[3] The Vector Space

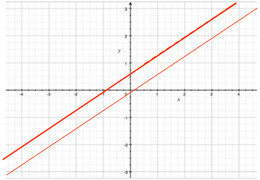
Affine space

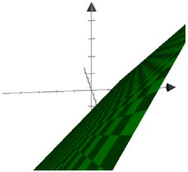
Definition: If $\mathbf{c}$ is a vector and V is a vector space then

$$\mathbf{c} + \mathrm{V}$$

is called an *affine space*.

Examples: A plane or a line not necessarily containing the origin.

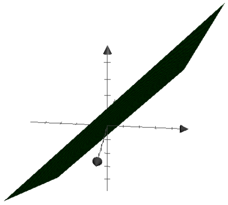




Affine space and affine combination

Example: The plane containing $\mathbf{u}_1 = [3, 0, 0]$, $\mathbf{u}_2 = [-3, 1, -1]$, and $\mathbf{u}_3 = [1, -1, 1]$.

Want to express this plane as $\mathbf{u}_1 + \mathcal{V}$ where $\mathcal{V}$ is the span of two vectors (a plane containing the origin)

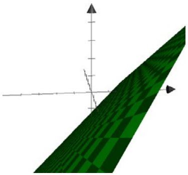


Let $\mathcal{V} = \text{Span}\ \{\mathbf{a}, \mathbf{b}\}$ where

$\mathbf{a} = \mathbf{u}_2 - \mathbf{u}_1$ and $\mathbf{b} = \mathbf{u}_3 - \mathbf{u}_1$

Since $\mathbf{u}_1 + \mathcal{V}$ is a translation of a plane, it is also a plane.

- Span $\{\mathbf{a}, \mathbf{b}\}$ contains $\mathbf{0}$, so $\mathbf{u}_1 +$ Span $\{\mathbf{a}, \mathbf{b}\}$ contains $\mathbf{u}_1$.
- Span $\{\mathbf{a}, \mathbf{b}\}$ contains $\mathbf{u}_2 - \mathbf{u}_1$ so $\mathbf{u}_1 +$ Span $\{\mathbf{a}, \mathbf{b}\}$ contains $\mathbf{u}_2$.
- Span $\{\mathbf{a}, \mathbf{b}\}$ contains $\mathbf{u}_3 - \mathbf{u}_1$ so $\mathbf{u}_1 +$ Span $\{\mathbf{a}, \mathbf{b}\}$ contains $\mathbf{u}_3$.



Thus the plane $\mathbf{u}_1 +$ Span $\{\mathbf{a}, \mathbf{b}\}$ contains $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$.

Only one plane contains those three points, so this is that one.

Affine space and affine combination

Example: The plane containing $\mathbf{u}_1 = [3, 0, 0]$, $\mathbf{u}_2 = [-3, 1, -1]$, and $\mathbf{u}_1 = [1, -1, 1]$:

$$\mathbf{u}_1 + \text{Span}\ \{\mathbf{u}_2 - \mathbf{u}_1, \mathbf{u}_3 - \mathbf{u}_1\}$$

Cleaner way to write it?

$$\begin{aligned}
\mathbf{u}_1 + \text{Span}\ \{\mathbf{u}_2 - \mathbf{u}_1, \mathbf{u}_3 - \mathbf{u}_1\} &= \{\mathbf{u}_1 + \alpha\,(\mathbf{u}_2 - \mathbf{u}_1) + \beta\,(\mathbf{u}_3 - \mathbf{u}_1) : \alpha, \beta \in \mathbb{R}\} \\
&= \{\mathbf{u}_1 + \alpha\,\mathbf{u}_2 - \alpha\,\mathbf{u}_1 + \beta\,\mathbf{u}_3 - \beta\,\mathbf{u}_1 : \alpha, \beta \in \mathbb{R}\} \\
&= \{(1 - \alpha - \beta)\,\mathbf{u}_1 + \alpha\,\mathbf{u}_2 + \beta\,\mathbf{u}_3 : \alpha, \beta \in \mathbb{R}\} \\
&= \{\gamma\,\mathbf{u}_1 + \alpha\,\mathbf{u}_2 + \beta\,\mathbf{u}_3 : \gamma + \alpha + \beta = 1\}
\end{aligned}$$

Definition: A linear combination $\gamma\,\mathbf{u}_1 + \alpha\,\mathbf{u}_2 + \beta\,\mathbf{u}_3$ where $\gamma + \alpha + \beta = 1$ is an *affine combination*.

