Review of vector terms

- A D-vector over $\mathbb{F}$ is a function with domain D and co-domain $\mathbb{F}$.
  $\mathbb{F}$ must be a field.
- The set of such vectors is written $\mathbb{F}^D$ (recall from *The Function*)
- An n-vector over $\mathbb{F}$ is a function with domain $\{0, 1, 2, \ldots, n-1\}$ and co-domain $\mathbb{F}$.
  Can also represent as an n-element list.

Vector algebraic properties

Addition

- **Addition is associative:** $(\mathbf{u}+\mathbf{v})+\mathbf{w}=\mathbf{u}+(\mathbf{v}+\mathbf{w})$
- **Addition is commutative:** $\mathbf{u}+\mathbf{v}=\mathbf{v}+\mathbf{u}$

Scalar-vector multiplication

- **Scalar-vector multiplication is associative:** $(\alpha\,\beta)\,\mathbf{v}=\alpha\,(\beta\,\mathbf{v})$

Both addition and scalar-vector multiplication

- **Scalar-vector multiplication distributes over addition:** $\alpha(\mathbf{u}+\mathbf{v})=\alpha\mathbf{u}+\alpha\mathbf{v}$

Dot-product

- **Dot-product is commutative:** $\mathbf{u}\cdot\mathbf{v}=\mathbf{v}\cdot\mathbf{u}$
- **Dot-product is homogeneous:** $(\alpha\mathbf{u})\cdot\mathbf{v}=\alpha(\mathbf{u}\cdot\mathbf{v})$
- **Dot-product distributes over addition:** $\mathbf{u}\cdot(\mathbf{v}+\mathbf{w})=\mathbf{u}\cdot\mathbf{v}+\mathbf{u}\cdot\mathbf{w}$

Solving a triangular system of linear equations

How to find solution to this linear system?

$$\begin{array}{rcrcr} [1, 0.5, -2, 4] & \cdot & \mathbf{x} & = & -8 \\ [0, 3, 3, 2] & \cdot & \mathbf{x} & = & 3 \\ [0, 0, 1, 5] & \cdot & \mathbf{x} & = & -4 \\ [0, 0, 0, 2] & \cdot & \mathbf{x} & = & 6 \end{array}$$

Write $\mathbf{x} = [x_1, x_2, x_3, x_4]$.
System becomes

$$\begin{array}{rcrcrcrcr} 1x_1 & + & 0.5x_2 & - & 2x_3 & + & 4x_4 & = & -8 \\ & & 3x_2 & + & 3x_3 & + & 2x_4 & = & 3 \\ & & & & 1x_3 & + & 5x_4 & = & -4 \\ & & & & & & 2x_4 & = & 6 \end{array}$$

Solving a triangular system of linear equations: Backward substitution

$$\begin{array}{rcrcrcrcr} 1x_1 & + & 0.5x_2 & - & 2x_3 & + & 4x_4 & = & -8 \\ & & 3x_2 & + & 3x_3 & + & 2x_4 & = & 3 \\ & & & & 1x_3 & + & 5x_4 & = & -4 \\ & & & & & & 2x_4 & = & 6 \end{array}$$

Solution strategy:

- Solve for x_4 using fourth equation.
- Plug value for x_4 into third equations and solve for x_3.
- Plug values for x_4 and x_3 into second equation and solve for x_2.
- Plug values for x_4, x_3, x_2 into first equation and solve for x_1.

[3] The Vector Space

Linear Combinations

An expression

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

is a *linear combination* of the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$.

The scalars $\alpha_1, \ldots, \alpha_n$ are the *coefficients* of the linear combination.

Example: One linear combination of $[2, 3.5]$ and $[4, 10]$ is

$$-5\,[2, 3.5] + 2\,[4, 10]$$

which is equal to $[-5 \cdot 2, -5 \cdot 3.5] + [2 \cdot 4, 2 \cdot 10]$

Another linear combination of the same vectors is

$$0\,[2, 3.5] + 0\,[4, 10]$$

which is equal to the zero vector $[0, 0]$.

Definition: A linear combination is *trivial* if the coefficients are all zero.

Linear Combinations: JunkCo

The JunkCo factory makes five products:





using various resources.

	metal	concrete	plastic	water	electricity
garden gnome	0	1.3	0.2	0.8	0.4
hula hoop	0	0	1.5	0.4	0.3
slinky	0.25	0	0	0.2	0.7
silly putty	0	0	0.3	0.7	0.5
salad shooter	0.15	0	0.5	0.4	0.8

For each product, a vector specifying how much of each resource is used per unit of product.

