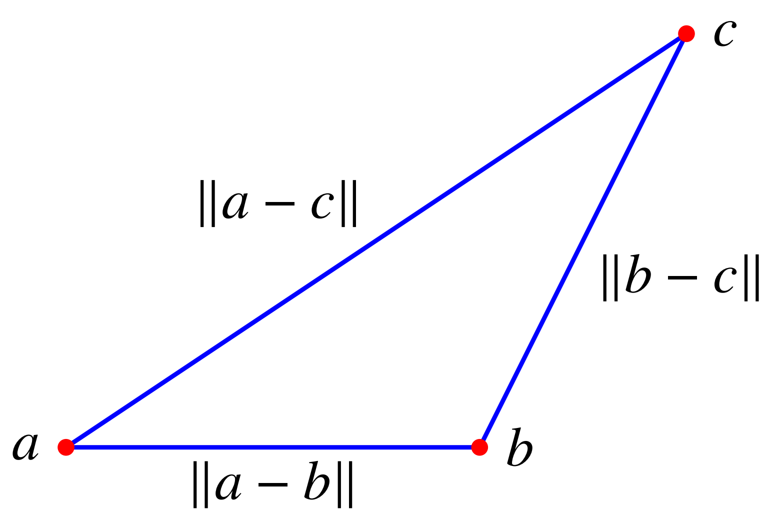
- $\mathbf{rms}(a - b)$ is the *RMS deviation* between a and b

Triangle inequality

- triangle with vertices at positions a, b, c
- edge lengths are $\|a-b\|$, $\|b-c\|$, $\|a-c\|$
- by triangle inequality

$$\|a-c\| = \|(a-b)+(b-c)\| \leq \|a-b\| + \|b-c\|$$

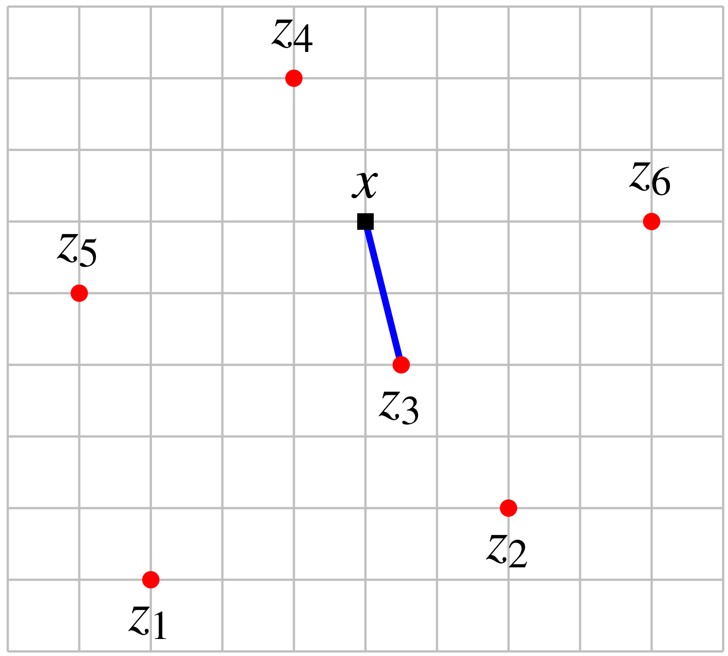
i.e., third edge length is no longer than sum of other two



Feature distance and nearest neighbors

- if x and y are feature vectors for two entities, $\|x - y\|$ is the *feature distance*
- if $z_1, \ldots, z_m$ is a list of vectors, z_j is the *nearest neighbor* of x if

$$\|x - z_j\| \leq \|x - z_i\|, \quad i = 1, \ldots, m$$



- these simple ideas are very widely used

Document dissimilarity

- 5 Wikipedia articles: ‘Veterans Day’, ‘Memorial Day’, ‘Academy Awards’, ‘Golden Globe Awards’, ‘Super Bowl’
- word count histograms, dictionary of 4423 words
- pairwise distances shown below

	Veterans Day	Memorial Day	Academy Awards	Golden Globe Awards	Super Bowl
Veterans Day	0	0.095	0.130	0.153	0.170
Memorial Day	0.095	0	0.122	0.147	0.164
Academy A.	0.130	0.122	0	0.108	0.164
Golden Globe A.	0.153	0.147	0.108	0	0.181
Super Bowl	0.170	0.164	0.164	0.181	0

Outline

Norm

Distance

Standard deviation

Angle

Standard deviation

- for n-vector x, $\mathbf{avg}(x) = \mathbf{1}^T x/n$
- *de-meaned vector* is $\tilde{x} = x - \mathbf{avg}(x)\mathbf{1}$ (so $\mathbf{avg}(\tilde{x}) = 0$)
- *standard deviation* of x is

$$\mathbf{std}(x) = \mathbf{rms}(\tilde{x}) = \frac{\|x - (\mathbf{1}^T x/n)\mathbf{1}\|}{\sqrt{n}}$$

- $\mathbf{std}(x)$ gives 'typical' amount x_i vary from $\mathbf{avg}(x)$
- $\mathbf{std}(x) = 0$ only if $x = \alpha\mathbf{1}$ for some α
- greek letters μ, σ commonly used for mean, standard deviation
- a basic formula:

$$\mathbf{rms}(x)^2 = \mathbf{avg}(x)^2 + \mathbf{std}(x)^2$$

Outline

Norm

Distance

Standard deviation

Angle

Angle

- *angle* between two nonzero vectors a, b defined as

$$\angle(a,b) = \arccos\left(\frac{a^T b}{\|a\|\,\|b\|}\right)$$

- $\angle(a,b)$ is the number in $[0,\pi]$ that satisfies

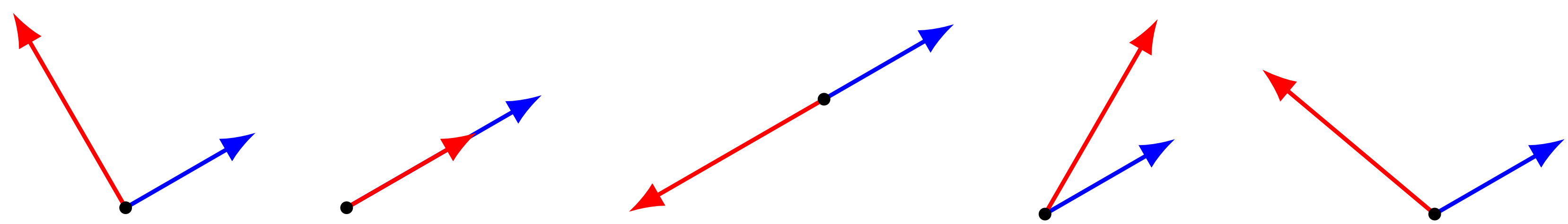
$$a^T b = \|a\|\,\|b\| \cos\left(\angle(a,b)\right)$$

