Quiz

Parts 1 and 2: Describe two interpretations of the matrix-vector product $A\mathbf{v}$, one involving rows and one involving columns.

Part 3: Describe an interpretation of the matrix-matrix product AB, one involving either rows or columns.

Parts 4 and 5: What are the two spaces associated with a matrix M, and what do they have to do with the function defined by the rule $\mathbf{x} \mapsto M\mathbf{x}$?

Matrix-vector equation for sensor node

Define `D = {'radio', 'sensor', 'memory', 'CPU'}`.

Goal: Compute a D-vector $\mathbf{u}$ that, for each hardware component, gives the current drawn by that component.

Four test periods:

- total milliampere-seconds in these test periods $\mathbf{b} = [140, 170, 60, 170]$
- for each test period, vector specifying how long each hardware device was operating:
  - $\mathbf{duration}_1$ = `Vec(D, 'radio':.1, 'CPU':.3)`
  - $\mathbf{duration}_2$ = `Vec(D, 'sensor':.2, 'CPU':.4)`
  - $\mathbf{duration}_3$ = `Vec(D, 'memory':.3, 'CPU':.1)`
  - $\mathbf{duration}_4$ = `Vec(D, 'memory':.5, 'CPU':.4)`

To get $\mathbf{u}$, solve $A * \mathbf{x} = \mathbf{b}$ where $A = \begin{bmatrix} \mathbf{duration}_1 \\ \hline \mathbf{duration}_2 \\ \hline \mathbf{duration}_3 \\ \hline \mathbf{duration}_4 \end{bmatrix}$

The `solver` module, and floating-point arithmetic

For arithmetic over $\mathbb{R}$, Python uses floats, so round-off errors occur:

```
>>> 10.0**16 + 1 == 10.0**16
True
```

Consequently algorithms such as that used in `solve(A, b)` do not find exactly correct solutions.
To see if solution $\mathbf{u}$ obtained is a reasonable solution to $A * \mathbf{x} = \mathbf{b}$, see if the vector $\mathbf{b} - A * \mathbf{u}$ has entries that are close to zero:

```
>>> A = listlist2mat([[1,3],[5,7]])
>>> u = solve(A, b)
>>> b - A*u
Vec({0, 1},{0: -4.440892098500626e-16, 1: -8.881784197001252e-16})
```

The vector $\mathbf{b} - A * \mathbf{u}$ is called the *residual*. Easy way to test if entries of the residual are close to zero: compute the dot-product of the residual with itself:

```
>>> res = b - A*u
>>> res * res
9.860761315262648e-31
```

Checking the output from `solve(A, b)`

For some matrix-vector equations $A * \mathbf{x} = \mathbf{b}$, there is no solution.
In this case, the vector returned by `solve(A, b)` gives rise to a largeish residual:

```
>>> A = listlist2mat([[1,2],[4,5],[-6,1]])
>>> b = list2vec([1,1,1])
>>> u = solve(A, b)
>>> res = b - A*u
>>> res * res
0.24287856071964012
```

Some matrix-vector equations are *ill-conditioned*, which can prevent an algorithm using floats from getting even approximate solutions, even when solutions exists:

```
>>> A = listlist2mat([[1e20,1],[1,0]])
>>> b = list2vec([1,1])
>>> u = solve(A, b)
>>> b - A*u
Vec({0, 1},{0: 0.0, 1: 1.0})
```

We will not study conditioning in this course.

Triangular matrix

Recall: We considered *triangular* linear systems, e.g.

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

We can rewrite this linear system as a matrix-vector equation:

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

The matrix is a *triangular* matrix.

Definition: An $n \times n$ *upper triangular* matrix A is a matrix with the property that $A_{ij} = 0$ for $i > j$. Note that the entries forming the upper triangle can be be zero or nonzero.

We can use backward substitution to solve such a matrix-vector equation.

Triangular matrices will play an important role later.

Algebraic properties of matrix-vector multiplication

Proposition: Let A be an $R \times C$ matrix.

