Vectors

Stephen Boyd
EE103
Stanford University

September 20, 2017
Outline

Notation

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 a vector are called entries, coefficients, or elements.
- Length of vector is its size, length, or dimension.
- Vector above has dimension 4; its third entry is 3.6.
- Vector of length \(n\) is called an \(n\)-vector.
- Numbers are called scalars.
Vectors via symbols

- we’ll use symbols to denote vectors, e.g., $a$, $X$, $p$, $\beta$, $E^\text{aut}$
- other conventions: $g$, $\vec{a}$
- $i$th 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 $i$th 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 \textit{stacked vector} or \textit{concatenation} (of $b$, $c$, and $d$) is

\[
a = \begin{bmatrix} b \\ c \\ d \end{bmatrix}
\]

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

\[
a = (b_1, b_2, \ldots, b_m, c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_p)
\]
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: \((R, G, B)\)
- quantities of \(n\) different commodities (or resources), e.g., a 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 \(i\)th feature or attribute of an entity
- word count: \(x_i\) is the number of times word \(i\) appears in a document
Zero, ones, and unit vectors

- An $n$-vector with all entries 0 is denoted $0_n$ or just 0. 
- An $n$-vector with all entries 1 is denoted $1_n$ or just 1. 
- A unit vector has one entry 1 and all others 0. 
- Denoted $e_i$ where $i$ is the entry that is 1. 
- Unit vectors of length 3:

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

- a vector is *sparse* if many of its entries are 0
- can be stored and manipulated efficiently on a computer
- $\text{nnz}(x)$ is number of entries that are nonzero
- examples: zero vectors, unit vectors
Outline

Notation

Addition and scalar multiplication

Inner product

Complexity
Vector addition

- *n*-vectors $a$ and $b$ and 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 $\alpha$ and $n$-vector $a$ can be multiplied
  \[ \alpha a = (\alpha a_1, \ldots, \alpha a_n) \]

- also denoted $a\alpha$

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

- examples:
  - audio mixing
  - replicating a cash flow

- a simple identity: for any $n$-vector $b$,

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

Notation

Addition and scalar multiplication

Inner product

Complexity
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: $\langle a, b \rangle$, $\langle a | b \rangle$, $(a, b)$, $a \cdot b$

- properties:
  - $a^T b = b^T a$
  - $(\gamma a)^T b = \gamma(a^T b)$
  - $(a + b)^T c = a^T c + b^T c$
Simple examples

- $e_i^T a = a_i$ (picks out $i$th entry)
- $1^T a = a_1 + \cdots + a_n$ (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, $d = (1, 1/(1+r), \ldots, 1/(1+r)^{n-1})$ is discount vector (with interest rate $r$); $d^T c$ 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

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: computer speed/flops
- 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