- coincides with ordinary angle between vectors in 2-D and 3-D

Classification of angles

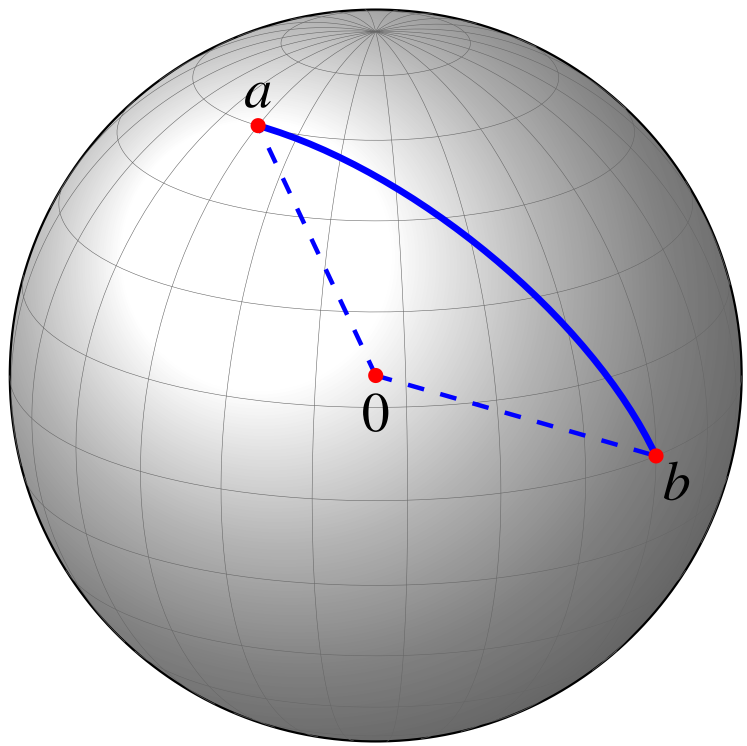
$\theta = \angle(a, b)$

- $\theta = \pi/2 = 90^\circ$: a and b are *orthogonal*, written $a \perp b$ ($a^T b = 0$)
- $\theta = 0$: a and b are *aligned* ($a^T b = \|a\|\|b\|$)
- $\theta = \pi = 180^\circ$: a and b are *anti-aligned* ($a^T b = -\|a\|\,\|b\|$)
- $\theta \leq \pi/2 = 90^\circ$: a and b make an *acute angle* ($a^T b \geq 0$)
- $\theta \geq \pi/2 = 90^\circ$: a and b make an *obtuse angle* ($a^T b \leq 0$)



Spherical distance

if a, b are on sphere of radius R, distance *along the sphere* is $R\angle(a,b)$



Document dissimilarity by angles

- measure dissimilarity by angle of word count histogram vectors
- pairwise angles (in degrees) for 5 Wikipedia pages shown below

	Veterans Day	Memorial Day	Academy Awards	Golden Globe Awards	Super Bowl
Veterans Day	0	60.6	85.7	87.0	87.7
Memorial Day	60.6	0	85.6	87.5	87.5
Academy A.	85.7	85.6	0	58.7	85.7
Golden Globe A.	87.0	87.5	58.7	0	86.0
Super Bowl	87.7	87.5	86.1	86.0	0

Correlation coefficient

- vectors a and b, and de-meaned vectors

$$\tilde{a} = a - \mathbf{avg}(a)\mathbf{1}, \qquad \tilde{b} = b - \mathbf{avg}(b)\mathbf{1}$$

- *correlation coefficient* (between a and b, with $\tilde{a} \neq 0$, $\tilde{b} \neq 0$)

$$\rho = \frac{\tilde{a}^T \tilde{b}}{\|\tilde{a}\| \, \|\tilde{b}\|}$$

- $\rho = \cos \angle(\tilde{a}, \tilde{b})$
  - $\rho = 0$: a and b are *uncorrelated*
  - $\rho > 0.8$ (or so): a and b are *highly correlated*
  - $\rho < -0.8$ (or so): a and b are *highly anti-correlated*
- very roughly: highly correlated means a_i and b_i are typically both above (below) their means together

Outline

Linear independence

Basis

Orthonormal vectors

Gram–Schmidt algorithm

Linear dependence

- set of n-vectors $\{a_1, \ldots, a_k\}$ (with $k \geq 1$) is *linearly dependent* if

  $$\beta_1 a_1 + \cdots + \beta_k a_k = 0$$

  holds for some $\beta_1, \ldots, \beta_k$, that are not all zero
- equivalent to: at least one a_i is a linear combination of the others
- we say '$a_1, \ldots, a_k$ are linearly dependent'

- $\{a_1\}$ is linearly dependent only if $a_1 = 0$
- $\{a_1, a_2\}$ is linearly dependent only if one a_i is a multiple of the other
- for more than two vectors, there is no simple to state condition

Example

- the vectors

$$a_1 = \begin{bmatrix} 0.2 \\ -7 \\ 8.6 \end{bmatrix}, \qquad a_2 = \begin{bmatrix} -0.1 \\ 2 \\ -1 \end{bmatrix}, \qquad a_3 = \begin{bmatrix} 0 \\ -1 \\ 2.2 \end{bmatrix}$$

  are linearly dependent, since $a_1 + 2a_2 - 3a_3 = 0$

- can express any of them as linear combination of the other two, *e.g.*,

$$a_2 = (-1/2)a_1 + (3/2)a_3$$

Linear independence

- set of n-vectors $\{a_1, \ldots, a_k\}$ (with $k \geq 1$) is *linearly independent* if it is not linearly dependent, *i.e.*,

$$\beta_1 a_1 + \cdots + \beta_k a_k = 0$$

  holds only when $\beta_1 = \cdots = \beta_k = 0$

- we say '$a_1, \ldots, a_k$ are linearly independent'
- equivalent to: no a_i is a linear combination of the others

- example: the unit n-vectors $e_1, \ldots, e_n$ are linearly independent

Linear combinations of linearly independent vectors

- suppose x is linear combination of linearly independent vectors $a_1, \ldots, a_k$:

$$x = \beta_1 a_1 + \cdots + \beta_k a_k$$

- the coefficients $\beta_1, \ldots, \beta_k$ are *unique*, *i.e.*, if

$$x = \gamma_1 a_1 + \cdots + \gamma_k a_k$$

  then $\beta_i = \gamma_i$ for $i = 1, \ldots, k$
- this means that (in principle) we can deduce the coefficients from x
- to see why, note that

$$(\beta_1 - \gamma_1) a_1 + \cdots + (\beta_k - \gamma_k) a_k = 0$$

  and so (by linear independence) $\beta_1 - \gamma_1 = \cdots = \beta_k - \gamma_k = 0$