- For any C-vector $\mathbf{v}$ and any scalar α,

$$A * (\alpha\, \mathbf{v}) = \alpha\, (A * \mathbf{v})$$

- For any C-vectors $\mathbf{u}$ and $\mathbf{v}$,

$$A * (\mathbf{u} + \mathbf{v}) = A * \mathbf{u} + A * \mathbf{v}$$

Algebraic properties of matrix-vector multiplication

To prove

$$A * (\alpha\, \mathbf{v}) = \alpha\, (A * \mathbf{v})$$

we need to show corresponding entries are equal:

Need to show

$$\text{entry } i \text{ of } \quad A * (\alpha\, \mathbf{v}) = \text{entry } i \text{ of } \quad \alpha\, (A * \mathbf{v})$$

Proof: Write $A = \begin{bmatrix} \mathbf{a}_1 \\ \hline \vdots \\ \hline \mathbf{a}_m \end{bmatrix}$.

By dot-product def. of matrix-vector mult,

$$\begin{aligned} \text{entry } i \text{ of } \quad A * (\alpha\, \mathbf{v}) &= \mathbf{a}_i \cdot \alpha\, \mathbf{v} \\ &= \alpha\, (\mathbf{a}_i \cdot \mathbf{v}) \end{aligned}$$

by homogeneity of dot-product

By definition of scalar-vector multiply,

$$\begin{aligned} \text{entry } i \text{ of } \alpha\, (A * \mathbf{v}) &= \alpha\, (\text{entry } i \text{ of } A * \mathbf{v}) \\ &= \alpha\, (\mathbf{a}_i \cdot \mathbf{v}) \end{aligned}$$

by dot-product definition of matrix-vector multiply

QED

Algebraic properties of matrix-vector multiplication

To prove

$$A * (\mathbf{u} + \mathbf{v}) = A * \mathbf{u} + A * \mathbf{v}$$

we need to show corresponding entries are equal:

Need to show

$$\text{entry } i \text{ of } \quad A * (\mathbf{u} + \mathbf{v}) = \text{entry } i \text{ of } \quad A * \mathbf{u} + A * \mathbf{v}$$

Proof: Write $A = \begin{bmatrix} \mathbf{a}_1 \\ \hline \vdots \\ \hline \mathbf{a}_m \end{bmatrix}$.

By dot-product def. of matrix-vector mult,

$$\begin{aligned} \text{entry } i \text{ of } \quad A * (\mathbf{u} + \mathbf{v}) &= \mathbf{a}_i \cdot (\mathbf{u} + \mathbf{v}) \\ &= \mathbf{a}_i \cdot \mathbf{u} + \mathbf{a}_i \cdot \mathbf{v} \end{aligned}$$

by distributive property of dot-product

By dot-product def. of matrix-vector mult,

$$\begin{aligned} \text{entry } i \text{ of } \quad A * \mathbf{u} &= \mathbf{a}_i \cdot \mathbf{u} \\ \text{entry } i \text{ of } \quad A * \mathbf{v} &= \mathbf{a}_i \cdot \mathbf{v} \end{aligned}$$

so

$$\text{entry } i \text{ of } \quad A * \mathbf{u} + A * \mathbf{v} = \mathbf{a}_i \cdot \mathbf{u} + \mathbf{a}_i \cdot \mathbf{v}$$

QED

Matrix-matrix multiplication and function composition

Corresponding to an $R \times C$ matrix A over a field $\mathbb{F}$, there is a function

$$f : \mathbb{F}^C \longrightarrow \mathbb{F}^R$$

namely the function defined by $f(\mathbf{y}) = A * \mathbf{y}$

Matrix-matrix multiplication and function composition

Matrices A and B $\Rightarrow$ functions $f(\mathbf{y}) = A * \mathbf{y}$ and $g(\mathbf{x}) = B * \mathbf{x}$ and $h(\mathbf{x}) = (AB) * \mathbf{x}$

Matrix-Multiplication Lemma $f \circ g = h$

Example:

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \Rightarrow f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 + x_2 \\ x_2 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \Rightarrow g\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_1 + x_2 \end{bmatrix}$$

product $AB = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$

corresponds to function $h\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 + x_2 \\ x_1 + x_2 \end{bmatrix}$

$f \circ g\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = f\left(\begin{bmatrix} x_1 \\ x_1 + x_2 \end{bmatrix}\right) = \begin{bmatrix} 2x_1 + x_2 \\ x_1 + x_2 \end{bmatrix}$ so $f \circ g = h$

Matrix-matrix multiplication and function composition

Matrices A and B $\Rightarrow$ functions $f(\mathbf{y}) = A * \mathbf{y}$ and $g(\mathbf{x}) = B * \mathbf{x}$ and $h(\mathbf{x}) = (AB) * \mathbf{x}$

Matrix-Multiplication Lemma: $f \circ g = h$

Proof: Let columns of B be $\mathbf{b}_1, \ldots, \mathbf{b}_n$. By the matrix-vector definition of matrix-matrix multiplication, column j of AB is $A *$ (column j of B).

