TMA4115 Matematikk 3

Andrew Stacey

Norges Teknisk-Naturvitenskapelige Universitet
Trondheim

Spring 2010

Lecture 15: Euclidean Space, Sweet Euclidean Space

Andrew Stacey

Norges Teknisk-Naturvitenskapelige Universitet
Trondheim

3rd March 2010

Key Points

- “Vector spaces” are the right “homes” for the inputs and outputs of matrices
- “Vector spaces” are spaces that “look like $\mathbb{R}^n$”

Recap

- Special Processes $\longleftrightarrow$ Matrices

Static versus Dynamic

Matrices	Special Processes

Static versus Dynamic

Matrices	Special Processes
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 4 & 6 \end{bmatrix}$	$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} x_1 + 2x_2 \\ 3x_1 + 4x_2 \\ 4x_1 + 6x_2 \end{bmatrix}$

Static versus Dynamic

Matrices	Special Processes
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 4 & 6 \end{bmatrix}$	$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} x_1 + 2x_2 \\ 3x_1 + 4x_2 \\ 4x_1 + 6x_2 \end{bmatrix}$
Data	Function
Representation	Actuality
Static	Dynamic

Implementation

```
apply_matrix_A (x)
{

  for (i=1; i<=m; i++) {
     y.i = 0;
     for (j=1; j<=n; j++) {
       y.i += A.i.j*x.j;
       }
     }
  }
  return y;
}
```

Implementation

```
vector apply_matrix_A (vector x)
{
  vector y;
  for (i=1; i<=m; i++) {
     y.i = 0;
     for (j=1; j<=n; j++) {
       y.i += A.i.j*x.j;
       }
     }
  }
  return y;
}
```

Implementation

```
m-vector apply_matrix_A (n-vector x)
{
  m-vector y;
  for (i=1; i<=m; i++) {
     y.i = 0;
     for (j=1; j<=n; j++) {
       y.i += A.i.j*x.j;
       }
     }
  }
  return y;
}
```

Implementation of “Do Nothing”

Implementation of “Do Nothing”

```
do_nothing(x)
{
  return x;
}
```

Implementation of “Do Nothing”

```
do_nothing(x)
{
  return x;
}
```

With extremely strict typing:

```
type do_nothing(type x)
{
  return x;
}
```

Implementation of “Do Nothing”

```
do_nothing(x)
{
  return x;
}
```

With extremely strict typing:

```
type do_nothing(type x)
{
  return x;
}
```

- For each `type`, a different process.

Implementation of “Do Nothing”

```
do_nothing(x)
{
  return x;
}
```

With extremely strict typing:

```
type do_nothing(type x)
{
  return x;
}
```

- For each `type`, a different process.
- For each n, a different identity matrix.

Implementation of “Say Sausages”

```
say_sausages(x)
{
  return ‘‘sausages’’;
}
```

Implementation of “Say Sausages”

```
say_sausages(x)
{
  return ‘‘sausages’’;
}
```

With extremely strict typing:

```
food say_sausages(type x)
{
  return ‘‘sausages’’;
}
```

Implementation of “Say Sausages”

```
say_sausages(x)
{
  return ‘‘sausages’’;
}
```

With extremely strict typing:

```
food say_sausages(type x)
{
  return ‘‘sausages’’;
}
```

- For each `type`, a different process.

Mathematical Data Typing

Rough Definition

A
set
or
mathematical data type
is a collection of elements that can be precisely defined

i.e. the question:
"is x in X?" ($x \in X$?)
always returns TRUE or FALSE

Mathematical Data Typing

Rough Definition

A

set

or

mathematical data type

is a collection of elements that can be precisely defined

i.e. the question:

"is x in X?" ($x \in X$?)

always returns TRUE or FALSE

A process always has a set of

valid inputs

and

possible outputs

Typing Wrapper

```
boolean can_I_do_function (function f, element x)
{
  if (f.valid_inputs = x.containing_set)
  {
     return TRUE;
  }
  else
  {
     return FALSE;
  }
}
```

Typing Wrapper

```
boolean can_I_do_function (function f, element x)
{
  if (f.valid_inputs = x.containing_set)
  {
     return TRUE;
  }
  else
  {
     return FALSE;
  }
}
```

Caution

Always declare your variables!