Affine combination

Definition: A linear combination

$$a_1\mathbf{u}_1 + a_2\mathbf{u}_2 + \cdots + a_n\mathbf{u}_n$$

where

$$a_1 + a_2 + \cdots + a_n = 1$$

is an *affine combination*.

Definition: The set of all affine combinations of vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ is called the *affine hull* of those vectors.

$$\text{Affine hull of } \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n = \mathbf{u}_1 + \text{Span}\{\mathbf{u}_2 - \mathbf{u}_1, \ldots, \mathbf{u}_n - \mathbf{u}_1\}$$

This shows that the affine hull of some vectors is an affine space..

Geometric objects not containing the origin: equations

Can express a plane as $\mathbf{u}_1 + \mathcal{V}$ or affine hull of $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$.

More familiar way to express a plane:

The solution set of an equation $ax + by + cz = d$

In vector terms,

$$\{[x, y, z] : [a, b, c] \cdot [x, y, z] = d\}$$

In general, a geometric object (point, line, plane, ...) can be expressed as the solution set of a system of linear equations.

$$\{\mathbf{x} : \mathbf{a}_1 \cdot \mathbf{x} = \beta_1, \ldots, \mathbf{a}_m \cdot \mathbf{x} = \beta_m\}$$

Conversely, is the solution set an affine space?

Consider solution set of a contradictory system of equations, e.g. $1\,x = 1, 2\,x = 1$:

- Solution set is empty....
- ...but a vector space $\mathcal{V}$ always contains the zero vector,
- ...so an affine space $\mathbf{u}_1 + \mathcal{V}$ always contains at least one vector.

Turns out this the only exception:

Theorem: The solution set of a linear system is either empty or an affine space.

Affine spaces and linear systems

Theorem: The solution set of a linear system is either empty or an affine space.

Each linear system corresponds to a linear system with zero right-hand sides:

$$\begin{aligned} \mathbf{a}_1 \cdot \mathbf{x} &= \beta_1 \\ &\vdots \\ \mathbf{a}_m \cdot \mathbf{x} &= \beta_m \end{aligned} \quad \Longrightarrow \quad \begin{aligned} \mathbf{a}_1 \cdot \mathbf{x} &= 0 \\ &\vdots \\ \mathbf{a}_m \cdot \mathbf{x} &= 0 \end{aligned}$$

Definition:

A linear equation $\mathbf{a} \cdot \mathbf{x} = 0$ with zero right-hand side is a *homogeneous* linear equation.

A system of homogeneous linear equations is called a *homogeneous* linear system.

We already know: The solution set of a homogeneous linear system is a vector space.

Lemma: Let $\mathbf{u}_1$ be a solution to a linear system. Then, for any other vector $\mathbf{u}_2$, $\mathbf{u}_2$ is also a solution

if and only if

$\mathbf{u}_2 - \mathbf{u}_1$ is a solution to the corresponding homogeneous linear system.

Affine spaces and linear systems

$$\begin{array}{c} \mathbf{a}_1 \cdot \mathbf{x} = \beta_1 \\ \vdots \\ \mathbf{a}_m \cdot \mathbf{x} = \beta_m \end{array} \quad \Longrightarrow \quad \begin{array}{c} \mathbf{a}_1 \cdot \mathbf{x} = 0 \\ \vdots \\ \mathbf{a}_m \cdot \mathbf{x} = 0 \end{array}$$

Lemma: Let $\mathbf{u}_1$ be a solution to a linear system. Then, for any other vector $\mathbf{u}_2$, $\mathbf{u}_2$ is also a solution if and only if $\mathbf{u}_2 - \mathbf{u}_1$ is a solution to the corresponding homogeneous linear system.