For any n-vector $\mathbf{x} = [x_1, \ldots, x_n]$,

$$\begin{aligned} g(\mathbf{x}) &= B * \mathbf{x} && \text{by definition of } g \\ &= x_1\mathbf{b}_1 + \cdots + x_n\mathbf{b}_n && \text{by linear combinations definition} \end{aligned}$$

Therefore

$$\begin{aligned} f(g(\mathbf{x})) &= f(x_1\mathbf{b}_1 + \cdots x_n\mathbf{b}_n) \\ &= x_1(f(\mathbf{b}_1)) + \cdots + x_n(f(\mathbf{b}_n)) && \text{by algebraic properties} \\ &= x_1(A * \mathbf{b}_1) + \cdots + x_n(A * \mathbf{b}_n) && \text{by definition of } f \\ &= x_1(\text{column 1 of } AB) + \cdots + x_n(\text{column } n \text{ of } AB) && \text{by matrix-vector def.} \\ &= (AB) * \mathbf{x} && \text{by linear-combinations def.} \\ &= h(\mathbf{x}) && \text{by definition of } h \end{aligned}$$

Associativity of matrix-matrix multiplication

Matrices A and B $\Rightarrow$ functions $f(\mathbf{y}) = A * \mathbf{y}$ and $g(\mathbf{x}) = B * \mathbf{x}$ and $h(\mathbf{x}) = (AB) * \mathbf{x}$

Matrix-Multiplication Lemma: $f \circ g = h$

Matrix-matrix multiplication corresponds to function composition.

Corollary: Matrix-matrix multiplication is associative:

$$(AB)C = A(BC)$$

Proof: Function composition is associative. QED

Example:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 \\ 1 & 2 \end{bmatrix} \right) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 5 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 5 \\ 1 & 7 \end{bmatrix}$$

$$\left(\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} -1 & 3 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -1 & 3 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 5 \\ 1 & 7 \end{bmatrix}$$

Matrices and their functions

Now we study the relationship between a matrix M and the function $\mathbf{x} \mapsto M * \mathbf{x}$

- *Easy:* Going from a matrix M to the function $\mathbf{x} \mapsto M * \mathbf{x}$
- *A little harder:* Going from the function $\mathbf{x} \mapsto M * \mathbf{x}$ to the matrix M.

In studying this relationship, we come up with the fundamental notion of a *linear transformation*.

From matrix to function

Starting with a M, define the function $f(\mathbf{x}) = M * x$.

Domain and co-domain?

If M is an $R \times C$ matrix over $\mathbb{F}$ then

- domain of f is $\mathbb{F}^C$
- co-domain of f is $\mathbb{F}^R$

Example: Let M be the matrix

	#	@	?
a	1	2	3
b	10	20	30

and define $f(\mathbf{x}) = M * \mathbf{x}$

- Domain of f is $\mathbb{R}^{\{\#,@,?\}}$.
- Co-domain of f is $\mathbb{R}^{\{a,b\}}$.

f maps

#	@	?
2	2	-2

to

a	b
0	0

Example: Define $f(\mathbf{x}) = \begin{bmatrix} 1 & 2 & 3 \\ 10 & 20 & 30 \end{bmatrix} * \mathbf{x}$.

- Domain of f is $\mathbb{R}^3$
- Co-domain of f is $\mathbb{R}^2$

f maps $[2, 2, -2]$ to $[0, 0]$

From function to matrix

We have a function $f : \mathbb{F}^A \longrightarrow \mathbb{F}^B$

We want to compute matrix M such that $f(\mathbf{x}) = M * \mathbf{x}$.

- Since the domain is $\mathbb{F}^A$, we know that the input $\mathbf{x}$ is an A-vector.
- For the product $M * \mathbf{x}$ to be legal, we need the column-label set of M to be A.
- Since the co-domain is $\mathbb{F}^B$, we know that the output $f(\mathbf{x}) = M * \mathbf{x}$ is B-vector.
- To achieve that, we need row-label set of M to be B.