A Home for Vectors

Question

1. What is the right set of “valid inputs” for a matrix?
2. What is the right set of “possible outputs” for a matrix?

A Home for Vectors

Question

1. What is the right set of “valid inputs” for a matrix?
2. What is the right set of “possible outputs” for a matrix?

Answer

1. All vectors of the right length
2. All vectors of the right length

A Home for Vectors

Question

1. What is the right set of “valid inputs” for a matrix?
2. What is the right set of “possible outputs” for a matrix?

Answer

1. All vectors of the right length
2. All vectors of the right length

Note: possible outputs not actual outputs.

Vectors

Question

What is a vector?

Vectors

Question

What is a vector?

Answer

An ordered list of n real numbers.

Vectors

Question

What is a vector?

Answer

An ordered list of n real numbers.

Example

$(1, 2, 3, 4)$, $(\pi, \mathrm{e}, \pi^{\mathrm{e}}, \mathrm{e}^{\pi})$

Vectors

Question

What is a vector?

Answer

An ordered list of n real numbers.

Example

$(1, 2, 3, 4)$, $(\pi, \mathrm{e}, \pi^{\mathrm{e}}, \mathrm{e}^{\pi})$

Exact notation not important so long as order is clear.

$$(1, 2), \quad \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \begin{matrix} & 2 \\ 1 & \end{matrix}$$

Vectors

Question

What is a vector?

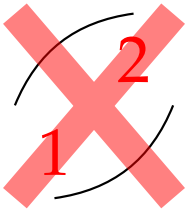
Answer

An ordered list of n real numbers.

Example

$(1, 2, 3, 4)$, $(\pi, \mathrm{e}, \pi^{\mathrm{e}}, \mathrm{e}^{\pi})$

Exact notation not important so long as order is clear.

$(1, 2)$, $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$, 

Euclidean Space

Definition

n–dimensional (Euclidean) space
is the set of all vectors (of real numbers) of length n

Notation: $\mathbb{R}^n$

Euclidean Space

Definition

n–dimensional (Euclidean) space
is the set of all vectors (of real numbers) of length n

Notation: $\mathbb{R}^n$

- EP: n–dimensional space

Euclidean Space

Definition

n–dimensional (Euclidean) space
is the set of all vectors (of real numbers) of length n

Notation: $\mathbb{R}^n$

- EP: n–dimensional space (bit vague)

Euclidean Space

Definition

n–dimensional (Euclidean) space
is the set of all vectors (of real numbers) of length n

Notation: $\mathbb{R}^n$

- EP: n–dimensional space (bit vague)
- Euclidean can imply distance as well

Euclidean Space

Definition

n–dimensional (Euclidean) space
is the set of all vectors (of real numbers) of length n

Notation: $\mathbb{R}^n$

- EP: n–dimensional space (bit vague)
- Euclidean can imply distance as well
- "$\mathbf{x} \in \mathbb{R}^n$" means "let $\mathbf{x}$ be an ordered list of n real numbers"

Euclidean Space

Definition

n–dimensional (Euclidean) space
is the set of all vectors (of real numbers) of length n

Notation: $\mathbb{R}^n$

- EP: n–dimensional space (bit vague)
- Euclidean can imply distance as well
- "$\mathbf{x} \in \mathbb{R}^n$" means "let $\mathbf{x}$ be an ordered list of n real numbers"
- "$\mathbf{x} \in \mathbb{R}^n$?" means "is $\mathbf{x}$ an ordered list of n real numbers?"