Outline

Linear independence

Basis

Orthonormal vectors

Gram–Schmidt algorithm

Independence-dimension inequality

- *a linearly independent set of n-vectors can have at most n elements*
- put another way: *any set of $n+1$ or more n-vectors is linearly dependent*

Basis

- a set of n linearly independent n-vectors $a_1, \ldots, a_n$ is called a *basis*
- any n-vector b can be expressed as a linear combination of them:

$$b = \beta_1 a_1 + \cdots + \beta_n a_n$$

  for some $\beta_1, \ldots, \beta_n$

- and these coefficients are unique
- formula above is called *expansion of b in the $a_1, \ldots, a_n$ basis*
- example: $e_1, \ldots, e_n$ is a basis, expansion of b is

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

Outline

Linear independence

Basis

Orthonormal vectors

Gram–Schmidt algorithm

Orthonormal vectors

- set of n-vectors $a_1, \ldots, a_k$ are *(mutually) orthogonal* if $a_i \perp a_j$ for $i \neq j$
- they are *normalized* if $\|a_i\| = 1$ for $i = 1, \ldots, k$
- they are *orthonormal* if both hold
- can be expressed using inner products as

$$a_i^T a_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

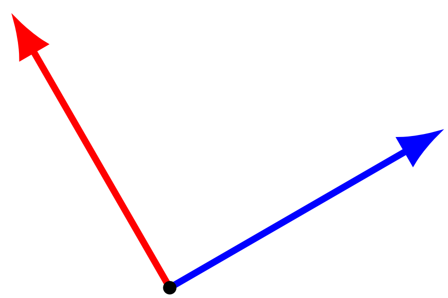
- orthonormal sets of vectors are linearly independent
- by independence-dimension inequality, must have $k \leq n$
- when $k = n$, $a_1, \ldots, a_n$ are an *orthonormal basis*

Examples of orthonormal bases

- standard unit n-vectors $e_1, \ldots, e_n$
- the 3-vectors

$$\begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \qquad \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \qquad \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

- the 2-vectors shown below



Orthonormal expansion

- if $a_1, \ldots, a_n$ is an orthonormal basis, we have for any n-vector x

$$x = (a_1^T x)a_1 + \cdots + (a_n^T x)a_n$$

- called *orthonormal expansion of* x (in the orthonormal basis)
- to verify formula, take inner product of both sides with a_i

6. Matrices

Outline

Matrices

Matrix-vector multiplication

Examples

Matrices

- a *matrix* is a rectangular array of numbers, *e.g.*,

$$\begin{bmatrix} 0 & 1 & -2.3 & 0.1 \\ 1.3 & 4 & -0.1 & 0 \\ 4.1 & -1 & 0 & 1.7 \end{bmatrix}$$

- its *size* is given by (row dimension) $\times$ (column dimension)
  e.g., matrix above is 3×4
- *elements* also called *entries* or *coefficients*
- B_{ij} is i,j element of matrix B
- i is the *row index*, j is the *column index*; indexes start at 1
- two matrices are *equal* (denoted with $=$) if they are the same size and corresponding entries are equal

Matrix shapes

an $m \times n$ matrix A is

- *tall* if $m > n$
- *wide* if $m < n$
- *square* if $m = n$

Column and row vectors

- we consider an $n \times 1$ matrix to be an n-vector
- we consider a 1×1 matrix to be a number
- a $1 \times n$ matrix is called a *row vector*, *e.g.*,

$$\begin{bmatrix} 1.2 & -0.3 & 1.4 & 2.6 \end{bmatrix}$$

which is *not* the same as the (column) vector

$$\begin{bmatrix} 1.2 \\ -0.3 \\ 1.4 \\ 2.6 \end{bmatrix}$$

Columns and rows of a matrix

- suppose A is an $m \times n$ matrix with entries A_{ij} for $i = 1, \ldots, m$, $j = 1, \ldots, n$
- its jth *column* is (the m-vector)

$$\begin{bmatrix} A_{1j} \\ \vdots \\ A_{mj} \end{bmatrix}$$

- its ith *row* is (the n-row-vector)

$$\begin{bmatrix} A_{i1} & \cdots & A_{in} \end{bmatrix}$$

- *slice* of matrix: $A_{p:q,r:s}$ is the $(q-p+1) \times (s-r+1)$ matrix

$$A_{p:q,r:s} = \begin{bmatrix} A_{pr} & A_{p,r+1} & \cdots & A_{ps} \\ A_{p+1,r} & A_{p+1,r+1} & \cdots & A_{p+1,s} \\ \vdots & \vdots & & \vdots \\ A_{qr} & A_{q,r+1} & \cdots & A_{qs} \end{bmatrix}$$

Block matrices

- we can form *block matrices*, whose entries are matrices, such as

$$A = \begin{bmatrix} B & C \\ D & E \end{bmatrix}$$

where B, C, D, and E are matrices (called *submatrices* or *blocks* of A)
- matrices in each block row must have same height (row dimension)
- matrices in each block column must have same width (column dimension)
- example: if

$$B = \begin{bmatrix} 0 & 2 & 3 \end{bmatrix}, \quad C = \begin{bmatrix} -1 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & 2 & 1 \\ 1 & 3 & 5 \end{bmatrix}, \quad E = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

then

$$\begin{bmatrix} B & C \\ D & E \end{bmatrix} = \begin{bmatrix} 0 & 2 & 3 & -1 \\ 2 & 2 & 1 & 4 \\ 1 & 3 & 5 & 4 \end{bmatrix}$$

Column and row representation of matrix

- A is an $m \times n$ matrix
- can express as block matrix with its (m-vector) columns $a_1, \ldots, a_n$

$$A = \left[\begin{array}{cccc} a_1 & a_2 & \cdots & a_n \end{array} \right]$$

- or as block matrix with its (n-row-vector) rows $b_1, \ldots, b_m$

$$A = \left[\begin{array}{c} b_1 \\ b_2 \\ \vdots \\ b_m \end{array} \right]$$

Examples

- *image:* X_{ij} is i,j pixel value in a monochrome image
- *rainfall data:* A_{ij} is rainfall at location i on day j
- *multiple asset returns:* R_{ij} is return of asset j in period i
- *contingency table:* A_{ij} is number of objects with first attribute i and second attribute j
- *feature matrix:* X_{ij} is value of feature i for entity j

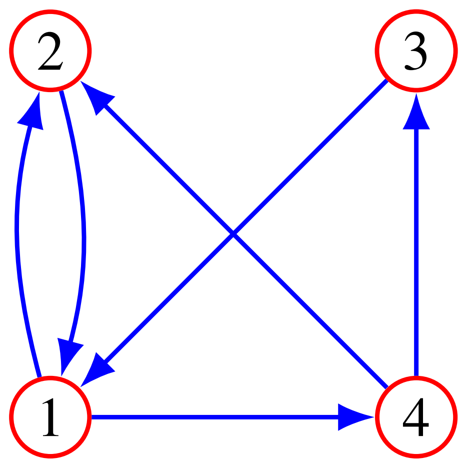
in each of these, what do the rows and columns mean?