Now we know that M must be a $B \times A$ matrix....

... but what about its entries?

From function to matrix

- We have a function $f : \mathbb{F}^n \longrightarrow \mathbb{F}^m$
- We think there is an $m \times n$ matrix M such that $f(\mathbf{x}) = M * \mathbf{x}$

How to go from the function f to the entries of M?

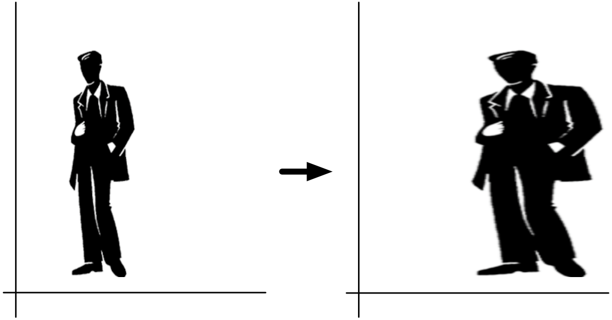
- Write mystery matrix in terms of its columns: $M = \left[\begin{array}{c|c|c} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{array} \right]$
- Use standard generators $\mathbf{e}_1 = [1, 0, \ldots, 0, 0], \ldots, \mathbf{e}_n = [0, \ldots, 0, 1]$ with *linear-combinations* definition of matrix-vector multiplication:

$$f(\mathbf{e}_1) = \left[\begin{array}{c|c|c} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{array} \right] * [1, 0, \ldots, 0, 0] = \mathbf{v}_1$$

$$\vdots$$

$$f(\mathbf{e}_n) = \left[\begin{array}{c|c|c} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{array} \right] * [0, 0, \ldots, 0, 1] = \mathbf{v}_n$$

From function to matrix: horizontal scaling

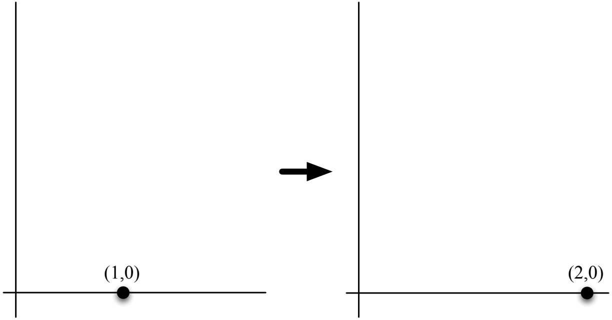


Define $s([x, y])$ = stretching by two in horizontal direction
Assume $s([x, y]) = M * [x, y]$ for some matrix M.

- We know $s([1, 0]) = [2, 0]$ because we are stretching by two in horizontal direction
- We know $s([0, 1]) = [0, 1]$ because no change in vertical direction.

Therefore $M = \left[\begin{array}{c|c} 2 & 0 \\ 0 & 1 \end{array}\right]$

From function to matrix: horizontal scaling

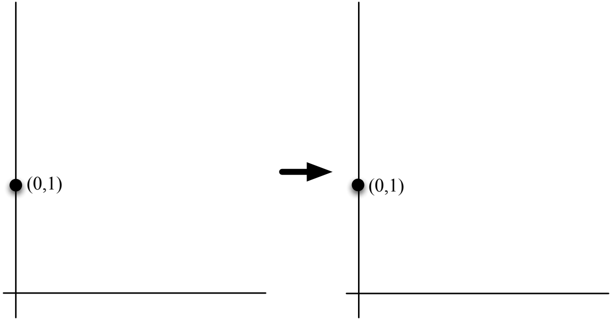


Define $s([x, y])$ = stretching by two in horizontal direction
Assume $s([x, y]) = M * [x, y]$ for some matrix M.

- We know $s([1, 0]) = [2, 0]$ because we are stretching by two in horizontal direction
- We know $s([0, 1]) = [0, 1]$ because no change in vertical direction.

Therefore $M = \left[\begin{array}{c|c} 2 & 0 \\ 0 & 1 \end{array} \right]$

From function to matrix: horizontal scaling



Define $s([x, y])$ = stretching by two in horizontal direction
Assume $s([x, y]) = M * [x, y]$ for some matrix M.

