Advanced Machine Learning

Maths is the Language of ML

$$\mathbf{Mx=b}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \\ 1 & 5 & 25 & 125 & 625 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 2 \\ -3 \\ 1 \end{pmatrix}$$

$$\mathbf{b=M^{-1}x}$$

$$\nabla_X \mathrm{Tr}(X^{-1}A) = -X^{-1}AX^{-1}$$

$$\mathbf{Mv}_i = \lambda_i \mathbf{v}_i$$

Course information, vectors, vector spaces, operators

Outline

1. **Course Details**
2. Vector Spaces
3. Operators

$\mathbf{Mx=b}$

$\mathbf{Mv}_i = \lambda_i \mathbf{v}_i$

$\mathbf{b=M^{-1}x}$

Advanced Machine Learning

- Machine learning has grown steadily since 1950's but is now mainstream
- Companies such as Google and Microsoft are fighting each other to get the best machine learning practitioners
- You should all have had a course covering the basics: learning from data, classification, regression, perceptrons, MLPs, etc.
- This course takes you one step further

Advanced Machine Learning

- Machine learning has grown steadily since 1950's but is now mainstream
- Companies such as Google and Microsoft are fighting each other to get the best machine learning practitioners
- You should all have had a course covering the basics: learning from data, classification, regression, perceptrons, MLPs, etc.
- This course takes you one step further

Advanced Machine Learning

- Machine learning has grown steadily since 1950's but is now mainstream
- Companies such as Google and Microsoft are fighting each other to get the best machine learning practitioners
- You should all have had a course covering the basics: learning from data, classification, regression, perceptrons, MLPs, etc.
- This course takes you one step further

Advanced Machine Learning

- Machine learning has grown steadily since 1950's but is now mainstream
- Companies such as Google and Microsoft are fighting each other to get the best machine learning practitioners
- You should all have had a course covering the basics: learning from data, classification, regression, perceptrons, MLPs, etc.
- This course takes you one step further

Course Outline

- The course is taught by myself
- We are going to cover around 10 advanced topics in 20 lectures
  - ⋆ 16:00-16:45 Monday: Building 45 room 2039 L/R B
  - ⋆ 9:00-9:45 Wednesday: Building 5 room 2017 L/T J
- You are going to do a projects in groups of around 4—supervised once a week by us
  - ⋆ 15:00-15:45 Thursday: Building 5 room 2017 L/T J

Course Outline

- The course is taught by myself
- We are going to cover around 10 advanced topics in 20 lectures
  - ⋆ 16:00-16:45 Monday: Building 45 room 2039 L/R B
  - ⋆ 9:00-9:45 Wednesday: Building 5 room 2017 L/T J
- You are going to do a projects in groups of around 4—supervised once a week by us
  - ⋆ 15:00-15:45 Thursday: Building 5 room 2017 L/T J

Course Outline

- The course is taught by myself
- We are going to cover around 10 advanced topics in 20 lectures
  - ⋆ 16:00-16:45 Monday: Building 45 room 2039 L/R B
  - ⋆ 9:00-9:45 Wednesday: Building 5 room 2017 L/T J
- You are going to do a projects in groups of around 4—supervised once a week by us
  - ⋆ 15:00-15:45 Thursday: Building 5 room 2017 L/T J

Topics Covered

- Vector spaces and linear algebra
- Generalisation
- Kernel methods
- Feature selection
- Deep learning
- Bayesian learning/MCMC/Graphical Models
- Matrix completion/sparse methods
- Ensemble methods/boosting
- Text/topical models and LDA

Topics Covered

- Vector spaces and linear algebra
- Generalisation
- Kernel methods
- Feature selection
- Deep learning
- Bayesian learning/MCMC/Graphical Models
- Matrix completion/sparse methods
- Ensemble methods/boosting
- Text/topical models and LDA

Topics Covered

- Vector spaces and linear algebra
- Generalisation
- Kernel methods
- Feature selection
- Deep learning
- Bayesian learning/MCMC/Graphical Models
- Matrix completion/sparse methods
- Ensemble methods/boosting
- Text/topical models and LDA

Topics Covered

- Vector spaces and linear algebra
- Generalisation
- Kernel methods
- Feature selection
- Deep learning
- Bayesian learning/MCMC/Graphical Models
- Matrix completion/sparse methods
- Ensemble methods/boosting
- Text/topical models and LDA

Topics Covered

- Vector spaces and linear algebra
- Generalisation
- Kernel methods
- Feature selection
- Deep learning
- Bayesian learning/MCMC/Graphical Models
- Matrix completion/sparse methods
- Ensemble methods/boosting
- Text/topical models and LDA

Topics Covered

- Vector spaces and linear algebra
- Generalisation
- Kernel methods
- Feature selection
- Deep learning
- Bayesian learning/MCMC/Graphical Models
- Matrix completion/sparse methods
- Ensemble methods/boosting
- Text/topical models and LDA

Topics Covered

- Vector spaces and linear algebra
- Generalisation
- Kernel methods
- Feature selection
- Deep learning
- Bayesian learning/MCMC/Graphical Models
- Matrix completion/sparse methods
- Ensemble methods/boosting
- Text/topical models and LDA

Topics Covered

- Vector spaces and linear algebra
- Generalisation
- Kernel methods
- Feature selection
- Deep learning
- Bayesian learning/MCMC/Graphical Models
- Matrix completion/sparse methods
- Ensemble methods/boosting
- Text/topical models and LDA

Topics Covered

- Vector spaces and linear algebra
- Generalisation
- Kernel methods
- Feature selection
- Deep learning
- Bayesian learning/MCMC/Graphical Models
- Matrix completion/sparse methods
- Ensemble methods/boosting
- Text/topical models and LDA

Outline

1. Course Details
2. **Vector Spaces**
3. Operators

$Mx=b$

$Mv_i=\lambda_i v_i$

$b=M^{-1}x$

Matrices, Vectors and All That

- The language of machine learning is mathematics
- Sometimes we draw pretty pictures to explain the mathematics
- Much of the mathematics we will use involves vectors, matrices and functions
- You need to master the language of mathematics, otherwise you won't understand the algorithms
- I'm going to spend this lecture and the next revising the mathematics you need to know

Matrices, Vectors and All That

- The language of machine learning is mathematics
- Sometimes we draw pretty pictures to explain the mathematics
- Much of the mathematics we will use involves vectors, matrices and functions
- You need to master the language of mathematics, otherwise you won't understand the algorithms
- I'm going to spend this lecture and the next revising the mathematics you need to know

Matrices, Vectors and All That

- The language of machine learning is mathematics
- Sometimes we draw pretty pictures to explain the mathematics
- Much of the mathematics we will use involves vectors, matrices and functions
- You need to master the language of mathematics, otherwise you won't understand the algorithms
- I'm going to spend this lecture and the next revising the mathematics you need to know

Matrices, Vectors and All That

- The language of machine learning is mathematics
- Sometimes we draw pretty pictures to explain the mathematics
- Much of the mathematics we will use involves vectors, matrices and functions
- You need to master the language of mathematics, otherwise you won't understand the algorithms
- I'm going to spend this lecture and the next revising the mathematics you need to know

Matrices, Vectors and All That

- The language of machine learning is mathematics
- Sometimes we draw pretty pictures to explain the mathematics
- Much of the mathematics we will use involves vectors, matrices and functions
- You need to master the language of mathematics, otherwise you won't understand the algorithms
- I'm going to spend this lecture and the next revising the mathematics you need to know

Matrices, Vectors and All That

- The language of machine learning is mathematics
- Sometimes we draw pretty pictures to explain the mathematics
- Much of the mathematics we will use involves vectors, matrices and functions
- You need to master the language of mathematics, otherwise you won't understand the algorithms
- I'm going to spend this lecture and the next revising the mathematics you need to know (but I'm going use a slightly posher language than you are probably used to)

Vectors

- We often work with objects with many components (features)
- To help handle this we will use vector notation

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

- We represent vectors by bold symbols
- All our vectors of column vectors by default
- We treat them as $n \times 1$ matrix

- We write row vectors as transposes of column vectors

$$\boldsymbol{y}^{\mathsf{T}} = (y_1, y_2, \ldots, y_n)$$

Vectors

- We often work with objects with many components (features)
- To help handle this we will use vector notation

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

- We represent vectors by bold symbols
- All our vectors of column vectors by default
- We treat them as $n \times 1$ matrix

- We write row vectors as transposes of column vectors

$$\boldsymbol{y}^{\mathsf{T}} = (y_1, y_2, \ldots, y_n)$$

Vectors

- We often work with objects with many components (features)
- To help handle this we will use vector notation

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

- We represent vectors by bold symbols
- All our vectors of column vectors by default
- We treat them as $n \times 1$ matrix

- We write row vectors as transposes of column vectors

$$\boldsymbol{y}^{\mathsf{T}} = (y_1, y_2, \ldots, y_n)$$

Vectors

- We often work with objects with many components (features)
- To help handle this we will use vector notation

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

- We represent vectors by bold symbols
- All our vectors of column vectors by default
- We treat them as $n \times 1$ matrix

- We write row vectors as transposes of column vectors

$$\boldsymbol{y}^{\mathsf{T}} = (y_1, y_2, \ldots, y_n)$$

Vectors

- We often work with objects with many components (features)
- To help handle this we will use vector notation

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

- We represent vectors by bold symbols
- All our vectors of column vectors by default
- We treat them as $n \times 1$ matrix

- We write row vectors as transposes of column vectors

$$\boldsymbol{y}^{\mathsf{T}} = (y_1, y_2, \ldots, y_n)$$

Vectors

- We often work with objects with many components (features)
- To help handle this we will use vector notation

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

- We represent vectors by bold symbols
- All our vectors of column vectors by default
- We treat them as $n \times 1$ matrix

- We write row vectors as transposes of column vectors

$$\boldsymbol{y}^{\mathsf{T}} = (y_1, y_2, \ldots, y_n)$$

Vectors

- We often work with objects with many components (features)
- To help handle this we will use vector notation

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

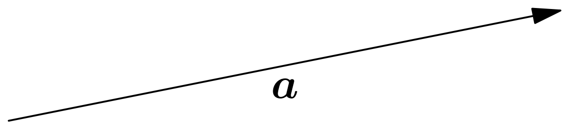
- We represent vectors by bold symbols
- All our vectors of column vectors by default
- We treat them as $n \times 1$ matrix

- We write row vectors as transposes of column vectors

$$\boldsymbol{y}^{\mathsf{T}} = (y_1, y_2, \ldots, y_n)$$

Basic Vector Operations

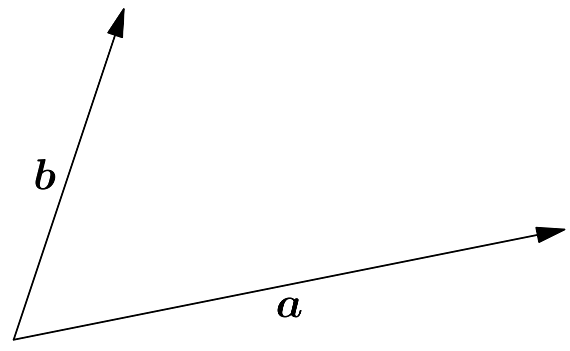
- The basic vector operations are adding