Graph or relation

- a *relation* is a set of pairs of *objects*, labeled $1, \ldots, n$, such as

$$\mathcal{R} = \{(1,2),\ (1,3),\ (2,1),\ (2,4),\ (3,4),\ (4,1)\}$$

- same as *directed graph*



- can be represented as $n \times n$ matrix with $A_{ij} = 1$ if $(i,j) \in \mathcal{R}$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Special matrices

- $m \times n$ *zero matrix* has all entries zero, written as $0_{m\times n}$ or just 0
- *identity matrix* is square matrix with $I_{ii} = 1$ and $I_{ij} = 0$ for $i \neq j$, *e.g.*,

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- *sparse matrix:* most entries are zero
  - examples: 0 and I
  - can be stored and manipulated efficiently
  - $\mathbf{nnz}(A)$ is number of nonzero entries

Diagonal and triangular matrices

- *diagonal matrix*: square matrix with $A_{ij} = 0$ when $i \neq j$
- $\textbf{diag}(a_1, \ldots, a_n)$ denotes the diagonal matrix with $A_{ii} = a_i$ for $i = 1, \ldots, n$
- example:

$$\textbf{diag}(0.2, -3, 1.2) = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 1.2 \end{bmatrix}$$

- *lower triangular matrix*: $A_{ij} = 0$ for $i < j$
- *upper triangular matrix*: $A_{ij} = 0$ for $i > j$
- examples:

$$\begin{bmatrix} 1 & -1 & 0.7 \\ 0 & 1.2 & -1.1 \\ 0 & 0 & 3.2 \end{bmatrix} \text{ (upper triangular)}, \qquad \begin{bmatrix} -0.6 & 0 \\ -0.3 & 3.5 \end{bmatrix} \text{ (lower triangular)}$$

Transpose

- the *transpose* of an $m \times n$ matrix A is denoted A^T, and defined by

$$(A^T)_{ij} = A_{ji}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m$$

- for example,

$$\begin{bmatrix} 0 & 4 \\ 7 & 0 \\ 3 & 1 \end{bmatrix}^T = \begin{bmatrix} 0 & 7 & 3 \\ 4 & 0 & 1 \end{bmatrix}$$

- transpose converts column to row vectors (and vice versa)
- $(A^T)^T = A$

Addition, subtraction, and scalar multiplication

- (just like vectors) we can add or subtract matrices of the same size:

$$(A+B)_{ij} = A_{ij} + B_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

  (subtraction is similar)

- scalar multiplication:

$$(\alpha A)_{ij} = \alpha A_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

- many obvious properties, *e.g.*,

$$A + B = B + A, \qquad \alpha(A+B) = \alpha A + \alpha B, \qquad (A+B)^T = A^T + B^T$$

Matrix norm

- for $m \times n$ matrix A, we define

$$\|A\| = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2 \right)^{1/2}$$

- agrees with vector norm when $n = 1$
- satisfies norm properties:

$$\|\alpha A\| = |\alpha| \|A\|$$
$$\|A + B\| \leq \|A\| + \|B\|$$
$$\|A\| \geq 0$$
$$\|A\| = 0 \text{ only if } A = 0$$

- distance between two matrices: $\|A - B\|$
- (there are other matrix norms, which we won't use)

Outline

Matrices

Matrix-vector multiplication

Examples

Matrix-vector product

- *matrix-vector product* of $m \times n$ matrix A, n-vector x, denoted $y = Ax$, with

$$y_i = A_{i1}x_1 + \cdots + A_{in}x_n, \quad i = 1, \ldots, m$$

- for example,

$$\begin{bmatrix} 0 & 2 & -1 \\ -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ -4 \end{bmatrix}$$

Row interpretation

- $y = Ax$ can be expressed as

$$y_i = b_i^T x, \quad i = 1, \ldots, m$$

  where $b_1^T, \ldots, b_m^T$ are rows of A

- so $y = Ax$ is a 'batch' inner product of all rows of A with x
- example: $A\mathbf{1}$ is vector of row sums of matrix A

Column interpretation

- $y = Ax$ can be expressed as

$$y = x_1 a_1 + x_2 a_2 + \cdots + x_n a_n$$

  where $a_1, \ldots, a_n$ are columns of A

- so $y = Ax$ is linear combination of columns of A, with coefficients $x_1, \ldots, x_n$
- important example: $Ae_j = a_j$
- columns of A are linearly independent if $Ax = 0$ implies $x = 0$

Outline

Matrices

Matrix-vector multiplication

Examples

General examples

- $0x = 0$, *i.e.*, multiplying by zero matrix gives zero
- $Ix = x$, *i.e.*, multiplying by identity matrix does nothing
- inner product $a^T b$ is matrix-vector product of $1 \times n$ matrix a^T and n-vector b
- $\tilde{x} = Ax$ is de-meaned version of x, with

$$A = \begin{bmatrix} 1 - 1/n & -1/n & \cdots & -1/n \\ -1/n & 1 - 1/n & \cdots & -1/n \\ \vdots & & \ddots & \vdots \\ -1/n & -1/n & \cdots & 1 - 1/n \end{bmatrix}$$

Difference matrix

- $(n-1) \times n$ *difference matrix* is

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ & & \ddots & \ddots & & & \\ & & & \ddots & \ddots & & \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

  $y = Dx$ is $(n-1)$-vector of differences of consecutive entries of x:

$$Dx = \begin{bmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{bmatrix}$$

- *Dirichlet energy:* $\|Dx\|^2$ is measure of wiggliness for x a time series

Return matrix – portfolio vector

- R is $T \times n$ matrix of asset returns
- R_{ij} is return of asset j in period i (say, in percentage)
- n-vector w gives portfolio (investments in the assets)
- T-vector Rw is time series of the portfolio return
- $\mathbf{avg}(Rw)$ is the portfolio (mean) return, $\mathbf{std}(Rw)$ is its risk