. . . The Company That He Keeps

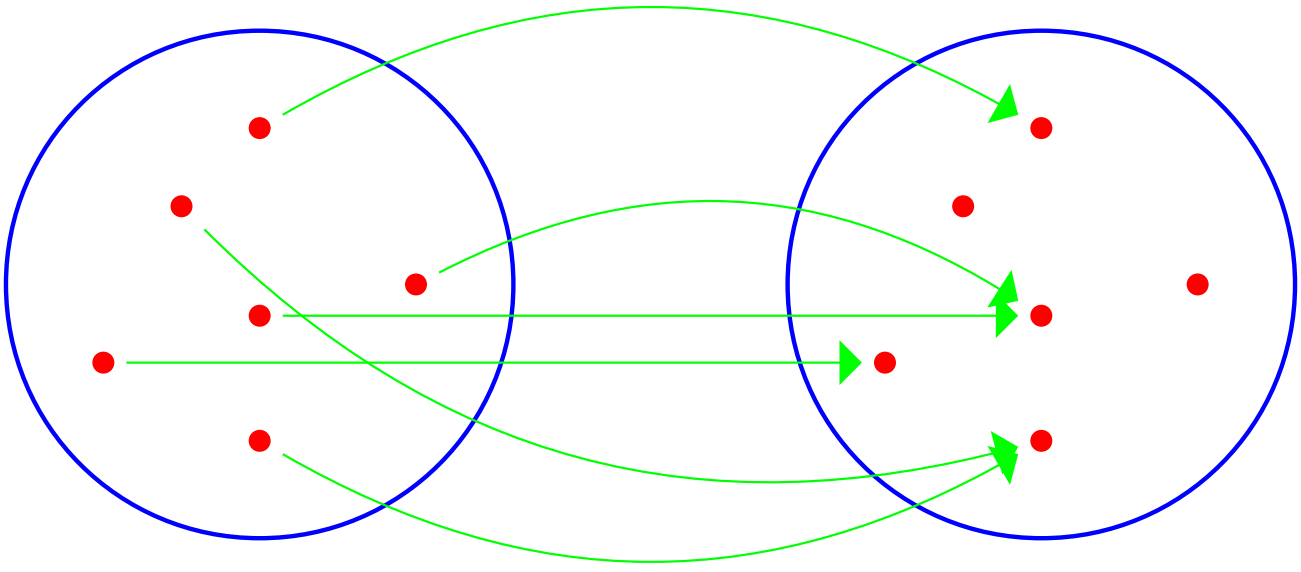
Idea

Use vectors to study matrices.

. . . The Company That He Keeps

Idea

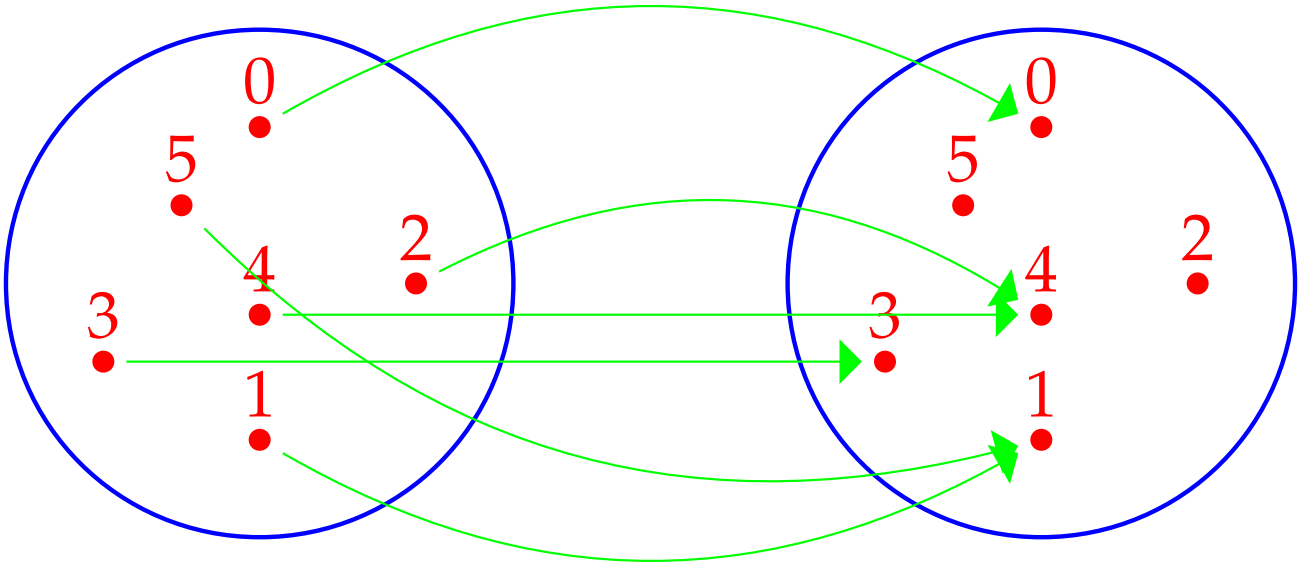
Use vectors to study matrices.