- multiplying by a scalar (a number)

Basic Vector Operations

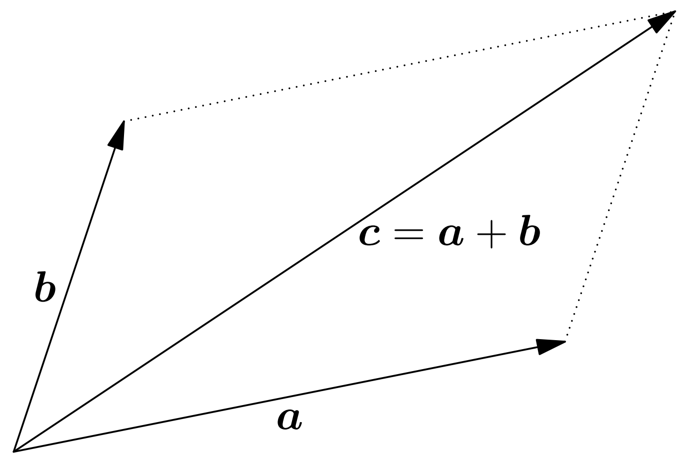
- The basic vector operations are adding



- multiplying by a scalar (a number)

Basic Vector Operations

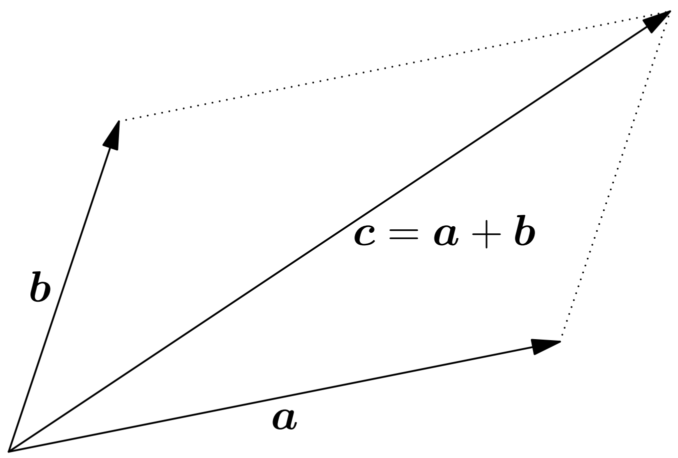
- The basic vector operations are adding



- multiplying by a scalar (a number)

Basic Vector Operations

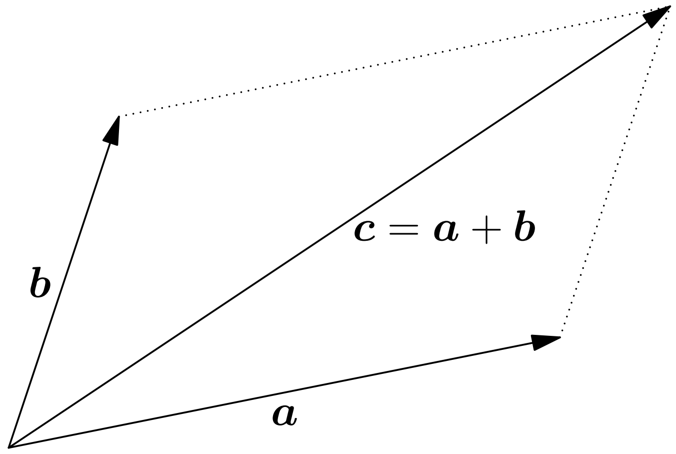
- The basic vector operations are adding



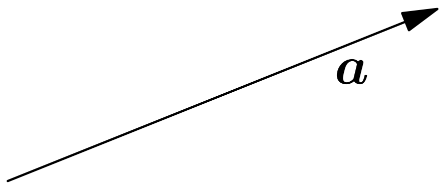
- multiplying by a scalar (a number)

Basic Vector Operations

- The basic vector operations are adding

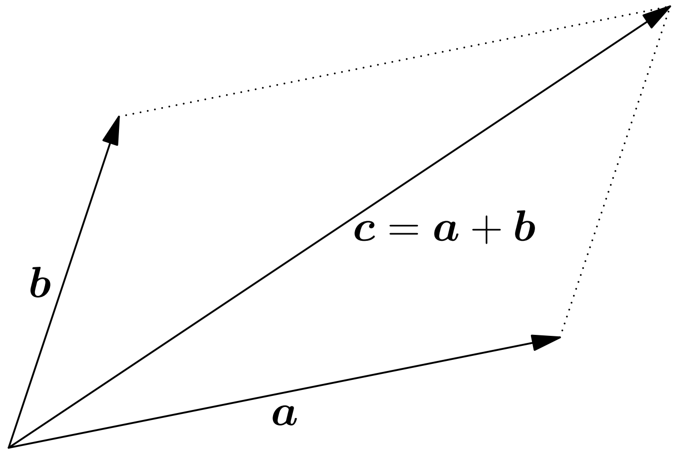


- multiplying by a scalar (a number)

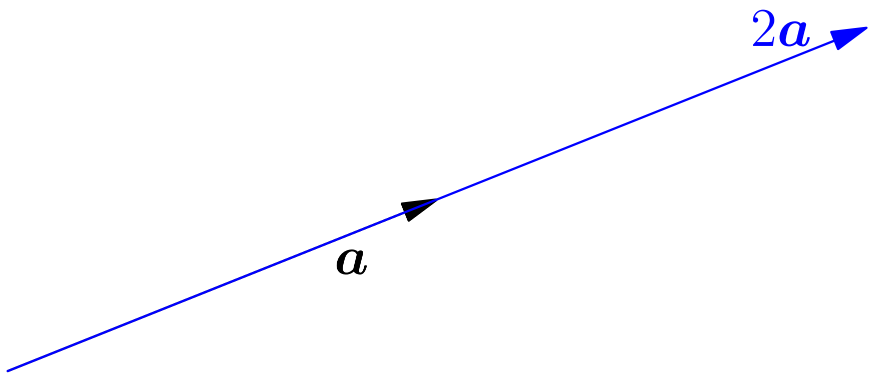


Basic Vector Operations

- The basic vector operations are adding



- multiplying by a scalar (a number)



Vector Space

- A vector space, $\mathcal{V}$, is a set of vectors which satisfies

  1. if $\boldsymbol{v}, \boldsymbol{w} \in \mathcal{V}$ then $a\,\boldsymbol{v} \in \mathcal{V}$ and $\boldsymbol{v} + \boldsymbol{w} \in \mathcal{V}$ (closure)
  2. $\boldsymbol{v} + \boldsymbol{w} = \boldsymbol{w} + \boldsymbol{v}$ (commutativity of addition)
  3. $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$ (associativity of addition)
  4. $\boldsymbol{v} + \mathbf{0} = \boldsymbol{v}$ (existence of additive identity $\mathbf{0}$)
  5. $1\,\boldsymbol{v} = \boldsymbol{v}$ (existence of multiplicative identity 1)
  6. $a\,(b\,\boldsymbol{v}) = (a\,b)\,\boldsymbol{v}$ (distributive properties)
  7. $a\,(\boldsymbol{v} + \boldsymbol{w}) = a\,\boldsymbol{v} + a\,\boldsymbol{w}$
  8. $(a + b)\,\boldsymbol{v} = a\,\boldsymbol{v} + b\,\boldsymbol{v}$

  (You don't need to remember these)

- Just from these properties we can deduce other properties

Vector Space

- A vector space, $\mathcal{V}$, is a set of vectors which satisfies

  1. if $\boldsymbol{v}, \boldsymbol{w} \in \mathcal{V}$ then $a\,\boldsymbol{v} \in \mathcal{V}$ and $\boldsymbol{v} + \boldsymbol{w} \in \mathcal{V}$ (closure)
  2. $\boldsymbol{v} + \boldsymbol{w} = \boldsymbol{w} + \boldsymbol{v}$ (commutativity of addition)
  3. $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$ (associativity of addition)
  4. $\boldsymbol{v} + \mathbf{0} = \boldsymbol{v}$ (existence of additive identity $\mathbf{0}$)
  5. $1\,\boldsymbol{v} = \boldsymbol{v}$ (existence of multiplicative identity 1)
  6. $a\,(b\,\boldsymbol{v}) = (a\,b)\,\boldsymbol{v}$ (distributive properties)
  7. $a\,(\boldsymbol{v} + \boldsymbol{w}) = a\,\boldsymbol{v} + a\,\boldsymbol{w}$
  8. $(a + b)\,\boldsymbol{v} = a\,\boldsymbol{v} + b\,\boldsymbol{v}$

  (You don't need to remember these)

- Just from these properties we can deduce other properties

Vector Space

- A vector space, $\mathcal{V}$, is a set of vectors which satisfies

  1. if $\boldsymbol{v}, \boldsymbol{w} \in \mathcal{V}$ then $a\,\boldsymbol{v} \in \mathcal{V}$ and $\boldsymbol{v} + \boldsymbol{w} \in \mathcal{V}$ (closure)
  2. $\boldsymbol{v} + \boldsymbol{w} = \boldsymbol{w} + \boldsymbol{v}$ (commutativity of addition)
  3. $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$ (associativity of addition)
  4. $\boldsymbol{v} + \boldsymbol{0} = \boldsymbol{v}$ (existence of additive identity $\boldsymbol{0}$)
  5. $1\,\boldsymbol{v} = \boldsymbol{v}$ (existence of multiplicative identity 1)
  6. $a\,(b\,\boldsymbol{v}) = (a\,b)\,\boldsymbol{v}$ (distributive properties)
  7. $a\,(\boldsymbol{v} + \boldsymbol{w}) = a\,\boldsymbol{v} + a\,\boldsymbol{w}$
  8. $(a + b)\,\boldsymbol{v} = a\,\boldsymbol{v} + b\,\boldsymbol{v}$

  (You don't need to remember these)

- Just from these properties we can deduce other properties

Vector Space

- A vector space, $\mathcal{V}$, is a set of vectors which satisfies

  1. if $\boldsymbol{v}, \boldsymbol{w} \in \mathcal{V}$ then $a\,\boldsymbol{v} \in \mathcal{V}$ and $\boldsymbol{v} + \boldsymbol{w} \in \mathcal{V}$ (closure)
  2. $\boldsymbol{v} + \boldsymbol{w} = \boldsymbol{w} + \boldsymbol{v}$ (commutativity of addition)
  3. $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$ (associativity of addition)
  4. $\boldsymbol{v} + \boldsymbol{0} = \boldsymbol{v}$ (existence of additive identity $\boldsymbol{0}$)
  5. $1\,\boldsymbol{v} = \boldsymbol{v}$ (existence of multiplicative identity 1)
  6. $a\,(b\,\boldsymbol{v}) = (a\,b)\,\boldsymbol{v}$ (distributive properties)
  7. $a\,(\boldsymbol{v} + \boldsymbol{w}) = a\,\boldsymbol{v} + a\,\boldsymbol{w}$
  8. $(a + b)\,\boldsymbol{v} = a\,\boldsymbol{v} + b\,\boldsymbol{v}$