Feature matrix – weight vector

- $X = [x_1 \ \cdots \ x_N]$ is $n \times N$ *feature matrix*
- column x_j is feature n-vector for object or example j
- X_{ij} is value of feature i for example j
- n-vector w is weight vector
- $s = X^T w$ is vector of scores for each example; $s_j = x_j^T w$

Input – output matrix

- A is $m \times n$ matrix
- $y = Ax$
- n-vector x is *input* or *action*
- m-vector y is *output* or *result*
- A_{ij} is the factor by which y_i depends on x_j
- A_{ij} is the *gain* from input j to output i
- *e.g.*, if A is lower triangular, then y_i only depends on $x_1, \ldots, x_i$

Complexity

- $m \times n$ matrix stored A as $m \times n$ array of numbers (for sparse A, store only $\mathbf{nnz}(A)$ nonzero values)
- matrix addition, scalar-matrix multiplication cost mn flops
- matrix-vector multiplication costs $m(2n-1) \approx 2mn$ flops (for sparse A, around $2\mathbf{nnz}(A)$ flops)

7. Matrix examples

Outline

Geometric transformations

Selectors

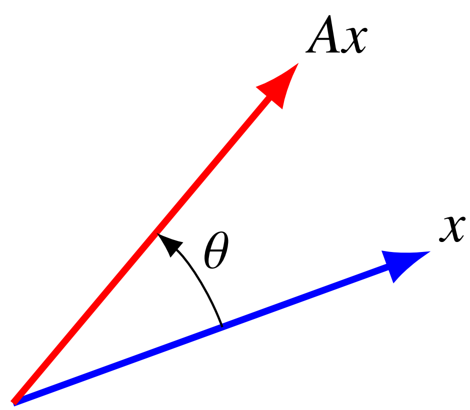
Incidence matrix

Convolution

Geometric transformations

- many geometric transformations and mappings of 2-D and 3-D vectors can be represented via matrix multiplication $y = Ax$
- for example, rotation by θ:

$$y = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} x$$



(to get the entries, look at Ae_1 and Ae_2)

Outline

Geometric transformations

Selectors

Incidence matrix

Convolution

Selectors

- an $m \times n$ *selector matrix*: each row is a unit vector (transposed)

$$A = \begin{bmatrix} e_{k_1}^T \\ \vdots \\ e_{k_m}^T \end{bmatrix}$$

- multiplying by A selects entries of x:

$$Ax = (x_{k_1}, x_{k_2}, \ldots, x_{k_m})$$

- example: the $m \times 2m$ matrix

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

  'down-samples' by 2: if x is a $2m$-vector then $y = Ax = (x_1, x_3, \ldots, x_{2m-1})$

- other examples: image cropping, permutation, ...

Outline

Geometric transformations

Selectors

Incidence matrix

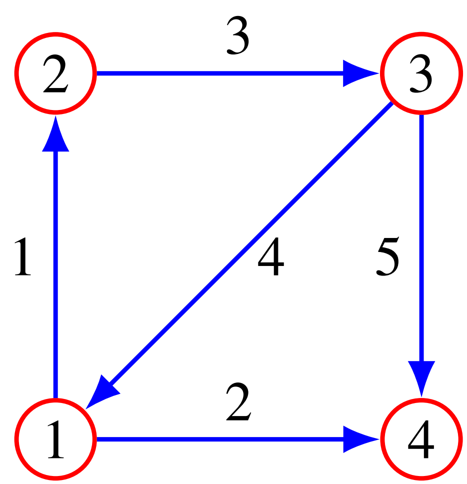
Convolution

Incidence matrix

- graph with n vertices or nodes, m (directed) edges or links
- incidence matrix is $n \times m$ matrix

$$A_{ij} = \begin{cases} 1 & \text{edge } j \text{ points to node } i \\ -1 & \text{edge } j \text{ points from node } i \\ 0 & \text{otherwise} \end{cases}$$

- example with $n = 4$, $m = 5$:



$$A = \begin{bmatrix} -1 & -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Flow conservation

- m-vector x gives flows (of something) along the edges
- examples: heat, money, power, mass, people, ...
- $x_j > 0$ means flow follows edge direction
- Ax is n-vector that gives the total or net flows
- $(Ax)_i$ is the net flow into node i
- $Ax = 0$ is *flow conservation*; x is called a *circulation*

Potentials and Dirichlet energy

- suppose v is an n-vector, called a *potential*
- v_i is potential value at node i
- $u = A^T v$ is an m-vector of *potential differences* across the m edges
- $u_j = v_l - v_k$, where edge j goes from k to node l
- *Dirichlet energy* is $\mathcal{D}(v) = \|A^T v\|^2$,

$$\mathcal{D}(v) = \sum_{\text{edges } (k,l)} (v_l - v_k)^2$$

(sum of squares of potential differences across the edges)

- $\mathcal{D}(v)$ is small when potential values of neighboring nodes are similar

8. Linear equations

Outline

Linear functions

Linear function models

Linear equations

Balancing chemical equations

Superposition

- $f : \mathbf{R}^n \to \mathbf{R}^m$ means f is a function that maps n-vectors to m-vectors
- we write $f(x) = (f_1(x), \ldots, f_m(x))$ to emphasize components of $f(x)$
- we write $f(x) = f(x_1, \ldots, x_n)$ to emphasize components of x
- f satisfies *superposition* if for all x, y, α, β

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

  (this innocent looking equation says a lot . . .)
- such an f is called *linear*

Matrix-vector product function

- with A an $m \times n$ matrix, define f as $f(x) = Ax$
- f is linear:

$$\begin{aligned} f(\alpha x + \beta y) &= A(\alpha x + \beta y) \\ &= A(\alpha x) + A(\beta y) \\ &= \alpha(Ax) + \beta(Ay) \\ &= \alpha f(x) + \beta f(y) \end{aligned}$$

- converse is true: if $f : \mathbf{R}^n \to \mathbf{R}^m$ is linear, then

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

with $A = \begin{bmatrix} f(e_1) & f(e_2) & \cdots & f(e_n) \end{bmatrix}$

Examples

- reversal: $f(x) = (x_n, x_{n-1}, \ldots, x_1)$

$$A = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & \iddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix}$$

- running sum: $f(x) = (x_1, x_1 + x_2, x_1 + x_2 + x_3, \ldots, x_1 + x_2 + \cdots + x_n)$