. . . The Company That He Keeps

Idea

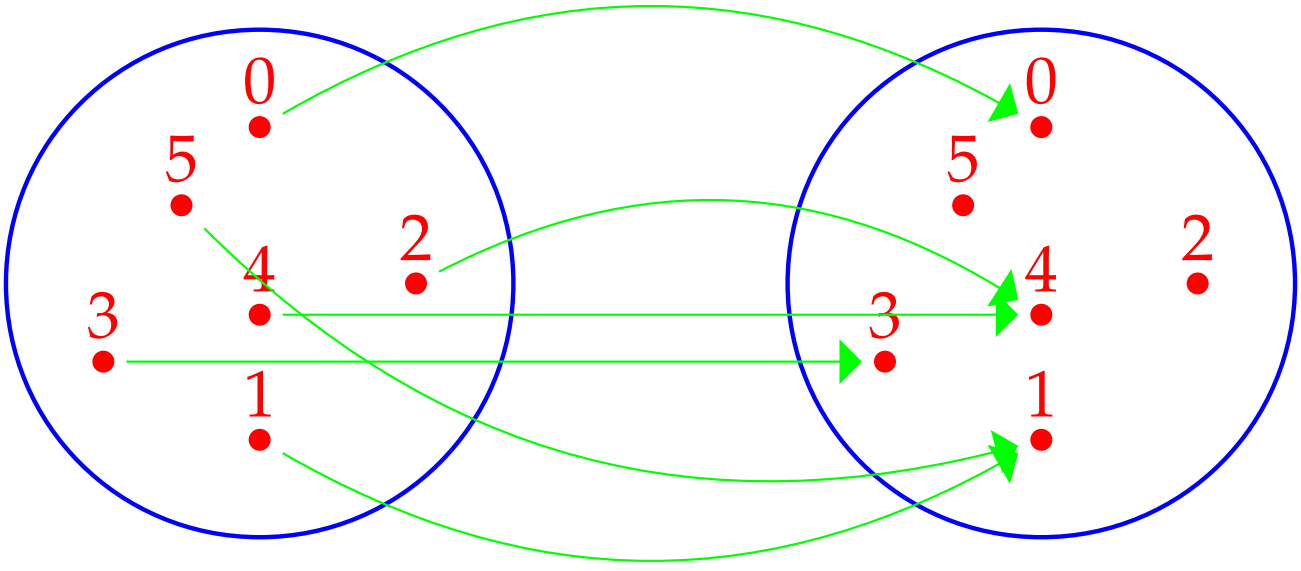
Use vectors to study matrices.



. . . The Company That He Keeps

Idea

Use vectors to study matrices.



$n \mapsto n^2 \mod 6$

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

What structure is on $\mathbb{R}$?

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

What structure is on $\mathbb{R}$?

1. Addition: $x + y$

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

What structure is on $\mathbb{R}$?

1. Addition: $x + y$
2. Subtraction: $x - y$

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

What structure is on $\mathbb{R}$?

1. Addition: $x + y$
2. Subtraction: $x - y$
3. Multiplication: $x \cdot y$

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

What structure is on $\mathbb{R}$?

1. Addition: $x + y$
2. Subtraction: $x - y$
3. Multiplication: $x \cdot y$
4. Additive Inverse: $-x$

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

What structure is on $\mathbb{R}$?

1. Addition: $x + y$
2. Subtraction: $x - y$
3. Multiplication: $x \cdot y$
4. Additive Inverse: $-x$
5. Multiplicative Inverse: x^{-1}

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

What structure is on $\mathbb{R}$?