  (You don't need to remember these)

- Just from these properties we can deduce other properties

Vector Space

- A vector space, $\mathcal{V}$, is a set of vectors which satisfies

  1. if $\boldsymbol{v}, \boldsymbol{w} \in \mathcal{V}$ then $a\,\boldsymbol{v} \in \mathcal{V}$ and $\boldsymbol{v} + \boldsymbol{w} \in \mathcal{V}$ (closure)
  2. $\boldsymbol{v} + \boldsymbol{w} = \boldsymbol{w} + \boldsymbol{v}$ (commutativity of addition)
  3. $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$ (associativity of addition)
  4. $\boldsymbol{v} + \boldsymbol{0} = \boldsymbol{v}$ (existence of additive identity $\boldsymbol{0}$)
  5. $1\,\boldsymbol{v} = \boldsymbol{v}$ (existence of multiplicative identity 1)
  6. $a\,(b\,\boldsymbol{v}) = (a\,b)\,\boldsymbol{v}$ (distributive properties)
  7. $a\,(\boldsymbol{v} + \boldsymbol{w}) = a\,\boldsymbol{v} + a\,\boldsymbol{w}$
  8. $(a + b)\,\boldsymbol{v} = a\,\boldsymbol{v} + b\,\boldsymbol{v}$

  (You don't need to remember these)

- Just from these properties we can deduce other properties

Vector Space

- A vector space, $\mathcal{V}$, is a set of vectors which satisfies

1. if $\boldsymbol{v}, \boldsymbol{w} \in \mathcal{V}$ then $a\,\boldsymbol{v} \in \mathcal{V}$ and $\boldsymbol{v} + \boldsymbol{w} \in \mathcal{V}$ (closure)
2. $\boldsymbol{v} + \boldsymbol{w} = \boldsymbol{w} + \boldsymbol{v}$ (commutativity of addition)
3. $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$ (associativity of addition)
4. $\boldsymbol{v} + \mathbf{0} = \boldsymbol{v}$ (existence of additive identity $\mathbf{0}$)
5. $1\,\boldsymbol{v} = \boldsymbol{v}$ (existence of multiplicative identity 1)
6. $a\,(b\,\boldsymbol{v}) = (a\,b)\,\boldsymbol{v}$ (distributive properties)
7. $a\,(\boldsymbol{v} + \boldsymbol{w}) = a\,\boldsymbol{v} + a\,\boldsymbol{w}$
8. $(a + b)\,\boldsymbol{v} = a\,\boldsymbol{v} + b\,\boldsymbol{v}$

(You don't need to remember these)

- Just from these properties we can deduce other properties

Vector Space

- A vector space, $\mathcal{V}$, is a set of vectors which satisfies

  1. if $\boldsymbol{v}, \boldsymbol{w} \in \mathcal{V}$ then $a\,\boldsymbol{v} \in \mathcal{V}$ and $\boldsymbol{v} + \boldsymbol{w} \in \mathcal{V}$ (closure)
  2. $\boldsymbol{v} + \boldsymbol{w} = \boldsymbol{w} + \boldsymbol{v}$ (commutativity of addition)
  3. $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$ (associativity of addition)
  4. $\boldsymbol{v} + \boldsymbol{0} = \boldsymbol{v}$ (existence of additive identity $\boldsymbol{0}$)
  5. $1\,\boldsymbol{v} = \boldsymbol{v}$ (existence of multiplicative identity 1)
  6. $a\,(b\,\boldsymbol{v}) = (a\,b)\,\boldsymbol{v}$ (distributive properties)
  7. $a\,(\boldsymbol{v} + \boldsymbol{w}) = a\,\boldsymbol{v} + a\,\boldsymbol{w}$
  8. $(a + b)\,\boldsymbol{v} = a\,\boldsymbol{v} + b\,\boldsymbol{v}$

  (You don't need to remember these)

- Just from these properties we can deduce other properties

Vector Space

- A vector space, $\mathcal{V}$, is a set of vectors which satisfies

  1. if $\boldsymbol{v}, \boldsymbol{w} \in \mathcal{V}$ then $a\,\boldsymbol{v} \in \mathcal{V}$ and $\boldsymbol{v} + \boldsymbol{w} \in \mathcal{V}$ (closure)
  2. $\boldsymbol{v} + \boldsymbol{w} = \boldsymbol{w} + \boldsymbol{v}$ (commutativity of addition)
  3. $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$ (associativity of addition)
  4. $\boldsymbol{v} + \mathbf{0} = \boldsymbol{v}$ (existence of additive identity $\mathbf{0}$)
  5. $1\,\boldsymbol{v} = \boldsymbol{v}$ (existence of multiplicative identity 1)
  6. $a\,(b\,\boldsymbol{v}) = (a\,b)\,\boldsymbol{v}$ (distributive properties)
  7. $a\,(\boldsymbol{v} + \boldsymbol{w}) = a\,\boldsymbol{v} + a\,\boldsymbol{w}$
  8. $(a + b)\,\boldsymbol{v} = a\,\boldsymbol{v} + b\,\boldsymbol{v}$

  (You don't need to remember these)

- Just from these properties we can deduce other properties

$\mathbb{R}^n$

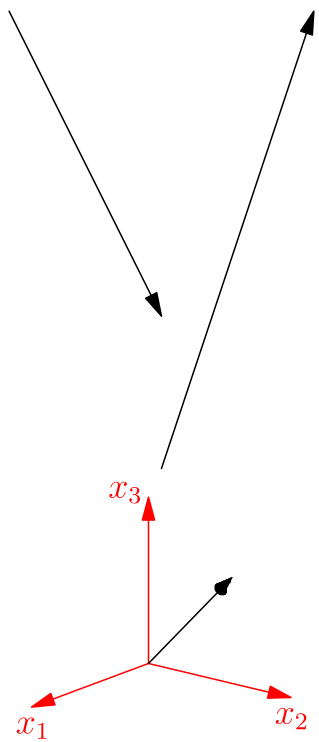
- When we first learn about vectors we think of them arrows in 3-D space

- If we centre them all at the origin then there is a one-to-one correspondence between vectors and points in space

- We call this vector space $\mathbb{R}^3$

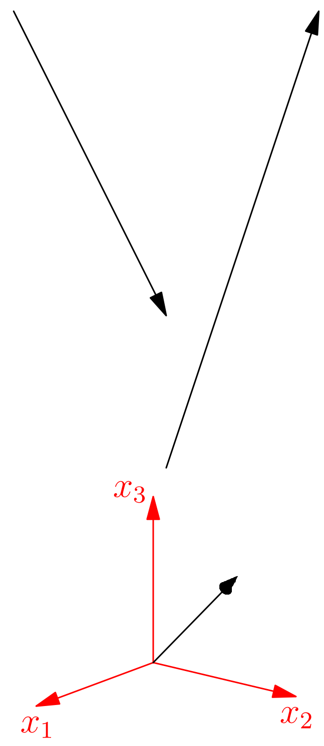
- Any set of quantities $\boldsymbol{x} = (x_1, x_2, \dots, x_n)^\mathsf{T}$ which satisfy the axioms above form a vector space $\mathbb{R}^n$

- Of course, we can't so easily draw pictures of high-dimensional vectors



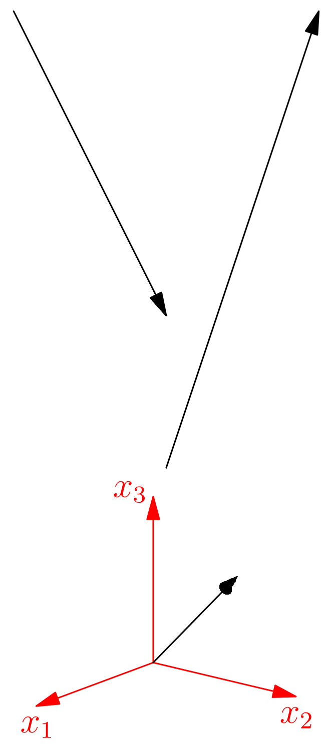
$\mathbb{R}^n$

- When we first learn about vectors we think of them arrows in 3-D space
- If we centre them all at the origin then there is a one-to-one correspondence between vectors and points in space
- We call this vector space $\mathbb{R}^3$
- Any set of quantities $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\mathsf{T}$ which satisfy the axioms above form a vector space $\mathbb{R}^n$
- Of course, we can't so easily draw pictures of high-dimensional vectors



$\mathbb{R}^n$

- When we first learn about vectors we think of them arrows in 3-D space
- If we centre them all at the origin then there is a one-to-one correspondence between vectors and points in space
- We call this vector space $\mathbb{R}^3$
- Any set of quantities $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\mathsf{T}$ which satisfy the axioms above form a vector space $\mathbb{R}^n$
- Of course, we can't so easily draw pictures of high-dimensional vectors



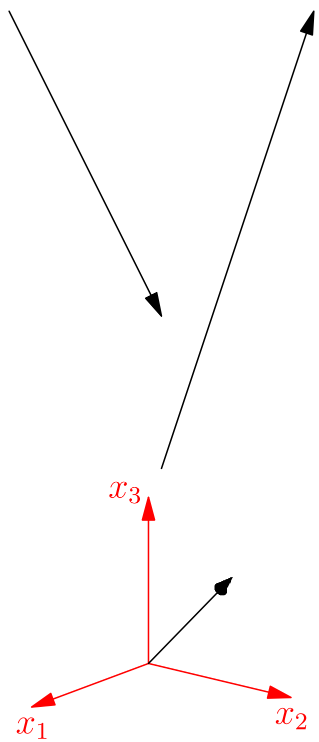
- When we first learn about vectors we think of them arrows in 3-D space

- If we centre them all at the origin then there is a one-to-one correspondence between vectors and points in space

- We call this vector space $\mathbb{R}^3$

- Any set of quantities $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\mathsf{T}$ which satisfy the axioms above form a vector space $\mathbb{R}^n$

- Of course, we can't so easily draw pictures of high-dimensional vectors



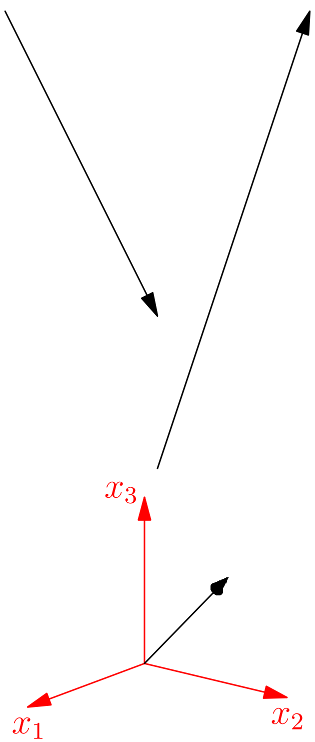
- When we first learn about vectors we think of them arrows in 3-D space