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

Affine functions

- function $f : \mathbf{R}^n \to \mathbf{R}^m$ is *affine* if it is a linear function plus a constant, *i.e.*,

$$f(x) = Ax + b$$

- same as:

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

  holds for all x, y, and α, β with $\alpha + \beta = 1$

- can recover A and b from f using

$$\begin{aligned} A &= \begin{bmatrix} f(e_1) - f(0) & f(e_2) - f(0) & \cdots & f(e_n) - f(0) \end{bmatrix} \\ b &= f(0) \end{aligned}$$

- affine functions sometimes (incorrectly) called linear

Outline

Linear and affine functions models

- in many applications, relations between n-vectors and m vectors are *approximated* as linear or affine
- sometimes the approximation is excellent, and holds over large ranges of the variables (*e.g.*, electromagnetics)
- sometimes the approximation is reasonably good over smaller ranges (*e.g.*, aircraft dynamics)
- in other cases it is quite approximate, but still useful (*e.g.*, econometric models)

Price elasticity of demand

- n goods or services
- prices given by n-vector p, demand given as n-vector d
- $\delta_i^{\text{price}} = (p_i^{\text{new}} - p_i)/p_i$ is fractional changes in prices
- $\delta_i^{\text{dem}} = (d_i^{\text{new}} - d_i)/d_i$ is fractional change in demands
- *price-demand elasticity model*: $\delta^{\text{dem}} = E\delta^{\text{price}}$

- what do the following mean?

$$E_{11} = -0.3, \qquad E_{12} = +0.1, \qquad E_{23} = -0.05$$

Taylor series approximation

- suppose $f : \mathbf{R}^n \to \mathbf{R}^m$ is differentiable
- first order Taylor approximation $\hat{f}$ of f near z:

$$\begin{aligned} \hat{f}_i(x) &= f_i(z) + \frac{\partial f_i}{\partial x_1}(z)(x_1 - z_1) + \cdots + \frac{\partial f_i}{\partial x_n}(z)(x_n - z_n) \\ &= f_i(z) + \nabla f_i(z)^T (x - z) \end{aligned}$$

- in compact notation: $\hat{f}(x) = f(z) + Df(z)(x - z)$
- $Df(z)$ is the $m \times n$ *derivative* or *Jacobian* matrix of f at z

$$Df(z)_{ij} = \frac{\partial f_i}{\partial x_j}(z), \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

- $\hat{f}(x)$ is a very good approximation of $f(x)$ for x near z
- $\hat{f}(x)$ is an affine function of x

Regression model

- regression model: $\hat{y} = x^T \beta + v$
  - x is n-vector of features or regressors
  - β is n-vector of model parameters; v is offset parameter
  - (scalar) $\hat{y}$ is our prediction of y
- now suppose we have N *examples* or *samples* $x^{(1)}, \ldots, x^{(N)}$, and associated responses $y^{(1)}, \ldots, y^{(N)}$
- associated predictions are $\hat{y}^{(i)} = (x^{(i)})^T \beta + v$
- write as $\hat{y}^{\mathrm{d}} = X^T \beta + v\mathbf{1}$
  - X is feature matrix with columns $x^{(1)}, \ldots, x^{(N)}$
  - y^{d} is N-vector of responses $(y^{(1)}, \ldots, y^{(N)})$
  - $\hat{y}^{\mathrm{d}}$ is N-vector of predictions $(\hat{y}^{(1)}, \ldots, \hat{y}^{(N)})$
- *prediction error* (vector) is $y^{\mathrm{d}} - \hat{y}^{\mathrm{d}} = y^{\mathrm{d}} - X^T \beta - v\mathbf{1}$

Outline

Linear functions

Linear function models

Linear equations

Balancing chemical equations

Systems of linear equations

- set (or *system*) of m linear equations in n variables $x_1, \ldots, x_n$:

$$\begin{aligned} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n &= b_1 \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n &= b_2 \\ &\vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n &= b_m \end{aligned}$$

- n-vector x is called the variable or unknowns
- A_{ij} are the *coefficients*; A is the coefficient matrix
- b is called the *right-hand side*
- can express very compactly as $Ax = b$

Systems of linear equations

- systems of linear equations classified as
  - under-determined if $m < n$ (A wide)
  - square if $m = n$ (A square)
  - over-determined if $m > n$ (A tall)
- x is called a *solution* if $Ax = b$
- depending on A and b, there can be
  - no solution
  - one solution
  - many solutions
- we'll see how to solve linear equations later

Outline

Linear functions

Linear function models

Linear equations

Balancing chemical equations

Chemical equations

- a chemical reaction involves p reactants, q products (molecules)
- expressed as

$$a_1 R_1 + \cdots + a_p R_p \longrightarrow b_1 P_1 + \cdots + b_q P_q$$

- $R_1, \ldots, R_p$ are reactants
- $P_1, \ldots, P_q$ are products
- $a_1, \ldots, a_p, b_1, \ldots, b_q$ are positive coefficients
- coefficients usually integers, but can be scaled
  - *e.g.*, multiplying all coefficients by $1/2$ doesn't change the reaction

Example: electrolysis of water

$$2H_2O \longrightarrow 2H_2 + O_2$$

- one reactant: water (H_2O)
- two products: hydrogen (H_2) and oxygen (O_2)
- reaction consumes 2 water molecules and produces 2 hydrogen molecules and 1 oxygen molecule

Balancing equations

- each molecule (reactant/product) contains specific numbers of (types of) atoms, given in its formula
  - *e.g.*, $\mathrm{H_2O}$ contains two H and one O
- *conservation of mass*: total number of each type of atom in a chemical equation must *balance*
- for each atom, total number on LHS must equal total on RHS
- *e.g.*, electrolysis reaction is balanced:
  - 4 units of H on LHS and RHS
  - 2 units of O on LHS and RHS
- finding (nonzero) coefficients to achieve balance is called *balancing* equations

Reactant and product matrices

- consider reaction with m types of atoms, p reactants, q products
- $m \times p$ *reactant matrix* R is defined by

$$R_{ij} = \text{number of atoms of type } i \text{ in reactant } R_j,$$

  for $i = 1, \ldots, m$ and $j = 1, \ldots, p$
- with $a = (a_1, \ldots, a_p)$ (vector of reactant coefficients)

$$Ra = \text{(vector of) total numbers of atoms of each type in reactants}$$

- define product $m \times q$ matrix P in similar way
- m-vector Pb is total numbers of atoms of each type in products
- conservation of mass is $Ra = Pb$

Balancing equations via linear equations

- conservation of mass is

$$\begin{bmatrix} R & -P \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = 0$$