For making one gnome:

$\mathbf{v}_1 = \{$`metal:0, concrete:1.3, plastic:0.2, water:.8, electricity:0.4`$\}$

Linear Combinations: JunkCo

For making one gnome:
$\mathbf{v}_1 = \{\texttt{metal:0, concrete:1.3, plastic:0.2, water:0.8, electricity:0.4}\}$
For making one hula hoop:
$\mathbf{v}_2 = \{\texttt{metal:0, concrete:0, plastic:1.5, water:0.4, electricity:0.3}\}$
For making one slinky:
$\mathbf{v}_3 = \{\texttt{metal:0.25, concrete:0, plastic:0, water:0.2, electricity:0.7}\}$
For making one silly putty:
$\mathbf{v}_4 = \{\texttt{metal:0, concrete:0, plastic:0.3, water:0.7, electricity:0.5}\}$
For making one salad shooter:
$\mathbf{v}_5 = \{\texttt{metal:1.5, concrete:0, plastic:0.5, water:0.4, electricity:0.8}\}$

Suppose the factory chooses to make α_1 gnomes, α_2 hula hoops, α_3 slinkies, α_4 silly putties, and α_5 salad shooters.

Total resource utilization is $\mathbf{b} = \alpha_1\,\mathbf{v}_1 + \alpha_2\,\mathbf{v}_2 + \alpha_3\,\mathbf{v}_3 + \alpha_4\,\mathbf{v}_4 + \alpha_5\,\mathbf{v}_5$

Linear Combinations: JunkCo: Industrial espionage

Total resource utilization is $\mathbf{b} = \alpha_1\,\mathbf{v}_1 + \alpha_2\,\mathbf{v}_2 + \alpha_3\,\mathbf{v}_3 + \alpha_4\,\mathbf{v}_4 + \alpha_5\,\mathbf{v}_5$

Suppose I am spying on JunkCo.

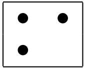
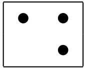
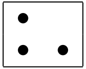
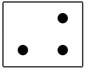
I find out how much metal, concrete, plastic, water, and electricity are consumed by the factory. That is, I know the vector $\mathbf{b}$. Can I use this knowledge to figure out how many gnomes they are making?

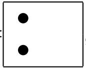
Computational Problem: *Expressing a given vector as a linear combination of other given vectors*

- *input:* a vector $\mathbf{b}$ and a list $[\mathbf{v}_1, \ldots, \mathbf{v}_n]$ of vectors
- *output:* a list $[\alpha_1, \ldots, \alpha_n]$ of coefficients such that $\mathbf{b} = \alpha_1\,\mathbf{v}_1 + \cdots + \alpha_n\,\mathbf{v}_n$ or a report that none exists.

Question: Is the solution unique?

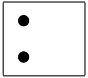
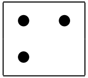
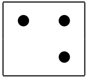
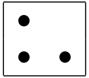
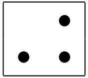
Lights Out

Button vectors for 2×2 *Lights Out:*    

For a given initial state vector $\mathbf{s} =$ , Which subset of button vectors sum to $\mathbf{s}$?

Reformulate in terms of linear combinations.

Write

 $= \alpha_1$  $+ \alpha_2$  $+ \alpha_3$  $+ \alpha_4$ 

What values for $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ make this equation true?

Solution: $\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 0, \alpha_4 = 0$

Solve an instance of *Lights Out* $\Rightarrow$ Which set of button vectors sum to $\mathbf{s}$?

$\Rightarrow$ Find subset of $GF(2)$ vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ whose sum equals $\mathbf{s}$ $\Rightarrow$ Express $\mathbf{s}$ as a linear combination of $\mathbf{v}_1, \ldots, \mathbf{v}_n$

Lights Out

We can solve the puzzle if we have an algorithm for

Computational Problem: *Expressing a given vector as a linear combination of other given vectors*

Span

Definition: The set of all linear combinations of some vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ is called the *span* of these vectors

Written Span $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$.