- If we centre them all at the origin then there is a one-to-one correspondence between vectors and points in space

- We call this vector space $\mathbb{R}^3$

- Any set of quantities $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\mathsf{T}$ which satisfy the axioms above form a vector space $\mathbb{R}^n$

- Of course, we can't so easily draw pictures of high-dimensional vectors



Other Vector Spaces

- Vectors are not the only object that form vector spaces
- Matrices satisfy all the conditions of a vector space
- Infinite sequences form a vector space
- Functions form a vector space
  - ⋆ Let $C(a, b)$ be the set of functions defined on the interval $[a, b]$
  - ⋆ Note that if $f(x), g(x) \in C(a, b)$ then $a\, f(x) \in C(a, b)$ and $f(x) + g(x) \in C(a, b)$
- Bounded vectors **don't** form a vector space

Other Vector Spaces

- Vectors are not the only object that form vector spaces
- Matrices satisfy all the conditions of a vector space
- Infinite sequences form a vector space
- Functions form a vector space
  - ⋆ Let $C(a, b)$ be the set of functions defined on the interval $[a, b]$
  - ⋆ Note that if $f(x), g(x) \in C(a, b)$ then $a\, f(x) \in C(a, b)$ and $f(x) + g(x) \in C(a, b)$
- Bounded vectors **don't** form a vector space

Other Vector Spaces

- Vectors are not the only object that form vector spaces
- Matrices satisfy all the conditions of a vector space
- Infinite sequences form a vector space
- Functions form a vector space
  - ⋆ Let $C(a, b)$ be the set of functions defined on the interval $[a, b]$
  - ⋆ Note that if $f(x), g(x) \in C(a, b)$ then $a\, f(x) \in C(a, b)$ and $f(x) + g(x) \in C(a, b)$
- Bounded vectors **don't** form a vector space

Other Vector Spaces

- Vectors are not the only object that form vector spaces
- Matrices satisfy all the conditions of a vector space
- Infinite sequences form a vector space
- Functions form a vector space
  - ⋆ Let $C(a,b)$ be the set of functions defined on the interval $[a,b]$
  - ⋆ Note that if $f(x), g(x) \in C(a,b)$ then $a\, f(x) \in C(a,b)$ and $f(x) + g(x) \in C(a,b)$
- Bounded vectors **don't** form a vector space

Other Vector Spaces

- Vectors are not the only object that form vector spaces
- Matrices satisfy all the conditions of a vector space
- Infinite sequences form a vector space
- Functions form a vector space
  - ⋆ Let $C(a, b)$ be the set of functions defined on the interval $[a, b]$
  - ⋆ Note that if $f(x), g(x) \in C(a, b)$ then $a\, f(x) \in C(a, b)$ and $f(x) + g(x) \in C(a, b)$
- Bounded vectors **don't** form a vector space

Other Vector Spaces

- Vectors are not the only object that form vector spaces
- Matrices satisfy all the conditions of a vector space
- Infinite sequences form a vector space
- Functions form a vector space
  - ⋆ Let $C(a, b)$ be the set of functions defined on the interval $[a, b]$
  - ⋆ Note that if $f(x), g(x) \in C(a, b)$ then $a\, f(x) \in C(a, b)$ and $f(x) + g(x) \in C(a, b)$
- Bounded vectors **don't** form a vector space

Metrics

- Vector spaces become more interesting if we have a notion of distance
- We say $d(\boldsymbol{x}, \boldsymbol{y})$ is a proper distance or **metric** if

   1. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non-negativity)
   2. $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ iff $\boldsymbol{x} = \boldsymbol{y}$ (identity of indiscernibles)
   3. $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (symmetry)
   4. $d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{z}) + d(\boldsymbol{z}, \boldsymbol{y})$ (triangular inequality)

- There are typically many possible distances (e.g. Euclidean distance, Manhattan distance, etc.)
- Often one or more condition isn't satisfied then we have a **pseudo-metric**

Metrics

- Vector spaces become more interesting if we have a notion of distance
- We say $d(\boldsymbol{x}, \boldsymbol{y})$ is a proper distance or **metric** if

  1. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non-negativity)
  2. $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ iff $\boldsymbol{x} = \boldsymbol{y}$ (identity of indiscernibles)
  3. $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (symmetry)
  4. $d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{z}) + d(\boldsymbol{z}, \boldsymbol{y})$ (triangular inequality)

- There are typically many possible distances (e.g. Euclidean distance, Manhattan distance, etc.)
- Often one or more condition isn't satisfied then we have a **pseudo-metric**

Metrics

- Vector spaces become more interesting if we have a notion of distance

- We say $d(\boldsymbol{x}, \boldsymbol{y})$ is a proper distance or **metric** if

  1. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non-negativity)
  2. $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ iff $\boldsymbol{x} = \boldsymbol{y}$ (identity of indiscernibles)
  3. $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (symmetry)
  4. $d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{z}) + d(\boldsymbol{z}, \boldsymbol{y})$ (triangular inequality)

- There are typically many possible distances (e.g. Euclidean distance, Manhattan distance, etc.)

- Often one or more condition isn't satisfied then we have a **pseudo-metric**

Metrics

- Vector spaces become more interesting if we have a notion of distance

- We say $d(\boldsymbol{x}, \boldsymbol{y})$ is a proper distance or **metric** if

  1. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non-negativity)
  2. $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ iff $\boldsymbol{x} = \boldsymbol{y}$ (identity of indiscernibles)
  3. $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (symmetry)
  4. $d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{z}) + d(\boldsymbol{z}, \boldsymbol{y})$ (triangular inequality)

- There are typically many possible distances (e.g. Euclidean distance, Manhattan distance, etc.)

- Often one or more condition isn't satisfied then we have a **pseudo-metric**

Metrics

- Vector spaces become more interesting if we have a notion of distance
- We say $d(\boldsymbol{x}, \boldsymbol{y})$ is a proper distance or **metric** if

  1. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non-negativity)
  2. $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ iff $\boldsymbol{x} = \boldsymbol{y}$ (identity of indiscernibles)
  3. $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (symmetry)
  4. $d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{z}) + d(\boldsymbol{z}, \boldsymbol{y})$ (triangular inequality)

- There are typically many possible distances (e.g. Euclidean distance, Manhattan distance, etc.)
- Often one or more condition isn't satisfied then we have a **pseudo-metric**

Metrics

- Vector spaces become more interesting if we have a notion of distance
- We say $d(\boldsymbol{x}, \boldsymbol{y})$ is a proper distance or **metric** if

  1. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non-negativity)
  2. $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ iff $\boldsymbol{x} = \boldsymbol{y}$ (identity of indiscernibles)
  3. $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (symmetry)
  4. $d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{z}) + d(\boldsymbol{z}, \boldsymbol{y})$ (triangular inequality)

- There are typically many possible distances (e.g. Euclidean distance, Manhattan distance, etc.)
- Often one or more condition isn't satisfied then we have a **pseudo-metric**

Metrics

- Vector spaces become more interesting if we have a notion of distance
- We say $d(\boldsymbol{x}, \boldsymbol{y})$ is a proper distance or **metric** if

1. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non-negativity)
2. $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ iff $\boldsymbol{x} = \boldsymbol{y}$ (identity of indiscernibles)
3. $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (symmetry)
4. $d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{z}) + d(\boldsymbol{z}, \boldsymbol{y})$ (triangular inequality)

- There are typically many possible distances (e.g. Euclidean distance, Manhattan distance, etc.)
- Often one or more condition isn't satisfied then we have a **pseudo-metric**

Norms

- Vector spaces are even more interesting with a notion of length
- **Norms** provide some measure of the size of a vector
- To formalise this we define the **norm** of an object $\boldsymbol{v}$ as $\|\boldsymbol{v}\|$ satisfying

1. $\|\boldsymbol{v}\| > 0$ if $\boldsymbol{v} \neq \mathbf{0}$
2. $\|a\,\boldsymbol{v}\| = a\|\boldsymbol{v}\|$
3. $\|\boldsymbol{u} + \boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$ (triangular inequality)

- When some criteria aren't satisfied we have a **pseudo-norms**
- Norms provide a metric $d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|$ (they are metric spaces)

Norms

- Vector spaces are even more interesting with a notion of length
- **Norms** provide some measure of the size of a vector
- To formalise this we define the **norm** of an object $\boldsymbol{v}$ as $\|\boldsymbol{v}\|$ satisfying

1. $\|\boldsymbol{v}\| > 0$ if $\boldsymbol{v} \neq \mathbf{0}$
2. $\|a\,\boldsymbol{v}\| = a\|\boldsymbol{v}\|$
3. $\|\boldsymbol{u}+\boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$ (triangular inequality)

- When some criteria aren't satisfied we have a **pseudo-norms**
- Norms provide a metric $d(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|$ (they are metric spaces)

Norms

- Vector spaces are even more interesting with a notion of length
- **Norms** provide some measure of the size of a vector
- To formalise this we define the **norm** of an object $\boldsymbol{v}$ as $\|\boldsymbol{v}\|$ satisfying
  1. $\|\boldsymbol{v}\| > 0$ if $\boldsymbol{v} \neq \mathbf{0}$
  2. $\|a\,\boldsymbol{v}\| = a\|\boldsymbol{v}\|$
  3. $\|\boldsymbol{u}+\boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$ (triangular inequality)
- When some criteria aren't satisfied we have a **pseudo-norms**
- Norms provide a metric $d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|$ (they are metric spaces)

Norms

- Vector spaces are even more interesting with a notion of length
- **Norms** provide some measure of the size of a vector
- To formalise this we define the **norm** of an object $\boldsymbol{v}$ as $\|\boldsymbol{v}\|$ satisfying
  1. $\|\boldsymbol{v}\| > 0$ if $\boldsymbol{v} \neq \mathbf{0}$
  2. $\|a\,\boldsymbol{v}\| = a\|\boldsymbol{v}\|$
  3. $\|\boldsymbol{u}+\boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$ (triangular inequality)
- When some criteria aren't satisfied we have a **pseudo-norms**
- Norms provide a metric $d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|$ (they are metric spaces)

Norms

- Vector spaces are even more interesting with a notion of length
- **Norms** provide some measure of the size of a vector
- To formalise this we define the **norm** of an object $\boldsymbol{v}$ as $\|\boldsymbol{v}\|$ satisfying
  1. $\|\boldsymbol{v}\| > 0$ if $\boldsymbol{v} \neq \mathbf{0}$
  2. $\|a\,\boldsymbol{v}\| = a\|\boldsymbol{v}\|$
  3. $\|\boldsymbol{u} + \boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$ (triangular inequality)
- When some criteria aren't satisfied we have a **pseudo-norms**
- Norms provide a metric $d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|$ (they are metric spaces)

Norms

- Vector spaces are even more interesting with a notion of length
- **Norms** provide some measure of the size of a vector
- To formalise this we define the **norm** of an object $\boldsymbol{v}$ as $\|\boldsymbol{v}\|$ satisfying