- simple solution is $a = b = 0$
- to find a nonzero solution, set any coefficient (say, a_1) to be 1
- balancing chemical equations can be expressed as solving a set of $m + 1$ linear equations in $p + q$ variables

$$\begin{bmatrix} R & -P \\ e_1^T & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = e_{m+1}$$

(we ignore here that a_i and b_i should be nonnegative integers)

Conservation of charge

- can extend to include charge, *e.g.*, $\mathrm{Cr_2O_7^{2-}}$ has charge -2
- *conservation of charge*: total charge on each side of reaction must balance
- we can simply treat charge as another type of atom to balance

Example

$$a_1 \mathrm{Cr_2O_7^{2-}} + a_2 \mathrm{Fe^{2+}} + a_3 \mathrm{H^+} \longrightarrow b_1 \mathrm{Cr^{3+}} + b_2 \mathrm{Fe^{3+}} + b_3 \mathrm{H_2O}$$

- 5 atoms/charge: Cr, O, Fe, H, charge
- reactant and product matrix:

$$R = \begin{bmatrix} 2 & 0 & 0 \\ 7 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & 2 & 1 \end{bmatrix}, \qquad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \\ 3 & 3 & 0 \end{bmatrix}$$

- balancing equations (including $a_1 = 1$ constraint)

$$\begin{bmatrix} 2 & 0 & 0 & -1 & 0 & 0 \\ 7 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 \\ -2 & 2 & 1 & -3 & -3 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Balancing equations example

- solving the system yields

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ 14 \\ 2 \\ 6 \\ 7 \end{bmatrix}$$

- the balanced equation is

$$\mathrm{Cr_2O_7^{2-}} + 6\mathrm{Fe}^{2+} + 14\mathrm{H}^+ \longrightarrow 2\mathrm{Cr}^{3+} + 6\mathrm{Fe}^{3+} + 7\mathrm{H_2O}$$

10. Matrix multiplication

Outline

Matrix multiplication

Composition of linear functions

Matrix powers

QR factorization

Matrix multiplication

- can multiply $m \times p$ matrix A and $p \times n$ matrix B to get $C = AB$:

$$C_{ij} = \sum_{k=1}^{p} A_{ik}B_{kj} = A_{i1}B_{1j} + \cdots + A_{ip}B_{pj}$$

  for $i = 1, \ldots, m$, $j = 1, \ldots, n$

- to get C_{ij}: move along ith row of A, jth column of B
- example:

$$\begin{bmatrix} -1.5 & 3 & 2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 \\ 0 & -2 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 3.5 & -4.5 \\ -1 & 1 \end{bmatrix}$$

Special cases of matrix multiplication

- scalar-vector product (with scalar on right!) $x\alpha$
- inner product $a^T b$
- matrix-vector multiplication Ax
- *outer product* of m-vector a and n-vector b

$$ab^T = \begin{bmatrix} a_1b_1 & a_1b_2 & \cdots & a_1b_n \\ a_2b_1 & a_2b_2 & \cdots & a_2b_n \\ \vdots & \vdots & & \vdots \\ a_mb_1 & a_mb_2 & \cdots & a_mb_n \end{bmatrix}$$

Properties

- $(AB)C = A(BC)$, so both can be written ABC
- $A(B + C) = AB + AC$
- $(AB)^T = B^T A^T$
- $AI = A$ and $IA = A$
- $AB = BA$ *does not hold in general*

Block matrices

block matrices can be multiplied using the same formula, *e.g.*,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

(provided the products all make sense)

Column interpretation

- denote columns of B by b_i:

$$B = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}$$

- then we have

$$\begin{aligned} AB &= A \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix} \\ &= \begin{bmatrix} Ab_1 & Ab_2 & \cdots & Ab_n \end{bmatrix} \end{aligned}$$

- so AB is 'batch' multiply of A times columns of B

Multiple sets of linear equations

- given k systems of linear equations, with same $m \times n$ coefficient matrix

$$Ax_i = b_i, \quad i = 1, \ldots, k$$

- write in compact matrix form as $AX = B$
- $X = [x_1 \ \cdots \ x_k]$, $B = [b_1 \ \cdots \ b_k]$

Inner product interpretation

- with a_i^T the rows of A, b_j the columns of B, we have

$$AB = \begin{bmatrix} a_1^T b_1 & a_1^T b_2 & \cdots & a_1^T b_n \\ a_2^T b_1 & a_2^T b_2 & \cdots & a_2^T b_n \\ \vdots & \vdots & & \vdots \\ a_m^T b_1 & a_m^T b_2 & \cdots & a_m^T b_n \end{bmatrix}$$

- so matrix product is all inner products of rows of A and columns of B, arranged in a matrix

Gram matrix

- let A be an $m \times n$ matrix with columns $a_1, \ldots, a_n$
- the *Gram matrix* of A is

$$G = A^T A = \begin{bmatrix} a_1^T a_1 & a_1^T a_2 & \cdots & a_1^T a_n \\ a_2^T a_1 & a_2^T a_2 & \cdots & a_2^T a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n^T a_1 & a_n^T a_2 & \cdots & a_n^T a_n \end{bmatrix}$$

- Gram matrix gives all inner products of columns of A
- example: $G = A^T A = I$ means columns of A are orthonormal

Complexity

- to compute $C_{ij} = (AB)_{ij}$ is inner product of p-vectors
- so total required flops is $(mn)(2p) = 2mnp$ flops
- multiplying two 1000×1000 matrices requires 2 billion flops
- ...and can be done in well under a second on current computers

11. Matrix inverses

Outline

Left and right inverses

Inverse

Solving linear equations

Examples

Pseudo-inverse

Left inverses

- a number x that satisfies $xa = 1$ is called the inverse of a
- inverse (*i.e.*, $1/a$) exists if and only if $a \neq 0$, and is unique
- a matrix X that satisfies $XA = I$ is called a *left inverse* of A
- if a left inverse exists we say that A is *left-invertible*
- example: the matrix
$$A = \begin{bmatrix} -3 & -4 \\ 4 & 6 \\ 1 & 1 \end{bmatrix}$$
has two different left inverses:
$$B = \frac{1}{9}\begin{bmatrix} -11 & -10 & 16 \\ 7 & 8 & -11 \end{bmatrix}, \qquad C = \frac{1}{2}\begin{bmatrix} 0 & -1 & 6 \\ 0 & 1 & -4 \end{bmatrix}$$