Span: Attacking the authentication scheme

If Eve knows the password satisfies

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

Then she can calculate right response to any challenge in Span $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$:

Proof: Suppose $\mathbf{a} = \alpha_1 \mathbf{a}_1 + \cdots + \alpha_m \mathbf{a}_m$. Then

$$\begin{aligned} \mathbf{a} \cdot \mathbf{x} &= (\alpha_1 \mathbf{a}_1 + \cdots + \alpha_m \mathbf{a}_m) \cdot \mathbf{x} \\ &= \alpha_1 \mathbf{a}_1 \cdot \mathbf{x} + \cdots + \alpha_m \mathbf{a}_m \cdot \mathbf{x} && \text{by distributivity} \\ &= \alpha_1 (\mathbf{a}_1 \cdot \mathbf{x}) + \cdots + \alpha_m (\mathbf{a}_m \cdot \mathbf{x}) && \text{by homogeneity} \\ &= \alpha_1 \beta_1 + \cdots + \alpha_m \beta_m \end{aligned}$$

Question: Any others? Answer will come later.

Span: $GF(2)$ vectors

Quiz: How many vectors are in Span $\{[1,1],[0,1]\}$ over the field $GF(2)$?

Answer: The linear combinations are

$$
\begin{aligned}
0\,[1,1] + 0\,[0,1] &= [0,0] \\
0\,[1,1] + 1\,[0,1] &= [0,1] \\
1\,[1,1] + 0\,[0,1] &= [1,1] \\
1\,[1,1] + 1\,[0,1] &= [1,0]
\end{aligned}
$$

Thus there are four vectors in the span.

Span: $GF(2)$ vectors

Question: How many vectors in Span $\{[1,1]\}$ over $GF(2)$?

Answer: The linear combinations are

$$0\,[1,1] = [0,0]$$
$$1\,[1,1] = [1,1]$$

Thus there are two vectors in the span.

Question: How many vectors in Span $\{\}$?

Answer: Only one: the zero vector

Question: How many vectors in Span $\{[2,3]\}$ over $\mathbb{R}$?

Answer: An infinite number: $\{\alpha\,[2,3] \ : \ \alpha \in \mathbb{R}\}$
Forms the line through the origin and $(2,3)$.

Generators

Definition: Let $\mathcal{V}$ be a set of vectors. If $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are vectors such that $\mathcal{V} = \text{Span}\ \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ then

- we say $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ is a *generating set* for $\mathcal{V}$;
- we refer to the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ as *generators* for $\mathcal{V}$.

Example: $\{[3, 0, 0], [0, 2, 0], [0, 0, 1]\}$ is a generating set for $\mathbb{R}^3$.

Proof: Must show two things:

1. Every linear combination is a vector in $\mathbb{R}^3$.
2. Every vector in $\mathbb{R}^3$ is a linear combination.

First statement is easy: every linear combination of 3-vectors over $\mathbb{R}$ is a 3-vector over $\mathbb{R}$, and $\mathbb{R}^3$ contains all 3-vectors over $\mathbb{R}$.

Proof of second statement: Let $[x, y, z]$ be any vector in $\mathbb{R}^3$. I must show it is a linear combination of my three vectors....

$$[x, y, z] = (x/3)\,[3, 0, 0] + (y/2)\,[0, 2, 0] + z\,[0, 0, 1]$$

Generators

Claim: Another generating set for $\mathbb{R}^3$ is $\{[1,0,0],[1,1,0],[1,1,1]\}$

Another way to prove that every vector in $\mathbb{R}^3$ is in the span:

- We already know $\mathbb{R}^3 = \text{Span }\{[3,0,0],[0,2,0],[0,0,1]\}$,
- so just show $[3,0,0]$, $[0,2,0]$, and $[0,0,1]$ are in $\text{Span }\{[1,0,0],[1,1,0],[1,1,1]\}$

$$\begin{aligned}
[3,0,0] &= 3[1,0,0] \\
[0,2,0] &= -2\,[1,0,0] + 2\,[1,1,0] \\
[0,0,1] &= -1\,[1,1,0] + 1\,[1,1,1]
\end{aligned}$$

Why is that sufficient?

- We already know any vector in $\mathbb{R}^3$ can be written as a linear combination of the old vectors.
- We know each old vector can be written as a linear combination of the new vectors.
- We can convert *a linear combination of linear combination of new vectors* into *a linear combination of new vectors*.

Generators

We can convert *a linear combination of linear combination of new vectors* into *a linear combination of new vectors*.