  1. $\|\boldsymbol{v}\| > 0$ if $\boldsymbol{v} \neq \boldsymbol{0}$
  2. $\|a\,\boldsymbol{v}\| = a\|\boldsymbol{v}\|$
  3. $\|\boldsymbol{u}+\boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$ (triangular inequality)

- When some criteria aren't satisfied we have a **pseudo-norms**
- Norms provide a metric $d(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|$ (they are metric spaces)

Norms

- Vector spaces are even more interesting with a notion of length
- **Norms** provide some measure of the size of a vector
- To formalise this we define the **norm** of an object $\boldsymbol{v}$ as $\|\boldsymbol{v}\|$ satisfying
  1. $\|\boldsymbol{v}\| > 0$ if $\boldsymbol{v} \neq \mathbf{0}$
  2. $\|a\,\boldsymbol{v}\| = a\|\boldsymbol{v}\|$
  3. $\|\boldsymbol{u} + \boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$ (triangular inequality)
- When some criteria aren't satisfied we have a **pseudo-norms**
- Norms provide a metric $d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|$ (they are metric spaces)

Vector Norms

- The familiar vector norm is the (Euclidean) two norm

$$\|\boldsymbol{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

- Other norms exist, such as the p-norm

$$\|\boldsymbol{v}\|_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p}$$

- Other special cases include the 1-norm and the infinite norm

$$\|\boldsymbol{v}\|_1 = \sum_{i=1}^{n} |v_i| \qquad\qquad \|\boldsymbol{v}\|_\infty = \max_i |v_i|$$

- The 0-norm is a pseudo-norm as it does not satisfy condition 2

$$\|\boldsymbol{v}\|_0 = \text{number of non-zero components}$$

Vector Norms

- The familiar vector norm is the (Euclidean) two norm

$$\|\boldsymbol{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

- Other norms exist, such as the p-norm

$$\|\boldsymbol{v}\|_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p}$$

- Other special cases include the 1-norm and the infinite norm

$$\|\boldsymbol{v}\|_1 = \sum_{i=1}^{n} |v_i| \qquad\qquad \|\boldsymbol{v}\|_\infty = \max_i |v_i|$$

- The 0-norm is a pseudo-norm as it does not satisfy condition 2

$$\|\boldsymbol{v}\|_0 = \text{number of non-zero components}$$

Vector Norms

- The familiar vector norm is the (Euclidean) two norm

$$\|\boldsymbol{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

- Other norms exist, such as the p-norm

$$\|\boldsymbol{v}\|_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p}$$

- Other special cases include the 1-norm and the infinite norm

$$\|\boldsymbol{v}\|_1 = \sum_{i=1}^{n} |v_i| \qquad\qquad \|\boldsymbol{v}\|_\infty = \max_i |v_i|$$

- The 0-norm is a pseudo-norm as it does not satisfy condition 2

$$\|\boldsymbol{v}\|_0 = \text{number of non-zero components}$$

Vector Norms

- The familiar vector norm is the (Euclidean) two norm

$$\|\boldsymbol{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

- Other norms exist, such as the p-norm

$$\|\boldsymbol{v}\|_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p}$$

- Other special cases include the 1-norm and the infinite norm

$$\|\boldsymbol{v}\|_1 = \sum_{i=1}^{n} |v_i| \qquad \|\boldsymbol{v}\|_\infty = \max_i |v_i|$$

- The 0-norm is a pseudo-norm as it does not satisfy condition 2

$$\|\boldsymbol{v}\|_0 = \text{number of non-zero components}$$

Vector Norms

- The familiar vector norm is the (Euclidean) two norm

$$\|\boldsymbol{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

- Other norms exist, such as the p-norm

$$\|\boldsymbol{v}\|_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p}$$

- Other special cases include the 1-norm and the infinite norm

$$\|\boldsymbol{v}\|_1 = \sum_{i=1}^{n} |v_i| \qquad\qquad \|\boldsymbol{v}\|_\infty = \max_i |v_i|$$

- The 0-norm is a pseudo-norm as it does not satisfy condition 2

$$\|\boldsymbol{v}\|_0 = \text{number of non-zero components}$$

Matrix Norms

- We can define norms for other objects
- The norm of a matrix encodes how large the mapping is
- The Frobenius norm is defined by

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |A_{ij}|^2}$$

- Many other norms exist including 1-norm, max-norm, etc.
- For square matrices, some, but not all, norms satisfy the another inequality

$$\|\mathbf{A}\,\mathbf{B}\| \leq \|\mathbf{A}\| \times \|\mathbf{B}\|$$

Matrix Norms

- We can define norms for other objects
- The norm of a matrix encodes how large the mapping is
- The Frobenius norm is defined by

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |A_{ij}|^2}$$

- Many other norms exist including 1-norm, max-norm, etc.
- For square matrices, some, but not all, norms satisfy the another inequality

$$\|\mathbf{A}\,\mathbf{B}\| \leq \|\mathbf{A}\| \times \|\mathbf{B}\|$$

Matrix Norms

- We can define norms for other objects
- The norm of a matrix encodes how large the mapping is
- The Frobenius norm is defined by

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|A_{ij}|^2}$$

- Many other norms exist including 1-norm, max-norm, etc.
- For square matrices, some, but not all, norms satisfy the another inequality

$$\|\mathbf{A}\,\mathbf{B}\| \leq \|\mathbf{A}\| \times \|\mathbf{B}\|$$

Matrix Norms

- We can define norms for other objects
- The norm of a matrix encodes how large the mapping is
- The Frobenius norm is defined by

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |A_{ij}|^2}$$

- Many other norms exist including 1-norm, max-norm, etc.
- For square matrices, some, but not all, norms satisfy the another inequality

$$\|\mathbf{A}\,\mathbf{B}\| \leq \|\mathbf{A}\| \times \|\mathbf{B}\|$$

Matrix Norms

- We can define norms for other objects
- The norm of a matrix encodes how large the mapping is
- The Frobenius norm is defined by

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |A_{ij}|^2}$$

- Many other norms exist including 1-norm, max-norm, etc.
- For square matrices, some, but not all, norms satisfy the another inequality

$$\|\mathbf{A}\,\mathbf{B}\| \leq \|\mathbf{A}\| \times \|\mathbf{B}\|$$

Compatible Norms

- A vector and matrix norm are said to be compatible if

$$\|\boldsymbol{M}\boldsymbol{v}\|_b \leq \|\boldsymbol{M}\|_a \times \|\boldsymbol{v}\|_b$$

  (Frobenius and Euclidean norms are compatible)

- Norms provide quick ways to bound the maximum growth of a vector under a mapping induced by the matrix

- We will see that a measure of the sensitivity of a mapping is in terms of the ratio of its maximum effect to its minimum effect on a vector

- This is known as the **conditioning**, given by $\|\mathbf{M}\| \times \|\mathbf{M}^{-1}\|$

Compatible Norms

- A vector and matrix norm are said to be compatible if

$$\|\boldsymbol{M}\boldsymbol{v}\|_b \leq \|\boldsymbol{M}\|_a \times \|\boldsymbol{v}\|_b$$

  (Frobenius and Euclidean norms are compatible)

- Norms provide quick ways to bound the maximum growth of a vector under a mapping induced by the matrix

- We will see that a measure of the sensitivity of a mapping is in terms of the ratio of its maximum effect to its minimum effect on a vector

- This is known as the **conditioning**, given by $\|\boldsymbol{M}\| \times \|\boldsymbol{M}^{-1}\|$

Compatible Norms

- A vector and matrix norm are said to be compatible if

$$\|\boldsymbol{M}\boldsymbol{v}\|_b \leq \|\boldsymbol{M}\|_a \times \|\boldsymbol{v}\|_b$$

  (Frobenius and Euclidean norms are compatible)

- Norms provide quick ways to bound the maximum growth of a vector under a mapping induced by the matrix

- We will see that a measure of the sensitivity of a mapping is in terms of the ratio of its maximum effect to its minimum effect on a vector

- This is known as the **conditioning**, given by $\|\boldsymbol{M}\| \times \|\boldsymbol{M}^{-1}\|$

Compatible Norms

- A vector and matrix norm are said to be compatible if

$$\|\boldsymbol{M}\boldsymbol{v}\|_b \leq \|\boldsymbol{M}\|_a \times \|\boldsymbol{v}\|_b$$

  (Frobenius and Euclidean norms are compatible)

- Norms provide quick ways to bound the maximum growth of a vector under a mapping induced by the matrix

- We will see that a measure of the sensitivity of a mapping is in terms of the ratio of its maximum effect to its minimum effect on a vector

- This is known as the **conditioning**, given by $\|\boldsymbol{M}\| \times \|\boldsymbol{M}^{-1}\|$

Function Norms

- Functions can also have norms, for example, if $f(x)$ is defined in some interval $\mathcal{I}$

$$\|f\|_{L_2} = \sqrt{\int_{x\in\mathcal{I}} f^2(x)\,\mathrm{d}x}$$

- The L_2 vector space is the set of function where $\|f\|_{L_2} < \infty$

- The L_1-norm is given by $\|f\|_{L_1} = \int_{x\in\mathcal{I}} |f(x)|\,\mathrm{d}x$

- The infinite-norm is given by $\|f\|_{\infty} = \max_{x\in\mathcal{I}} |f(x)|$

Function Norms

- Functions can also have norms, for example, if $f(x)$ is defined in some interval $\mathcal{I}$

$$\|f\|_{L_2} = \sqrt{\int_{x\in\mathcal{I}} f^2(x)\,\mathrm{d}x}$$

- The L_2 vector space is the set of function where $\|f\|_{L_2} < \infty$

- The L_1-norm is given by $\|f\|_{L_1} = \int_{x\in\mathcal{I}} |f(x)|\,\mathrm{d}x$

- The infinite-norm is given by $\|f\|_\infty = \max_{x\in\mathcal{I}} |f(x)|$

Function Norms

- Functions can also have norms, for example, if $f(x)$ is defined in some interval $\mathcal{I}$

$$\|f\|_{L_2} = \sqrt{\int_{x\in\mathcal{I}} f^2(x)\,\mathrm{d}x}$$

- The L_2 vector space is the set of function where $\|f\|_{L_2} < \infty$
- The L_1-norm is given by $\|f\|_{L_1} = \int_{x\in\mathcal{I}} |f(x)|\,\mathrm{d}x$
- The infinite-norm is given by $\|f\|_\infty = \max_{x\in\mathcal{I}} |f(x)|$

Function Norms

- Functions can also have norms, for example, if $f(x)$ is defined in some interval $\mathcal{I}$

$$\|f\|_{L_2} = \sqrt{\int_{x \in \mathcal{I}} f^2(x)\,\mathrm{d}x}$$

- The L_2 vector space is the set of function where $\|f\|_{L_2} < \infty$
- The L_1-norm is given by $\|f\|_{L_1} = \int_{x \in \mathcal{I}} |f(x)|\,\mathrm{d}x$
- The infinite-norm is given by $\|f\|_\infty = \max_{x \in \mathcal{I}} |f(x)|$