Left inverse and column independence

- if A has a left inverse C then the columns of A are linearly independent
- to see this: if $Ax = 0$ and $CA = I$ then

$$0 = C0 = C(Ax) = (CA)x = Ix = x$$

- we'll see later the converse is also true, so

  a matrix is left-invertible if and only if its columns are linearly independent
- matrix generalization of

  a number is invertible if and only if it is nonzero
- so left-invertible matrices are tall or square

Solving linear equations with a left inverse

- suppose $Ax = b$, and A has a left inverse C
- then $Cb = C(Ax) = (CA)x = Ix = x$
- so multiplying the right-hand side by a left inverse yields the solution

Example

$$A = \begin{bmatrix} -3 & -4 \\ 4 & 6 \\ 1 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}$$

- over-determined equations $Ax = b$ have (unique) solution $x = (1, -1)$
- A has two different left inverses,

$$B = \frac{1}{9}\begin{bmatrix} -11 & -10 & 16 \\ 7 & 8 & -11 \end{bmatrix}, \qquad C = \frac{1}{2}\begin{bmatrix} 0 & -1 & 6 \\ 0 & 1 & -4 \end{bmatrix}$$

- multiplying the right-hand side with the left inverse B we get

$$Bb = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

- and also

$$Cb = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Right inverses

- a matrix X that satisfies $AX = I$ is a *right inverse* of A
- if a right inverse exists we say that A is *right-invertible*
- A is right-invertible if and only if A^T is left-invertible:

$$AX = I \iff (AX)^T = I \iff X^T A^T = I$$

- so we conclude

  A is right-invertible if and only if its rows are linearly independent

- right-invertible matrices are wide or square

Solving linear equations with a right inverse

- suppose A has a right inverse B
- consider the (square or underdetermined) equations $Ax = b$
- $x = Bb$ is a solution:

$$Ax = A(Bb) = (AB)b = Ib = b$$

- so $Ax = b$ has a solution for *any* b

Example

- same A, B, C in example above
- C^T and B^T are both right inverses of A^T
- under-determined equations $A^T x = (1, 2)$ has (different) solutions

$$B^T(1,2) = (1/3, 2/3, -2/3), \qquad C^T(1,2) = (0, 1/2, -1)$$

(there are many other solutions as well)

Outline

Left and right inverses

Inverse

Solving linear equations

Examples

Pseudo-inverse

Inverse

- if A has a left and a right inverse, they are unique and equal (and we say that A is *invertible*)
- so A must be square
- to see this: if $AX = I$, $YA = I$

$$X = IX = (YA)X = Y(AX) = YI = Y$$

- we denote them by A^{-1}:

$$A^{-1}A = AA^{-1} = I$$

- inverse of inverse: $(A^{-1})^{-1} = A$

Solving square systems of linear equations

- suppose A is invertible
- for any b, $Ax = b$ has the unique solution

$$x = A^{-1}b$$

- matrix generalization of simple scalar equation $ax = b$ having solution $x = (1/a)b$ (for $a \neq 0$)
- simple-looking formula $x = A^{-1}b$ is basis for many applications

Invertible matrices

the following are equivalent for a square matrix A:

- A is invertible
- columns of A are linearly independent
- rows of A are linearly independent
- A has a left inverse
- A has a right inverse

if any of these hold, all others do

Examples

- $I^{-1} = I$
- if Q is orthogonal, *i.e.*, square with $Q^TQ = I$, then $Q^{-1} = Q^T$
- 2×2 matrix A is invertible if and only $A_{11}A_{22} \neq A_{12}A_{21}$

$$A^{-1} = \frac{1}{A_{11}A_{22} - A_{12}A_{21}} \begin{bmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{bmatrix}$$

  - you need to know this formula
  - there are similar but *much* more complicated formulas for larger matrices (and no, you do not need to know them)

Non-obvious example

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 0 & 2 & 2 \\ -3 & -4 & -4 \end{bmatrix}$$

- A is invertible, with inverse

$$A^{-1} = \frac{1}{30} \begin{bmatrix} 0 & -20 & -10 \\ -6 & 5 & -2 \\ 6 & 10 & 2 \end{bmatrix}.$$

- verified by checking $AA^{-1} = I$ (or $A^{-1}A = I$)
- we'll soon see how to compute the inverse

Properties

- $(AB)^{-1} = B^{-1}A^{-1}$ (provided inverses exist)
- $(A^T)^{-1} = (A^{-1})^T$ (sometimes denoted A^{-T})
- negative matrix powers: $(A^{-1})^k$ is denoted A^{-k}
- with $A^0 = I$, identity $A^kA^l = A^{k+l}$ holds for any integers k, l

Outline

Left and right inverses

Inverse

Solving linear equations

Examples

Pseudo-inverse

Invertibility of Gram matrix

- A has linearly independent columns if and only if A^TA is invertible
- to see this, we'll show that $Ax = 0 \Leftrightarrow A^TAx = 0$
- $\Rightarrow$: if $Ax = 0$ then $(A^TA)x = A^T(Ax) = A^T0 = 0$
- $\Leftarrow$: if $(A^TA)x = 0$ then

$$0 = x^T(A^TA)x = (Ax)^T(Ax) = \|Ax\|^2 = 0$$

  so $Ax = 0$

Pseudo-inverse of tall matrix

- the *pseudo-inverse* of A with independent columns is

$$A^\dagger = (A^TA)^{-1}A^T$$

- it is a left inverse of A:

$$A^\dagger A = (A^TA)^{-1}A^TA = (A^TA)^{-1}(A^TA) = I$$

(we'll soon see that it's a very important left inverse of A)

- reduces to A^{-1} when A is square:

$$A^\dagger = (A^TA)^{-1}A^T = A^{-1}A^{-T}A^T = A^{-1}I = A^{-1}$$

Pseudo-inverse of wide matrix

- if A is wide, with linearly independent rows, AA^T is invertible
- pseudo-inverse is defined as

$$A^\dagger = A^T(AA^T)^{-1}$$

- $A^\dagger$ is a right inverse of A:

$$AA^\dagger = AA^T(AA^T)^{-1} = I$$

  (we'll see later it is an important right inverse)

- reduces to A^{-1} when A is square:

$$A^T(AA^T)^{-1} = A^T A^{-T} A^{-1} = A^{-1}$$

Pseudo-inverse via QR factorization

- suppose A has linearly independent columns, $A = QR$
- then $A^T A = (QR)^T (QR) = R^T Q^T QR = R^T R$
- so

$$A^\dagger = (A^T A)^{-1} A^T = (R^T R)^{-1} (QR)^T = R^{-1} R^{-T} R^T Q^T = R^{-1} Q^T$$

- can compute $A^\dagger$ using back substitution on columns of Q^T
- for A with linearly independent rows, $A^\dagger = QR^{-T}$