- We know $s([1, 0]) = [2, 0]$ because we are stretching by two in horizontal direction
- We know $s([0, 1]) = [0, 1]$ because no change in vertical direction.

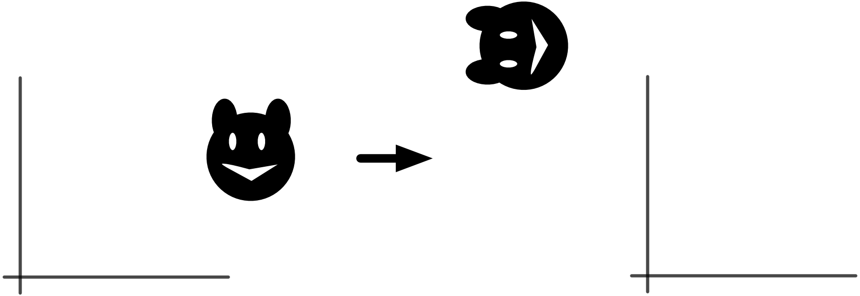
Therefore $M = \left[\begin{array}{c|c} 2 & 0 \\ 0 & 1 \end{array} \right]$

From function to matrix: rotation by 90 degrees

Define $r([x, y])$ = rotation by 90 degrees
Assume $r([x, y]) = M * [x, y]$ for some matrix M.

- We know rotating $[1, 0]$ should give $[0, 1]$ so $r([1, 0]) = [0, 1]$
- We know rotating $[0, 1]$ should give $[-1, 0]$ so $r([0, 1]) = [-1, 0]$

Therefore $M = \left[\begin{array}{c|c} 0 & -1 \\ 1 & 0 \end{array}\right]$

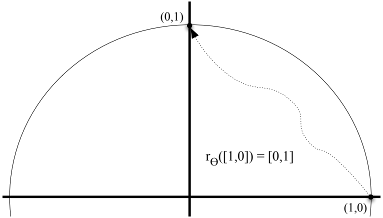


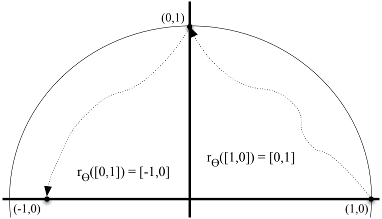
From function to matrix: rotation by 90 degrees

Define $r([x, y])$ = rotation by 90 degrees
Assume $r([x, y]) = M * [x, y]$ for some matrix M.

- We know rotating $[1, 0]$ should give $[0, 1]$ so $r([1, 0]) = [0, 1]$
- We know rotating $[0, 1]$ should give $[-1, 0]$ so $r([0, 1]) = [-1, 0]$

Therefore $M = \left[\begin{array}{c|c} 0 & -1 \\ 1 & 0 \end{array}\right]$



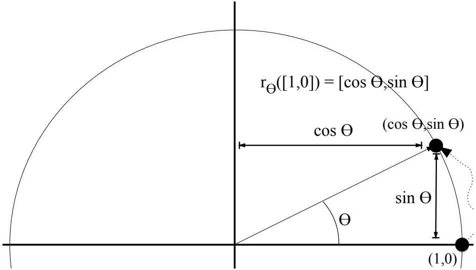


From function to matrix: rotation by θ degrees

Define $r([x, y])$ = rotation by θ.
Assume $r([x, y]) = M * [x, y]$ for some matrix M.

- We know $r([1, 0]) = [\cos\theta, \sin\theta]$ so column 1 is $[\cos\theta, \sin\theta]$
- We know $r([0, 1]) = [-\sin\theta, \cos\theta]$ so column 2 is $[-\sin\theta, \cos\theta]$

Therefore $M = \left[\begin{array}{c|c} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{array}\right]$



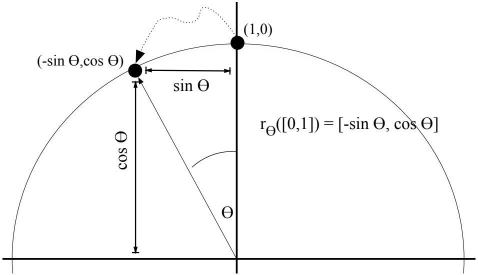
From function to matrix: rotation by θ degrees

Define $r([x, y]) =$ rotation by θ.

Assume $r([x, y]) = M * [x, y]$ for some matrix M.