Normed Vector Spaces

- Vector spaces with a distance (metric spaces) and vector spaces with a norm (normed vector spaces) are interesting objects
- They allow you to define a topology (open/closed sets, etc.)
- You can build up ideas about connectedness, continuity, contractive maps, fixed-point theorems, . . .
- For the most part we are going to consider an even more powerful vector space that has an inner-product defined

Normed Vector Spaces

- Vector spaces with a distance (metric spaces) and vector spaces with a norm (normed vector spaces) are interesting objects
- They allow you to define a topology (open/closed sets, etc.)
- You can build up ideas about connectedness, continuity, contractive maps, fixed-point theorems, . . .
- For the most part we are going to consider an even more powerful vector space that has an inner-product defined

Normed Vector Spaces

- Vector spaces with a distance (metric spaces) and vector spaces with a norm (normed vector spaces) are interesting objects
- They allow you to define a topology (open/closed sets, etc.)
- You can build up ideas about connectedness, continuity, contractive maps, fixed-point theorems, . . .
- For the most part we are going to consider an even more powerful vector space that has an inner-product defined

Normed Vector Spaces

- Vector spaces with a distance (metric spaces) and vector spaces with a norm (normed vector spaces) are interesting objects
- They allow you to define a topology (open/closed sets, etc.)
- You can build up ideas about connectedness, continuity, contractive maps, fixed-point theorems, . . .
- For the most part we are going to consider an even more powerful vector space that has an inner-product defined

Inner Products

- We will often consider objects with an *inner product*
- For vectors in $\mathbb{R}^n$

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{y} = \sum_{i=1}^{n} x_i \, y_i$$

- For functions

$$\langle f, g \rangle = \int_{x \in \mathcal{I}} f(x) \, g(x) \, \mathrm{d}x$$

- With matrices we don't usually think about inner products

Inner Products

- We will often consider objects with an *inner product*
- For vectors in $\mathbb{R}^n$

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\mathsf{T} \boldsymbol{y} = \sum_{i=1}^{n} x_i \, y_i$$

- For functions

$$\langle f, g \rangle = \int_{x \in \mathcal{I}} f(x) \, g(x) \, \mathrm{d}x$$

- With matrices we don't usually think about inner products

Inner Products

- We will often consider objects with an *inner product*
- For vectors in $\mathbb{R}^n$

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\mathsf{T} \boldsymbol{y} = \sum_{i=1}^{n} x_i \, y_i$$

- For functions

$$\langle f, g \rangle = \int_{x \in \mathcal{I}} f(x) \, g(x) \, \mathrm{d}x$$

- With matrices we don't usually think about inner products

Inner Products

- We will often consider objects with an *inner product*
- For vectors in $\mathbb{R}^n$

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{y} = \sum_{i=1}^{n} x_i \, y_i$$

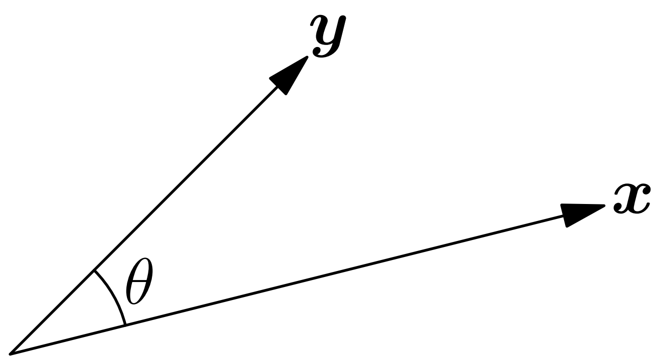
- For functions

$$\langle f, g \rangle = \int_{x \in \mathcal{I}} f(x) \, g(x) \, \mathrm{d}x$$

- With matrices we don't usually think about inner products

Angles Between Vectors

- A natural interpretation of the inner product is in providing a measure of the angle between vectors



$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{y} = \|\boldsymbol{x}\| \, \|\boldsymbol{y}\| \, \cos(\theta)$$

- Vectors are orthogonal if $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$

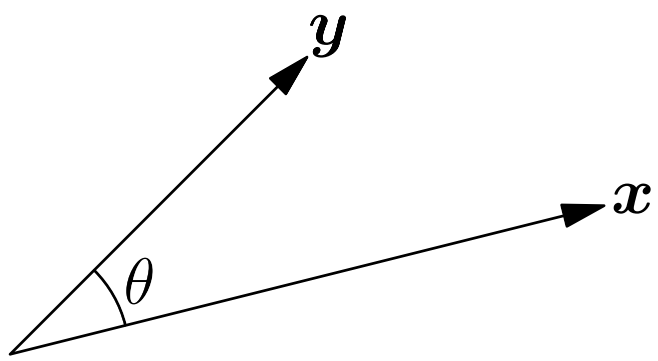
- We can extend this idea to functions

$$\langle f(x), g(x) \rangle = \int_{x \in \mathcal{I}} f(x) \, g(x) \mathrm{d}x = \|f(x)\| \, \|g(x)\| \, \cos(\theta)$$

- Note that $\sin(x)$ and $\cos(x)$ are orthogonal in the interval $[0, 2\,\pi]$

Angles Between Vectors

- A natural interpretation of the inner product is in providing a measure of the angle between vectors



$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{y} = \|\boldsymbol{x}\| \, \|\boldsymbol{y}\| \, \cos(\theta)$$

- Vectors are orthogonal if $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$

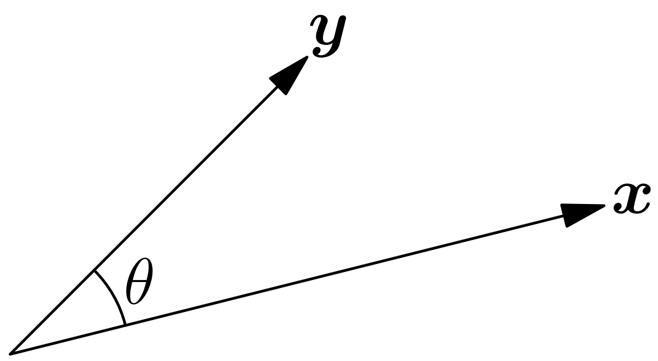
- We can extend this idea to functions

$$\langle f(x), g(x) \rangle = \int_{x \in \mathcal{I}} f(x)\, g(x) \mathrm{d}x = \|f(x)\| \, \|g(x)\| \, \cos(\theta)$$

- Note that $\sin(x)$ and $\cos(x)$ are orthogonal in the interval $[0, 2\,\pi]$

Angles Between Vectors

- A natural interpretation of the inner product is in providing a measure of the angle between vectors



$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{y} = \|\boldsymbol{x}\| \, \|\boldsymbol{y}\| \, \cos(\theta)$$

- Vectors are orthogonal if $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$

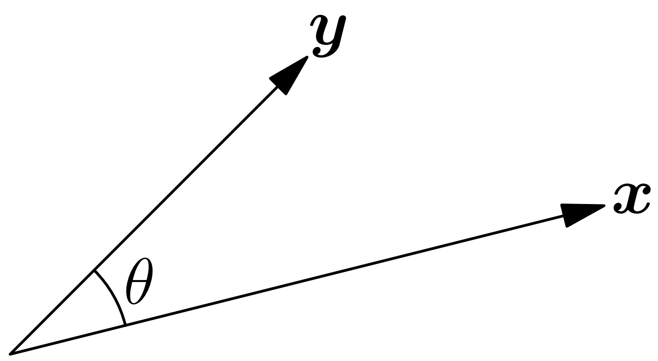
- We can extend this idea to functions

$$\langle f(x), g(x) \rangle = \int_{x \in \mathcal{I}} f(x) \, g(x) \mathrm{d}x = \|f(x)\| \, \|g(x)\| \, \cos(\theta)$$

- Note that $\sin(x)$ and $\cos(x)$ are orthogonal in the interval $[0, 2\,\pi]$

Angles Between Vectors

- A natural interpretation of the inner product is in providing a measure of the angle between vectors



$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{y} = \|\boldsymbol{x}\| \, \|\boldsymbol{y}\| \, \cos(\theta)$$

- Vectors are orthogonal if $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$

- We can extend this idea to functions

$$\langle f(x), g(x) \rangle = \int_{x \in \mathcal{I}} f(x)\, g(x) \mathrm{d}x = \|f(x)\| \, \|g(x)\| \, \cos(\theta)$$

- Note that $\sin(x)$ and $\cos(x)$ are orthogonal in the interval $[0, 2\,\pi]$

Basis Functions

- Any set of vectors $\{\boldsymbol{b}_i | i = 1, \ldots\}$ that span the space can be used as a basis or coordinate system

- The simplest and most useful case is when the vectors are orthogonal and normalised (i.e. $\|\boldsymbol{b}_i\| = 1$)

- In $\mathbb{R}^3$ we could use $\boldsymbol{b}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $\boldsymbol{b}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, $\boldsymbol{b}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

- This is not unique as we can rotate our basis vectors

- For an orthogonal basis we can write any vector as $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}^{\mathsf{T}}\boldsymbol{b}_1 \\ \boldsymbol{x}^{\mathsf{T}}\boldsymbol{b}_2 \\ \boldsymbol{x}^{\mathsf{T}}\boldsymbol{b}_3 \end{pmatrix}$

Basis Functions

- Any set of vectors $\{\boldsymbol{b}_i | i = 1, \ldots\}$ that span the space can be used as a basis or coordinate system

- The simplest and most useful case is when the vectors are orthogonal and normalised (i.e. $\|\boldsymbol{b}_i\| = 1$)

- In $\mathbb{R}^3$ we could use $\boldsymbol{b}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $\boldsymbol{b}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, $\boldsymbol{b}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

- This is not unique as we can rotate our basis vectors

- For an orthogonal basis we can write any vector as $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_1 \\ \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_2 \\ \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_3 \end{pmatrix}$

Basis Functions

- Any set of vectors $\{\boldsymbol{b}_i | i = 1, \ldots\}$ that span the space can be used as a basis or coordinate system

- The simplest and most useful case is when the vectors are orthogonal and normalised (i.e. $\|\boldsymbol{b}_i\| = 1$)

- In $\mathbb{R}^3$ we could use $\boldsymbol{b}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $\boldsymbol{b}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, $\boldsymbol{b}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

- This is not unique as we can rotate our basis vectors

- For an orthogonal basis we can write any vector as $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_1 \\ \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_2 \\ \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_3 \end{pmatrix}$

Basis Functions

- Any set of vectors $\{\boldsymbol{b}_i | i = 1, \ldots\}$ that span the space can be used as a basis or coordinate system

- The simplest and most useful case is when the vectors are orthogonal and normalised (i.e. $\|\boldsymbol{b}_i\| = 1$)

- In $\mathbb{R}^3$ we could use $\boldsymbol{b}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $\boldsymbol{b}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, $\boldsymbol{b}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

- This is not unique as we can rotate our basis vectors

- For an orthogonal basis we can write any vector as $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_1 \\ \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_2 \\ \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_3 \end{pmatrix}$

Basis Functions

- Any set of vectors $\{\boldsymbol{b}_i | i = 1, \ldots\}$ that span the space can be used as a basis or coordinate system