Proof: We assume $\mathbf{a}_1 \cdot \mathbf{u}_1 = \beta_1, \ldots, \mathbf{a}_m \cdot \mathbf{u}_1 = \beta_m$, so

$$\begin{array}{rcl} \mathbf{a}_1 \cdot \mathbf{u}_2 & = & \beta_1 \\ & \vdots & \\ \mathbf{a}_m \cdot \mathbf{u}_2 & = & \beta_m \end{array} \quad \text{iff} \quad \begin{array}{rcl} \mathbf{a}_1 \cdot \mathbf{u}_2 - \mathbf{a}_1 \cdot \mathbf{u}_1 & = & 0 \\ & \vdots & \\ \mathbf{a}_m \cdot \mathbf{u}_2 - \mathbf{a}_m \cdot \mathbf{u}_1 & = & 0 \end{array} \quad \text{iff} \quad \begin{array}{rcl} \mathbf{a}_1 \cdot (\mathbf{u}_2 - \mathbf{u}_1) & = & 0 \\ & \vdots & \\ \mathbf{a}_m \cdot (\mathbf{u}_2 - \mathbf{u}_2) & = & 0 \end{array}$$

QED

Lemma: Let $\mathbf{u}_1$ be a solution to a linear system. Then, for any other vector $\mathbf{u}_2$, $\mathbf{u}_2$ is also a solution if and only if $\mathbf{u}_2 - \mathbf{u}_1$ is a solution to the corresponding homogeneous linear system.

We use this lemma to prove the theorem:

Theorem: The solution set of a linear system is either empty or an affine space.

- Let $\mathcal{V}$ = set of solutions to corresponding homogeneous linear system.
- If the linear system has no solution, its solution set is empty.
- If it does has a solution $\mathbf{u}_1$ then

$$\{\text{solutions to linear system}\} = \{\mathbf{u}_2 : \mathbf{u}_2 - \mathbf{u}_1 \in \mathcal{V}\}$$

(substitute $\mathbf{v} = \mathbf{u}_2 - \mathbf{u}_1$)

$$= \{\mathbf{u}_1 + \mathbf{v} : \mathbf{v} \in \mathcal{V}\}$$

QED

Number of solutions to a linear system

We just proved:

If $\mathbf{u}_1$ is a solution to a linear system then $$\{\text{solutions to linear system}\} = \{\mathbf{u}_1 + \mathbf{v} : \mathbf{v} \in \mathcal{V}\}$$ where $\mathcal{V} = \{\text{solutions to corresponding homogeneous linear system}\}$

Implications:

Long ago we asked: *How can we tell if a linear system has only one solution?*

Now we know: If a linear system has a solution $\mathbf{u}_1$ then that solution is unique if the only solution to the corresponding homogeneous linear system is $\mathbf{0}$.

Long ago we asked: How can we find the number of solutions to a linear system over GF (2)?

Now we know: Number of solutions either is zero or is equal to the number of solutions to the corresponding *homogeneous* linear system.

Number of solutions: checksum function

MD5 checksums and sizes of the released files:

```
3c63a6d97333f4da35976b6a0755eb67  12732276  Python-3.2.2.tgz
9d763097a13a59ff53428c9e4d098a05  10743647  Python-3.2.2.tar.bz2
3720ce9460597e49264bbb63b48b946d   8923224  Python-3.2.2.tar.xz
f6001a9b2be57ecfbefa865e50698cdf  19519332  python-3.2.2-macosx10.3.dmg
8fe82d14dbb2e96a84fd6fa1985b6f73  16226426  python-3.2.2-macosx10.6.dmg
cccb03e14146f7ef82907cf12bf5883c  18241506  python-3.2.2-pdb.zip
72d11475c986182bcb0e5c91acec45bc  19940424  python-3.2.2.amd64-pdb.zip
ddeb3e3fb93ab5a900adb6f04edab21e  18542592  python-3.2.2.amd64.msi
8afb1b01e8fab738e7b234eb4fe3955c  18034688  python-3.2.2.msi
```

A *checksum function* maps long files to short sequences.

Idea:

- Web page shows the checksum of each file to be downloaded.
- Download the file and run the checksum function on it.
- If result does not match checksum on web page, you know the file has been corrupted.
- If random corruption occurs, how likely are you to detect it?

Impractical but instructive checksum function:

- *input*: an n-vector $\mathbf{x}$ over $GF(2)$
- *output*: $[\mathbf{a}_1 \cdot \mathbf{x}, \mathbf{a}_2 \cdot \mathbf{x}, \ldots, \mathbf{a}_{64} \cdot \mathbf{x}]$

where $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{64}$ are sixty-four n-vectors.