1. Addition: $x + y$
2. Subtraction: $x - y$
3. Multiplication: $x \cdot y$
4. Additive Inverse: $-x$
5. Multiplicative Inverse: x^{-1}
6. Special Elements: 0, 1

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

What structure is on $\mathbb{R}$?

1. Addition: $x + y$
2. Subtraction: $x - y$
3. Multiplication: $x \cdot y$
4. Additive Inverse: $-x$
5. Multiplicative Inverse: x^{-1}
6. Special Elements: 0, 1
7. Length: $|x|$

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

What structure is on $\mathbb{R}$?

1. Addition: $x + y$
2. Subtraction: $x - y$
3. Multiplication: $x \cdot y$
4. Additive Inverse: $-x$
5. Multiplicative Inverse: x^{-1}
6. Special Elements: 0, 1
7. Length: $|x|$
8. Scaling: λx

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

What structure is on $\mathbb{R}$?

1. Addition: $x + y$
2. Subtraction: $x - y$
3. Multiplication: $x \cdot y$
4. Additive Inverse: $-x$
5. Multiplicative Inverse: x^{-1}
6. Special Elements: 0, 1
7. Length: $|x|$
8. Scaling: λx
9. Comparision: $x \geq y$

Structure Reveals Structure

Idea

Use structure of $\mathbb{R}^n$ to study matrices.

What structure is on $\mathbb{R}$?

1. Addition: $x + y$
2. Subtraction: $x - y$
3. Multiplication: $x \cdot y$
4. Additive Inverse: $-x$
5. Multiplicative Inverse: x^{-1}
6. Special Elements: 0, 1
7. Length: $|x|$
8. Scaling: λx
9. Comparision: $x \geq y$
10. More?

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just . . .

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

1. Addition: $x + y$
2. Subtraction: $x - y$
3. Multiplication: $x \cdot y$
4. Additive Inverse: $-x$
5. Multiplicative Inverse: x^{-1}
6. Special Elements: 0, 1
7. Length: $|x|$
8. Scaling: λx
9. Comparision: $x \geq y$

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

1. Addition: $x + y$ $\qquad ax + ay = a(x + y)$
2. Subtraction: $x - y$
3. Multiplication: $x \cdot y$
4. Additive Inverse: $-x$
5. Multiplicative Inverse: x^{-1}
6. Special Elements: 0, 1
7. Length: $|x|$
8. Scaling: λx
9. Comparision: $x \geq y$

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

1. Addition: $x + y$ $\quad ax + ay = a(x + y)$
2. Subtraction: $x - y$ $\quad ax - ay = a(x - y)$
3. Multiplication: $x \cdot y$
4. Additive Inverse: $-x$
5. Multiplicative Inverse: x^{-1}
6. Special Elements: 0, 1
7. Length: $|x|$
8. Scaling: λx
9. Comparision: $x \geq y$

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

1. Addition: $x + y$ $\quad ax + ay = a(x + y)$
2. Subtraction: $x - y$ $\quad ax - ay = a(x - y)$
3. Multiplication: $x \cdot y$ $\quad ax \cdot ay = a^2(x \cdot y)$
4. Additive Inverse: $-x$
5. Multiplicative Inverse: x^{-1}
6. Special Elements: 0, 1
7. Length: $|x|$
8. Scaling: λx
9. Comparision: $x \geq y$

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

1. Addition: $x + y$ $\quad ax + ay = a(x + y)$
2. Subtraction: $x - y$ $\quad ax - ay = a(x - y)$
3. Multiplication: $x \cdot y$ $\quad ax \cdot ay = a^2(x \cdot y)$
4. Additive Inverse: $-x$ $\quad -ax = a(-x)$
5. Multiplicative Inverse: x^{-1}
6. Special Elements: 0, 1
7. Length: $|x|$
8. Scaling: λx
9. Comparision: $x \geq y$