- We know $r([1, 0]) = [\cos\theta, \sin\theta]$ so column 1 is $[\cos\theta, \sin\theta]$
- We know $r([0, 1]) = [-\sin\theta, \cos\theta]$ so column 2 is $[-\sin\theta, \cos\theta]$

Therefore $M = \left[\begin{array}{c|c} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{array}\right]$



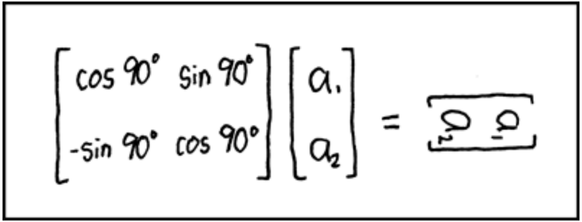
From function to matrix: rotation by θ degrees

Define $r([x, y])$ = rotation by θ.
Assume $r([x, y]) = M * [x, y]$ for some matrix M.

- We know $r([1, 0]) = [\cos\theta, \sin\theta]$ so column 1 is $[\cos\theta, \sin\theta]$
- We know $r([0, 1]) = [-\sin\theta, \cos\theta]$ so column 2 is $[-\sin\theta, \cos\theta]$

Therefore $M = \left[\begin{array}{c|c} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{array}\right]$

For clockwise rotation by 90 degrees, plug in θ = -90 degrees...



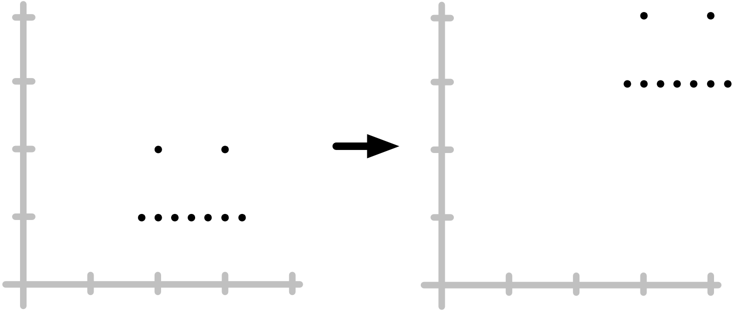
Matrix Transform (`http://xkcd.com/824`)

From function to matrix: translation

$t([x,y])$ = translation by $[1,2]$. Assume $t([x,y]) = M * [x,y]$ for some matrix M.

- We know $t([1,0]) = [2,2]$ so column 1 is $[2,2]$.
- We know $t([0,1]) = [1,3]$ so column 2 is $[1,3]$.

Therefore $M = \left[\begin{array}{c|c} 2 & 1 \\ 2 & 3 \end{array}\right]$

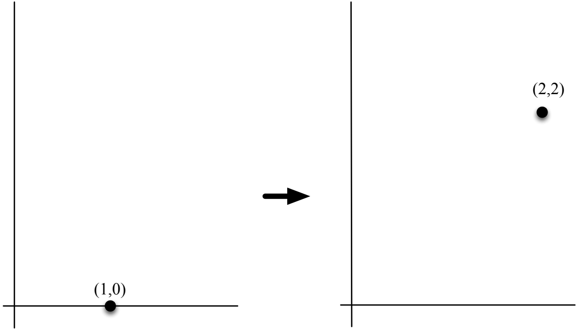


From function to matrix: translation

$t([x, y]) =$ translation by $[1, 2]$. Assume $t([x, y]) = M * [x, y]$ for some matrix M.

- We know $t([1, 0]) = [2, 2]$ so column 1 is $[2, 2]$.
- We know $t([0, 1]) = [1, 3]$ so column 2 is $[1, 3]$.

Therefore $M = \left[\begin{array}{c|c} 2 & 1 \\ 2 & 3 \end{array}\right]$

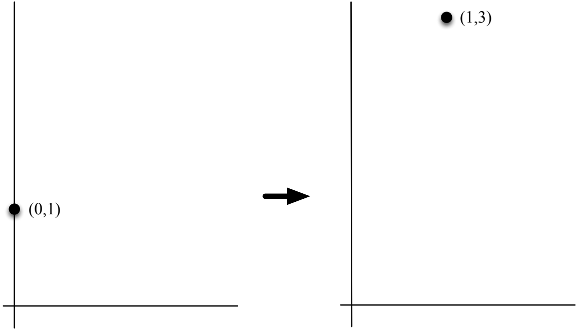


From function to matrix: translation

$t([x, y])$ = translation by $[1, 2]$. Assume $t([x, y]) = M * [x, y]$ for some matrix M.