Geometry of sets of vectors: convex hull

Earlier, we saw: The **u**-to-**v** line segment is

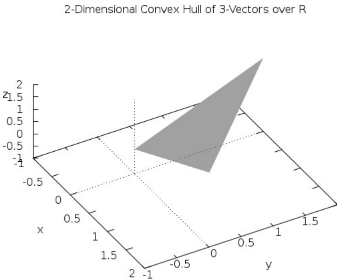
$$\{\alpha\,\mathbf{u} + \beta\,\mathbf{v} : \alpha \in \mathbb{R}, \beta \in \mathbb{R}, \alpha \geq 0, \beta \geq 0, \alpha + \beta = 1\}$$

Definition: For vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ over R, a linear combination

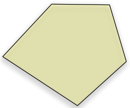
$$\alpha_1\,\mathbf{v}_1 + \cdots + \alpha_n\,\mathbf{v}_n$$

is a *convex combination* if the coefficients are all nonnegative and they sum to 1.

- Convex hull of a single vector is a point.
- Convex hull of two vectors is a line segment.
- Convex hull of three vectors is a triangle



Convex hull of more vectors? Could be higher-dimensional... but not necessarily.



For example, a convex polygon is the convex hull of its vertices

Subspace

Definition: A set w of vectors is called a subspace of vector space V if:

i. $w \leq v$

ii. w is a vector space

Example: Show that $W = \alpha[2,1] | \alpha \in \mathbb{R}$ is a subspace of $\mathbb{R}^2$

i. $\vec{w} \in w \rightarrow w \in \mathbb{R}^2$ i.e. $\vec{w} = [2\alpha, \alpha] \in \mathbb{R}^2$ because $\alpha, 2\alpha \in \mathbb{R}$

ii. $\vec{0} = [0,0] \in w$ (let $\alpha = 0$): $\forall \vec{w} \in w, \vec{w} + \vec{0} = \vec{w}$ and $\vec{w} - \vec{w} = \vec{0}$

- $\forall \beta \in \mathbb{R}$ and $\forall \vec{w} \in w, \beta\vec{w} = \beta(\alpha[2,1]) = \beta\alpha[2,1]$
- $\forall \vec{w}_1, \vec{w}_2 \in w, \vec{w}_1 + \vec{w}_2 = \alpha_1[2,1] + \alpha_2[2,1] = (\alpha_1 + \alpha_2)[2,1]$

[4] The Matrix

What is a matrix? Traditional answer

Traditional notion of a matrix: two-dimensional array.

$$\begin{bmatrix} 1 & 2 & 3 \\ 10 & 20 & 30 \end{bmatrix}$$

- Two rows: [1, 2, 3] and [10, 20, 30].
- Three columns: [1, 10], [2, 20], and [3, 30].
- A 2 × 3 matrix.

For a matrix A, the i, j element of A is the element in row i, column j

- is traditionally written $A_{i,j}$ but we will use $A[i, j]$
- Definition: A matrix $A_{\mathrm{m}\times n}$ over a field $\mathbb{F}$ is a dimensional array with m rows and n columns where $A_{i,j} \in \mathbb{F}$

List of row-lists, list of column-lists (Quiz)

- One obvious Python representation for a matrix: a list of row-lists:
  $\begin{bmatrix} 1 & 2 & 3 \\ 10 & 20 & 30 \end{bmatrix}$ represented by [[1,2,3],[10,20,30]].

- Another: a list of column-lists:
  $\begin{bmatrix} 1 & 2 & 3 \\ 10 & 20 & 30 \end{bmatrix}$ represented by [[1,10],[2,20],[3,30]].

Identity matrix

	a	b	c
a	1	0	0
b	0	1	0
c	0	0	1

Definition: $D \times D$ *identity matrix* is the matrix 1_D such that $1_D[k, k] = 1$ for all $k \in D$ and zero elsewhere.

Usually we omit the subscript when D is clear from the context.
Often letter I (for "identity") is used instead of 1

Column space and row space

One simple role for a matrix: packing together a bunch of columns or rows

Two vector spaces associated with a matrix M:

Definition:

- *column space* of M = Span {columns of M}
  Written Col M

- *row space* of M = Span {rows of M}
  Written Row M

Examples:

- Column space of $\begin{array}{ccc} 1 & 2 & 3 \\ 10 & 20 & 30 \end{array}$ is Span {[1, 10], [2, 20], [3, 30]}.
  In this case, the span is equal to Span {[1, 10]} since [2, 20] and [3, 30] are scalar multiples of [1, 10].

- The row space of the same matrix is Span {[1, 2, 3], [10, 20, 30]}.
  In this case, the span is equal to Span {[1, 2, 3]} since [10, 20, 30] is a scalar multiple of [1, 2, 3].

Transpose

Transpose swaps rows and columns.