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

1. Addition: $x + y$	$ax + ay = a(x + y)$
2. Subtraction: $x - y$	$ax - ay = a(x - y)$
3. Multiplication: $x \cdot y$	$ax \cdot ay = a^2(x \cdot y)$
4. Additive Inverse: $-x$	$-ax = a(-x)$
5. Multiplicative Inverse: x^{-1}	$(ax)^{-1} = a^{-1}x^{-1}$
6. Special Elements: 0, 1	
7. Length: $\lvert x \rvert$	
8. Scaling: λx	
9. Comparision: $x \geq y$	

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

1. Addition: $x + y$	$ax + ay = a(x + y)$
2. Subtraction: $x - y$	$ax - ay = a(x - y)$
3. Multiplication: $x \cdot y$	$ax \cdot ay = a^2(x \cdot y)$
4. Additive Inverse: $-x$	$-ax = a(-x)$
5. Multiplicative Inverse: x^{-1}	$(ax)^{-1} = a^{-1}x^{-1}$
6. Special Elements: 0, 1	$a0 = 0$, $a1 = a$
7. Length: $\lvert x \rvert$	
8. Scaling: λx	
9. Comparision: $x \geq y$	

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

1. Addition: $x + y$	$ax + ay = a(x + y)$
2. Subtraction: $x - y$	$ax - ay = a(x - y)$
3. Multiplication: $x \cdot y$	$ax \cdot ay = a^2(x \cdot y)$
4. Additive Inverse: $-x$	$-ax = a(-x)$
5. Multiplicative Inverse: x^{-1}	$(ax)^{-1} = a^{-1}x^{-1}$
6. Special Elements: 0, 1	$a0 = 0$, $a1 = a$
7. Length: $\|x\|$	$\|ax\| = \|a\|\|x\|$
8. Scaling: λx	
9. Comparision: $x \geq y$	

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

1. Addition: $x + y$	$ax + ay = a(x + y)$
2. Subtraction: $x - y$	$ax - ay = a(x - y)$
3. Multiplication: $x \cdot y$	$ax \cdot ay = a^2(x \cdot y)$
4. Additive Inverse: $-x$	$-ax = a(-x)$
5. Multiplicative Inverse: x^{-1}	$(ax)^{-1} = a^{-1}x^{-1}$
6. Special Elements: 0, 1	$a0 = 0$, $a1 = a$
7. Length: $\|x\|$	$\|ax\| = \|a\|\|x\|$
8. Scaling: λx	$\lambda(ax) = a(\lambda x)$
9. Comparision: $x \geq y$	

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

1.	Addition: $x + y$	$ax + ay = a(x + y)$
2.	Subtraction: $x - y$	$ax - ay = a(x - y)$
3.	Multiplication: $x \cdot y$	$ax \cdot ay = a^2(x \cdot y)$
4.	Additive Inverse: $-x$	$-ax = a(-x)$
5.	Multiplicative Inverse: x^{-1}	$(ax)^{-1} = a^{-1}x^{-1}$
6.	Special Elements: 0, 1	$a0 = 0$, $a1 = a$
7.	Length: $\lvert x \rvert$	$\lvert ax \rvert = \lvert a \rvert \lvert x \rvert$
8.	Scaling: λx	$\lambda(ax) = a(\lambda x)$
9.	Comparision: $x \geq y$	$ax \geq ay$ or $ax \leq ay$)

Structure Reveals Structure

Useful Structure

Useful structure: interacts nicely with matrices.

A 1×1–matrix is just ... a number! $\begin{bmatrix} a \end{bmatrix}$ is $x \mapsto ax$

1.	Addition: $x + y$	$ax + ay = a(x + y)$
2.	Subtraction: $x - y$	$ax - ay = a(x - y)$
3.	Multiplication: $x \cdot y$	$ax \cdot ay = a^2(x \cdot y)$
4.	Additive Inverse: $-x$	$-ax = a(-x)$
5.	Multiplicative Inverse: x^{-1}	$(ax)^{-1} = a^{-1}x^{-1}$
6.	Special Elements: 0, (1)	$a0 = 0$, $a1 = a$
7.	Length: $\lvert x \rvert$	$\lvert ax \rvert = \lvert a \rvert \lvert x \rvert$
8.	Scaling: λx	$\lambda(ax) = a(\lambda x)$
9.	Comparision: $x \geq y$	$ax \geq ay$ or $ax \leq ay$)