- The simplest and most useful case is when the vectors are orthogonal and normalised (i.e. $\|\boldsymbol{b}_i\| = 1$)

- In $\mathbb{R}^3$ we could use $\boldsymbol{b}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $\boldsymbol{b}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$, $\boldsymbol{b}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

- This is not unique as we can rotate our basis vectors

- For an orthogonal basis we can write any vector as $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_1 \\ \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_2 \\ \boldsymbol{x}^\mathsf{T}\boldsymbol{b}_3 \end{pmatrix}$

Orthogonal Functions

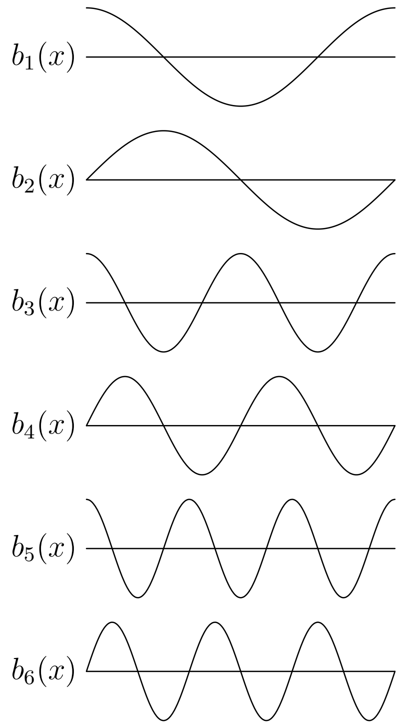
- For functions we can use any ortho-normal set of functions as a basis

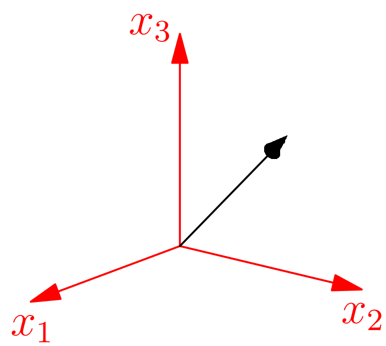
- The most familiar are the Fourier functions $\sin(n\,\theta)$ and $\cos(n\,\theta)$

- Any function in $C(0, 2\pi)$ can be represented by a point $\boldsymbol{f} = \begin{pmatrix} \langle f(x), b_1(x) \rangle \\ \langle f(x), b_2(x) \rangle \\ \vdots \end{pmatrix}$

- There might be an infinite number of components

- This is analogous to points in $\mathbb{R}^n$ (for large n)





Orthogonal Functions

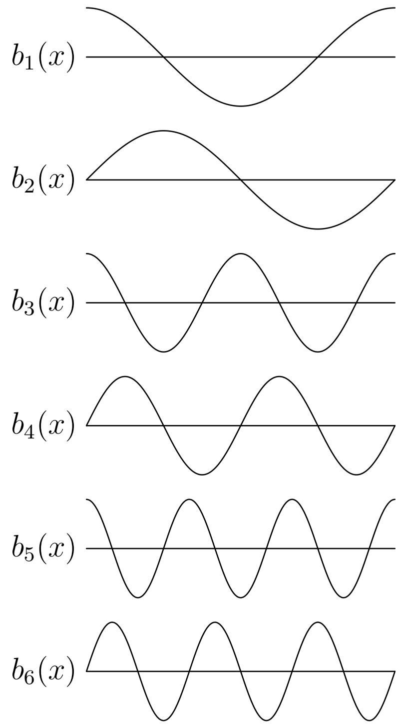
- For functions we can use any ortho-normal set of functions as a basis

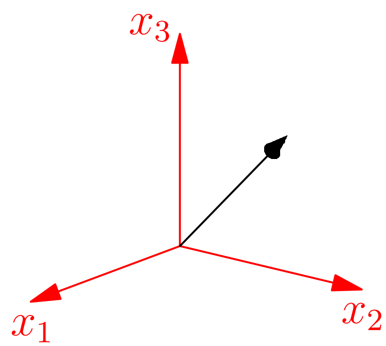
- The most familiar are the Fourier functions $\sin(n\,\theta)$ and $\cos(n\,\theta)$

- Any function in $C(0, 2\pi)$ can be represented by a point $\boldsymbol{f} = \begin{pmatrix} \langle f(x), b_1(x) \rangle \\ \langle f(x), b_2(x) \rangle \\ \vdots \end{pmatrix}$

- There might be an infinite number of components

- This is analogous to points in $\mathbb{R}^n$ (for large n)





Orthogonal Functions

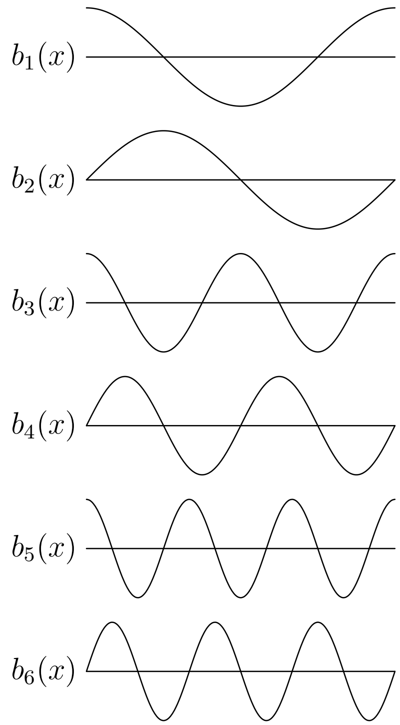
- For functions we can use any ortho-normal set of functions as a basis

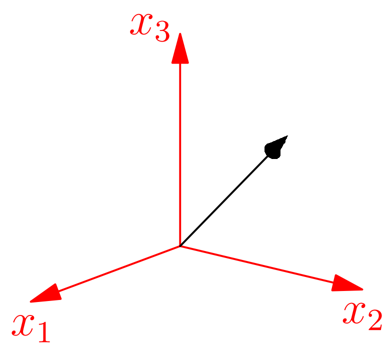
- The most familiar are the Fourier functions $\sin(n\,\theta)$ and $\cos(n\,\theta)$

- Any function in $C(0, 2\pi)$ can be represented by a point $\boldsymbol{f} = \begin{pmatrix} \langle f(x), b_1(x) \rangle \\ \langle f(x), b_2(x) \rangle \\ \vdots \end{pmatrix}$

- There might be an infinite number of components

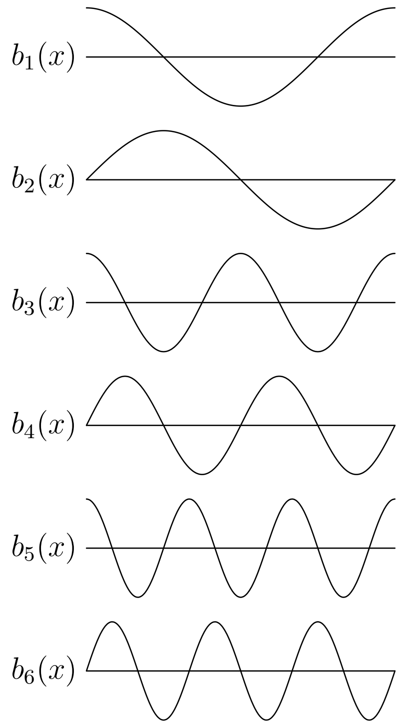
- This is analogous to points in $\mathbb{R}^n$ (for large n)

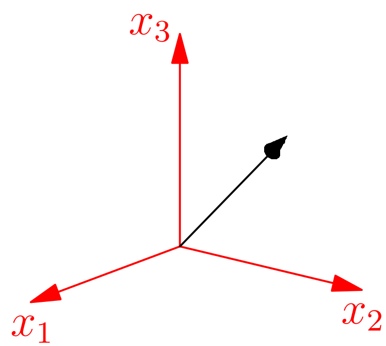




Orthogonal Functions

- For functions we can use any ortho-normal set of functions as a basis
- The most familiar are the Fourier functions $\sin(n\,\theta)$ and $\cos(n\,\theta)$
- Any function in $C(0, 2\pi)$ can be represented by a point $\boldsymbol{f} = \begin{pmatrix} \langle f(x), b_1(x) \rangle \\ \langle f(x), b_2(x) \rangle \\ \vdots \end{pmatrix}$
- There might be an infinite number of components
- This is analogous to points in $\mathbb{R}^n$ (for large n)





Orthogonal Functions

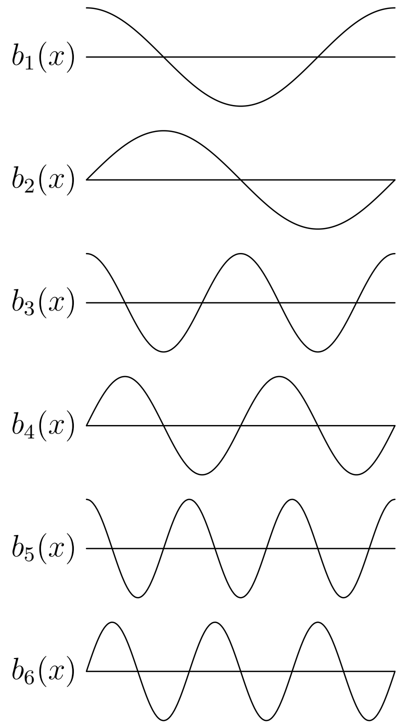
- For functions we can use any ortho-normal set of functions as a basis

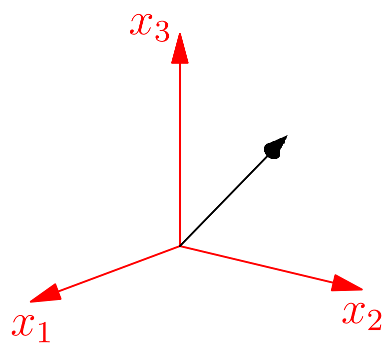
- The most familiar are the Fourier functions $\sin(n\,\theta)$ and $\cos(n\,\theta)$

- Any function in $C(0, 2\pi)$ can be represented by a point $\boldsymbol{f} = \begin{pmatrix} \langle f(x), b_1(x) \rangle \\ \langle f(x), b_2(x) \rangle \\ \vdots \end{pmatrix}$

- There might be an infinite number of components

- This is analogous to points in $\mathbb{R}^n$ (for large n)





Algebraic Structure

- We have gone to these lengths as we want to show that many properties of vectors are shared by other objects (matrices, functions, etc.)