	@	#	?
a	2	1	3
b	20	10	30

→

	a	b
@	2	20
#	1	10
?	3	30

Matrix-vector and vector-matrix multiplication

Two ways to multiply a matrix by a vector:

- matrix-vector multiplication
- vector-matrix multiplication

For each of these, two *equivalent definitions*:

- in terms of linear combinations
- in terms of dot-products

Matrix-vector multiplication in terms of linear combinations

Linear-Combinations Definition of matrix-vector multiplication: Let M be an $R \times C$ matrix.

- If $\mathbf{v}$ is a C-vector then

$$M * \mathbf{v} = \sum_{c \in C} \mathbf{v}[c] \text{ (column } c \text{ of } M)$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 10 & 20 & 30 \end{bmatrix} \cdot [7 \quad 0 \quad 4] = 7[1 \quad 10] + 0[2 \quad 20] + 4[3 \quad 30] = [19 \quad 190]$$

- If $\mathbf{v}$ is *not* a C-vector then

$$M * \mathbf{v} = \text{ERROR!}$$

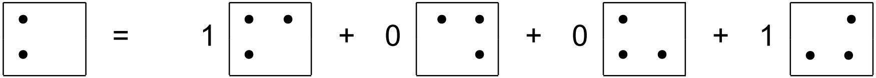
$$\begin{bmatrix} 1 & 2 & 3 \\ 10 & 20 & 30 \end{bmatrix} \cdot [7 \quad 0] = \text{ERROR!}$$

*

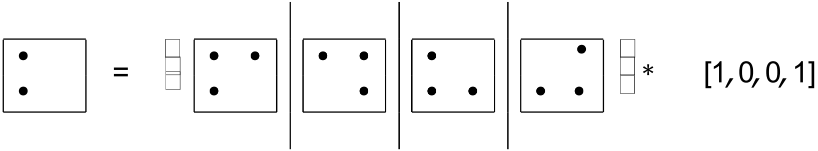
Matrix-vector multiplication in terms of linear combinations: *Lights Out*

A solution to a *Lights Out* configuration is a linear combination of "button vectors."

For example, the linear combination

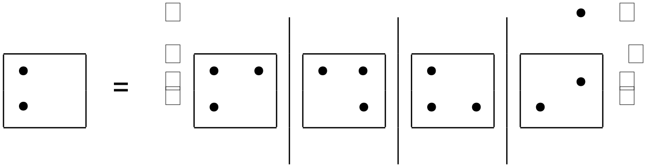


can be written as



Solving a matrix-vector equation: *Lights Out*

Solving an instance of *Lights Out* $\Rightarrow$ Solving a matrix-vector equation



$* \quad [a_1, a_2, a_3, a_4]$

Solving a matrix-vector equation

Fundamental Computational Problem: *Solving a matrix-vector equation*

- *input:* an $R \times C$ matrix A and an R-vector $\mathbf{b}$
- *output:* the C-vector $\mathbf{x}$ such that $A * \mathbf{x} = \mathbf{b}$

Solving a matrix-vector equation: 2 × 2 special case

Simple formula to solve

$$\begin{matrix} a. & c \\ b. & d \end{matrix} \cdot [x, y] = [p, q]$$

if $ad \,/\!= bc$:

$$x = \frac{dp - cq}{ad - bc} \text{ and } y = \frac{aq - bp}{ad - bc}$$

For example, to solve

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \cdot [x, y] = [-1, 1]$$

we set

$$x = \frac{4 \cdot -1 - 2 \cdot 1}{1 \cdot 4 - 2 \cdot 3} = \frac{-6}{-2} = 3$$

and

$$y = \frac{1 \cdot 1 - 3 \cdot -1}{1 \cdot 4 - 2 \cdot 3} = \frac{4}{-2} = -2$$

Later we study algorithms for more general cases.

Matrix-vector multiplication in terms of dot-products

Let M be an $R \times C$ matrix.

Dot-Product Definition of matrix-vector multiplication: $M * \mathbf{u}$ is the R-vector $\mathbf{v}$ such that $\mathbf{v}[r]$ is the dot-product of row r of M with $\mathbf{u}$.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 10 & 0 \end{bmatrix} * [3, -1] = [\ [1, 2] \cdot [3, -1],\ [3, 4] \cdot [3, -1],\ [10, 0] \cdot [3, -1]\ ]$$

$$= [1, 5, 30]$$