More Generally

Define addition, scaling, zero componentwise on vectors:

$$\begin{bmatrix}1\\2\\3\end{bmatrix} + \begin{bmatrix}4\\5\\6\end{bmatrix} = \begin{bmatrix}5\\7\\9\end{bmatrix}$$

$$3\begin{bmatrix}1\\2\\3\end{bmatrix} = \begin{bmatrix}3\\6\\9\end{bmatrix}$$

$$\mathbf{0} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

More Generally

Define addition, scaling, zero componentwise on vectors:

$$\begin{bmatrix}1\\2\\3\end{bmatrix} + \begin{bmatrix}4\\5\\6\end{bmatrix} = \begin{bmatrix}5\\7\\9\end{bmatrix}$$

$$3\begin{bmatrix}1\\2\\3\end{bmatrix} = \begin{bmatrix}3\\6\\9\end{bmatrix}$$

$$\mathbf{0} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

Note: fix size, then do operations.

Useful Structure

Addition, Subtraction, $\mathbf{0}$, Scaling
commute
with applying matrices

For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$

$$A(\mathbf{x} + \mathbf{y}) = A\mathbf{x} + A\mathbf{y}$$
$$A(\lambda\mathbf{x}) = \lambda A\mathbf{x}$$
$$A\mathbf{0} = \mathbf{0}$$

Useful Structure

Operations

- Addition: $\mathbf{x} + \mathbf{y}$
- Scaling: $\lambda \mathbf{x}$
- Zero Vector: $\mathbf{0}$

Useful Structure

Operations

- Addition: $\mathbf{x} + \mathbf{y}$
- Scaling: $\lambda \mathbf{x}$
- Zero Vector: $\mathbf{0}$

Identities

Useful Structure

Operations

- Addition: $\mathbf{x} + \mathbf{y}$
- Scaling: $\lambda \mathbf{x}$
- Zero Vector: $\mathbf{0}$

Identities

Associative	$\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
Commutative	$\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
Identity	$\mathbf{v} + \mathbf{0} = \mathbf{v}$
Inverses	$\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
Distributivity	$\lambda(\mathbf{u} + \mathbf{v}) = \lambda\mathbf{u} + \lambda\mathbf{v}$
Compatibility (multiplication)	$\lambda(\mu\mathbf{v}) = (\lambda\mu)\mathbf{v}$
Compatibility (addition)	$(\lambda + \mu)\mathbf{v} = \lambda\mathbf{v} + \mu\mathbf{v}$
Compatibility (unit)	$1\mathbf{v} = \mathbf{v}$

Aide–Mémoire

Addition, Scaling, Zero behave just like in $\mathbb{R}$.

Some Special Vectors

Introducing. . .

$$\mathbf{e}_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \longleftarrow j\text{th}$$

Some Special Vectors

Introducing. . .

$$\mathbf{e}_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \longleftarrow j\text{th}$$

The elementary vectors.

Special Property One

Lemma

Every vector is a unique sum of scales of elementary vectors.

Special Property One

Lemma

Every vector is a unique sum of scales of elementary vectors.

Proof.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \cdots + x_n \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$= x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + \cdots + x_n\mathbf{e}_n$$

□

Examples

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

$$\begin{bmatrix} \pi \\ e \\ 0 \\ 2 \end{bmatrix} = \pi \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + e \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + 0 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Special Property Two

Lemma

A matrix is completely determined by what it does to the elementary vectors.

Special Property Two

Lemma

A matrix is completely determined by what it does to the elementary vectors.

Proof.

$$\begin{aligned} A\mathbf{x} &= A(x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + \cdots + x_n\mathbf{e}_n) \\ &= x_1A\mathbf{e}_1 + x_2A\mathbf{e}_2 + \cdots + x_nA\mathbf{e}_n \end{aligned}$$

□

Example

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \left(x \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$

$$= x \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= x \begin{bmatrix} 1 \\ 3 \end{bmatrix} + y \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

Summary

- Vectors add and scale like numbers
- Decompose arbitrary vectors using elementary vectors
- Understand matrices by applying to elementary vectors