- We know $t([1, 0]) = [2, 2]$ so column 1 is $[2, 2]$.
- We know $t([0, 1]) = [1, 3]$ so column 2 is $[1, 3]$.

Therefore $M = \left[\begin{array}{c|c} 2 & 1 \\ 2 & 3 \end{array}\right]$



From function to matrix: identity function

Consider the function $f : \mathbb{R}^4 \longrightarrow \mathbb{R}^4$ defined by $f(\mathbf{x}) = \mathbf{x}$
This is the identity function on $\mathbb{R}^4$.

Assume $f(\mathbf{x}) = M * \mathbf{x}$ for some matrix M.

Plug in the standard generators $\mathbf{e}_1 = [1, 0, 0, 0], \mathbf{e}_2 = [0, 1, 0, 0], \mathbf{e}_3 = [0, 0, 1, 0], \mathbf{e}_4 = [0, 0, 0, 1]$

- $f(\mathbf{e}_1) = \mathbf{e}_1$ so first column is $\mathbf{e}_1$
- $f(\mathbf{e}_2) = \mathbf{e}_2$ so second column is $\mathbf{e}_2$
- $f(\mathbf{e}_3) = \mathbf{e}_3$ so third column is $\mathbf{e}_3$
- $f(\mathbf{e}_4) = \mathbf{e}_4$ so fourth column is $\mathbf{e}_4$

$$\text{So } M = \left[\begin{array}{c|c|c|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

Identity function $f(\mathbf{x})$ corresponds to identity matrix $\mathbb{1}$

Diagonal matrices

Let $d_1, \ldots, d_n$ be real numbers. Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be the function such that $f([x_1, \ldots, x_n]) = [d_1 x_1, \ldots, d_n x_n]$. The matrix corresponding to this function is

$$\begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix}$$

Such a matrix is called a *diagonal* matrix because the only entries allowed to be nonzero form a diagonal.

Definition: For a domain D, a $D \times D$ matrix M is a *diagonal* matrix if $M[r, c] = 0$ for every pair $r, c \in D$ such that $r \neq c$.

Special case: $d_1 = \cdots = d_n = 1$. In this case, $f(\mathbf{x}) = \mathbf{x}$ (*identity function*)

The matrix $\begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix}$ is an identity matrix.

Which functions can be expressed as matrix-vector products?

In each example, we *assumed* the function could be expressed as a matrix-vector product.

How can we verify that assumption?

We'll state two algebraic properties.

- If a function can be expressed as a matrix-vector product $\mathbf{x} \mapsto M * \mathbf{x}$, it has these properties.
- If the function from $\mathbb{F}^C$ to $\mathbb{F}^R$ has these properties, it can be expressed as a matrix-vector product.

Which functions can be expressed as matrix-vector products?

Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces over a field $\mathbb{F}$.

Suppose a function $f : \mathcal{V} \longrightarrow \mathcal{W}$ satisfies two properties:

Property L1: For every vector $\mathbf{v}$ in $\mathcal{V}$ and every scalar α in $\mathbb{F}$,

$$f(\alpha\,\mathbf{v}) = \alpha\, f(\mathbf{v})$$

Property L2: For every two vectors $\mathbf{u}$ and $\mathbf{v}$ in $\mathcal{V}$,

$$f(\mathbf{u} + \mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v})$$

We then call f a *linear transformation* .

Proposition: Let M be an $R \times C$ matrix, and suppose $f : \mathbb{F}^C \mapsto \mathbb{F}^R$ is defined by $f(\mathbf{x}) = M * \mathbf{x}$. Then f is a linear transformation.

Proof: Certainly $\mathbb{F}^C$ and $\mathbb{F}^R$ are vector spaces.

We showed that $M * (\alpha\,\mathbf{v}) = \alpha\, M * \mathbf{v}$. This proves that f satisfies Property L1.

We showed that $M * (\mathbf{u} + \mathbf{v}) = M * \mathbf{u} + M * \mathbf{v}$. This proves that f satisfies Property L2.
QED