- Write $[x, y, z]$ as a linear combination of the old vectors:

$$[x, y, z] = (x/3)\,[3, 0, 0] + (y/2)\,[0, 2, 0] + z\,[0, 0, 1]$$

- Replace each old vector with an equivalent linear combination of the new vectors:

$$\begin{aligned}[x, y, z] = (x/3)\left(3\,[1, 0, 0]\right) \quad &+ \quad (y/2)\left(-2\,[1, 0, 0] + 2\,[1, 1, 0]\right) \\ &+ \quad z\left(-1\,[1, 1, 0] + 1\,[1, 1, 1]\right)\end{aligned}$$

- Multiply through, using distributivity and associativity:

$$[x, y, z] = x\,[1, 0, 0] - y\,[1, 0, 0] + y\,[1, 1, 0] - z\,[1, 1, 0] + z\,[1, 1, 1]$$

- Collect like terms, using distributivity:

$$[x, y, z] = (x - y)\,[1, 0, 0] + (y - z)\,[1, 1, 0] + z\,[1, 1, 1]$$

Solving a triangular system of linear equations: Backward substitution

$$\begin{array}{rcrcrcrcr} 1x_1 & + & 0.5x_2 & - & 2x_3 & + & 4x_4 & = & -8 \\ & & 3x_2 & + & 3x_3 & + & 2x_4 & = & 3 \\ & & & & 1x_3 & + & 5x_4 & = & -4 \\ & & & & & & 2x_4 & = & 6 \end{array}$$

$$\begin{array}{lrcl} & 2x_4 & = & 6 \\ \text{so} & x_4 & = & 6/2 = 3 \\ \\ & 1x_3 & = & -4 - 5x_4 = -4 - 5(3) = -19 \\ \text{so} & x_3 & = & -19/1 = -19 \\ \\ & 3x_2 & = & 3 - 3x_3 - 2x_4 = 3 - 2(3) - 3(-19) = 54 \\ \text{so} & x_2 & = & 54/3 = 18 \\ \\ & 1x_1 & = & -8 - 0.5x_2 + 2x_3 - 4x_4 = -8 - 4(3) + 2(-19) - 0.5(18) = -67 \\ \text{so} & x_1 & = & -67/1 = -67 \end{array}$$

Backsub Quiz

Use Back Substitution to solve the following triangular system of linear equations.

$$\begin{array}{rcrcrcl} 2x_1 & + & 2x_2 & - & 6x_3 & = & 0 \\ & & -5x_2 & + & 4x_3 & = & 7 \\ & & & & 2x_3 & = & 1 \end{array}$$

Solving a triangular system of linear equations: Backward substitution

Hack to implement backward substitution using vectors:

- Initialize vector `x` to zero vector.
- Procedure will populate `x` entry by entry.
- When it is time to populate x_i, entries $x_{i+1}, x_{i+2}, \ldots, x_n$ will be populated, and other entries will be zero.
- Therefore can use dot-product:
  - Suppose you are computing x_2 using $[0, 3, 3, 2] \cdot [x_1, x_2, x_3, x_4] = 3$
  - So far, vector $\mathtt{x} = [x_1, x_2, x_3, x_4] = [0, 0, -19, 3]$.
  - $x_2 := (3 - ([0, 3, 3, 2] \cdot \mathtt{x}))\,/3$

```
def triangular_solve(rowlist, b):
    x = zero_vec(rowlist[0].D)
    for i in reversed(range(len(rowlist))):
        x[i] = (b[i] - rowlist[i] * x)/rowlist[i][i]
    return x
```

Solving a triangular system of linear equations: Backward substitution

```
def triangular_solve(rowlist, b):
    x = zero_vec(rowlist[0].D)
    for i in reversed(range(len(rowlist))):
        x[i] = (b[i] - rowlist[i] * x)/rowlist[i][i]
    return x
```

Observations:

- If `rowlist[i][i]` is zero, procedure will raise `ZeroDivisionError`.
- If this never happens, solution found is the *only* solution to the system.

Solving a triangular system of linear equations: Backward substitution

```
def triangular_solve(rowlist, b):
    x = zero_vec(rowlist[0].D)
    for i in reversed(range(len(rowlist))):
        x[i] = (b[i] - rowlist[i] * x)/rowlist[i][i]
    return x
```

Our code only works when vectors in `rowlist` have domain $D = \{0, 1, 2, \ldots, n-1\}$.

For arbitrary domains, need to specify an ordering for which system is "triangular":

```
def triangular_solve(rowlist, label_list, b):
    x = zero_vec(set(label_list))
    for r in reversed(range(len(rowlist))):
        c = label_list[r]
        x[c] = (b[r] - x*rowlist[r])/rowlist[r][c]
    return x
```