- Mathematicians study *algebraic structures* such as vector spaces, metric spaces, Hilbert spaces (infinite dimensional spaces with a norm and an inner product)

- Many of the concepts (distance/metric, norms, inner products) will re-occur many times within the course

Algebraic Structure

- We have gone to these lengths as we want to show that many properties of vectors are shared by other objects (matrices, functions, etc.)
- Mathematicians study *algebraic structures* such as vector spaces, metric spaces, Hilbert spaces (infinite dimensional spaces with a norm and an inner product)
- Many of the concepts (distance/metric, norms, inner products) will re-occur many times within the course

Algebraic Structure

- We have gone to these lengths as we want to show that many properties of vectors are shared by other objects (matrices, functions, etc.)
- Mathematicians study *algebraic structures* such as vector spaces, metric spaces, Hilbert spaces (infinite dimensional spaces with a norm and an inner product)
- Many of the concepts (distance/metric, norms, inner products) will re-occur many times within the course

Outline

1. Course Details
2. Vector Spaces
3. **Operators**

$\mathbf{Mx=b}$

$\mathbf{Mv}_i = \lambda_i \mathbf{v}_i$

$\mathbf{b=M^{-1}x}$

Operators

- In machine learning we are interested in transforming our input vectors into some output predictions

- To accomplish this we will apply some mapping or operators on the vector $\mathcal{T} : \mathcal{V} \to \mathcal{V}'$

- This says that $\mathcal{T}$ maps some object $\boldsymbol{x} \in \mathcal{V}$ to an object $\boldsymbol{y} = \mathcal{T}[\boldsymbol{x}]$ in a new vector space $\mathcal{V}'$

- This new vector space may or may not be the same as the original vector space

- Our objects may be any object in a vector space such as a function

Operators

- In machine learning we are interested in transforming our input vectors into some output predictions
- To accomplish this we will apply some mapping or operators on the vector $\mathcal{T} : \mathcal{V} \to \mathcal{V}'$
- This says that $\mathcal{T}$ maps some object $\boldsymbol{x} \in \mathcal{V}$ to an object $\boldsymbol{y} = \mathcal{T}[\boldsymbol{x}]$ in a new vector space $\mathcal{V}'$
- This new vector space may or may not be the same as the original vector space
- Our objects may be any object in a vector space such as a function

Operators

- In machine learning we are interested in transforming our input vectors into some output predictions

- To accomplish this we will apply some mapping or operators on the vector $\mathcal{T} : \mathcal{V} \to \mathcal{V}'$

- This says that $\mathcal{T}$ maps some object $\boldsymbol{x} \in \mathcal{V}$ to an object $\boldsymbol{y} = \mathcal{T}[\boldsymbol{x}]$ in a new vector space $\mathcal{V}'$

- This new vector space may or may not be the same as the original vector space

- Our objects may be any object in a vector space such as a function

Operators

- In machine learning we are interested in transforming our input vectors into some output predictions
- To accomplish this we will apply some mapping or operators on the vector $\mathcal{T} : \mathcal{V} \to \mathcal{V}'$
- This says that $\mathcal{T}$ maps some object $\boldsymbol{x} \in \mathcal{V}$ to an object $\boldsymbol{y} = \mathcal{T}[\boldsymbol{x}]$ in a new vector space $\mathcal{V}'$
- This new vector space may or may not be the same as the original vector space
- Our objects may be any object in a vector space such as a function

Operators

- In machine learning we are interested in transforming our input vectors into some output predictions
- To accomplish this we will apply some mapping or operators on the vector $\mathcal{T} : \mathcal{V} \to \mathcal{V}'$
- This says that $\mathcal{T}$ maps some object $\boldsymbol{x} \in \mathcal{V}$ to an object $\boldsymbol{y} = \mathcal{T}[\boldsymbol{x}]$ in a new vector space $\mathcal{V}'$
- This new vector space may or may not be the same as the original vector space
- Our objects may be any object in a vector space such as a function

Linear Operators

- Operators are in general very complex, but a particular nice set of operators are linear operators

- $\mathcal{T}$ is a linear operator if

  1. $\mathcal{T}[a\,\boldsymbol{x}] = a\,\mathcal{T}[\boldsymbol{x}]$
  2. $\mathcal{T}[\boldsymbol{x}+\boldsymbol{y}] = \mathcal{T}[\boldsymbol{x}] + \mathcal{T}[\boldsymbol{y}]$

- For normal vectors the most general linear operation is

$$\mathcal{T}[\boldsymbol{x}] = \mathbf{M}\,\boldsymbol{x}$$

where $\mathbf{M}$ is a matrix

Linear Operators

- Operators are in general very complex, but a particular nice set of operators are linear operators

- $\mathcal{T}$ is a linear operator if

  1. $\mathcal{T}[a\,\boldsymbol{x}] = a\,\mathcal{T}[\boldsymbol{x}]$
  2. $\mathcal{T}[\boldsymbol{x}+\boldsymbol{y}] = \mathcal{T}[\boldsymbol{x}] + \mathcal{T}[\boldsymbol{y}]$

- For normal vectors the most general linear operation is

$$\mathcal{T}[\boldsymbol{x}] = \mathbf{M}\,\boldsymbol{x}$$

where $\mathbf{M}$ is a matrix

Linear Operators

- Operators are in general very complex, but a particular nice set of operators are linear operators

- $\mathcal{T}$ is a linear operator if

  1. $\mathcal{T}[a\,\boldsymbol{x}] = a\,\mathcal{T}[\boldsymbol{x}]$
  2. $\mathcal{T}[\boldsymbol{x}+\boldsymbol{y}] = \mathcal{T}[\boldsymbol{x}] + \mathcal{T}[\boldsymbol{y}]$

- For normal vectors the most general linear operation is

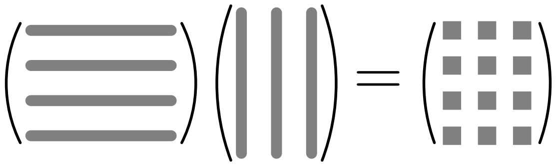
$$\mathcal{T}[\boldsymbol{x}] = \mathbf{M}\,\boldsymbol{x}$$

  where $\mathbf{M}$ is a matrix

Matrix multiplication

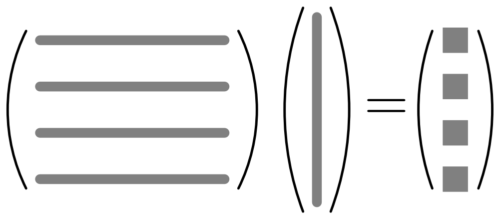
- For an $\ell \times m$ matrix $\mathbf{A}$ and an $m \times n$ matrix $\mathbf{B}$ we can compute the $(\ell \times n)$ product, $\mathbf{C} = \mathbf{A}\,\mathbf{B}$, such that

$$C_{ij} = \sum_{k=1}^{m} A_{ik}\, B_{kj}$$



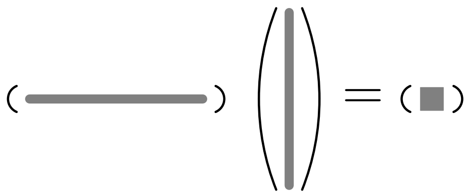
- Treating the vector $\boldsymbol{x}$ as a $n \times 1$ matrix then

$$\boldsymbol{y} = \mathbf{A}\,\boldsymbol{x} \quad \Rightarrow \quad y_i = \sum_j M_{ij} x_j$$



- Using the same matrix notation we can define the inner product as

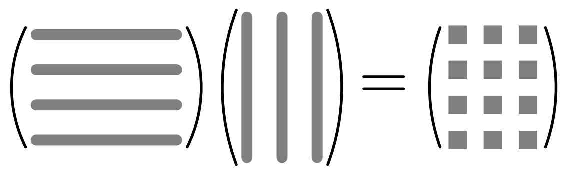
$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{y} = \sum_{i=1}^{n} x_i\, y_i$$



Matrix multiplication

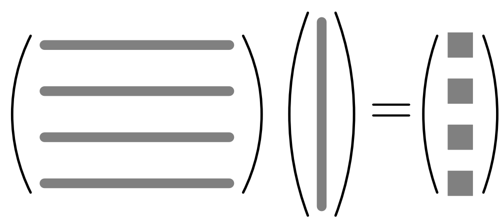
- For an $\ell \times m$ matrix $\mathbf{A}$ and an $m \times n$ matrix $\mathbf{B}$ we can compute the $(\ell \times n)$ product, $\mathbf{C} = \mathbf{A}\,\mathbf{B}$, such that

$$C_{ij} = \sum_{k=1}^{m} A_{ik}\, B_{kj}$$



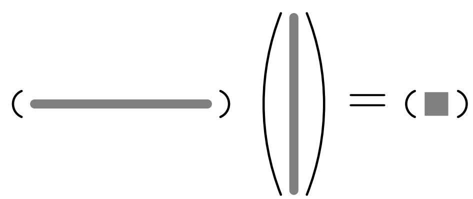
- Treating the vector $\boldsymbol{x}$ as a $n \times 1$ matrix then

$$\boldsymbol{y} = \mathbf{A}\,\boldsymbol{x} \quad \Rightarrow \quad y_i = \sum_j M_{ij} x_j$$



- Using the same matrix notation we can define the inner product as

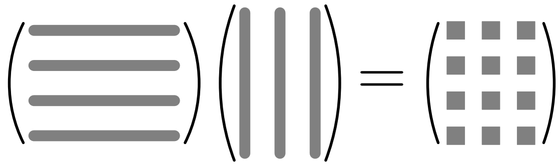
$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{y} = \sum_{i=1}^{n} x_i\, y_i$$



Matrix multiplication

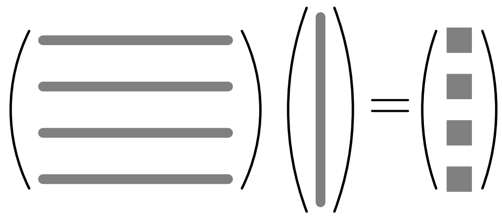
- For an $\ell \times m$ matrix $\mathbf{A}$ and an $m \times n$ matrix $\mathbf{B}$ we can compute the $(\ell \times n)$ product, $\mathbf{C} = \mathbf{A}\,\mathbf{B}$, such that

$$C_{ij} = \sum_{k=1}^{m} A_{ik}\, B_{kj}$$



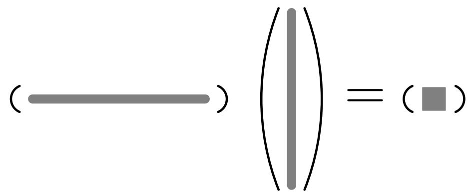
- Treating the vector $\boldsymbol{x}$ as a $n \times 1$ matrix then

$$\boldsymbol{y} = \mathbf{A}\,\boldsymbol{x} \quad \Rightarrow \quad y_i = \sum_{j} M_{ij} x_j$$



- Using the same matrix notation we can define the inner product as

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^{\mathsf{T}} \boldsymbol{y} = \sum_{i=1}^{n} x_i\, y_i$$



Kernels

- The equivalent of a matrix for functions (i.e. a linear operator) is known as a kernel $K(x, y)$

$$g(x) = \mathcal{T}[f(x)] = \int_{y \in \mathcal{I}} K(x, y)\, f(y)\, \mathrm{d}y$$

- Our domain does not need to be one dimensional, e.g.

$$g(\boldsymbol{x}) = \mathcal{T}[f(\boldsymbol{x})] = \int_{\boldsymbol{y} \in \mathcal{I}} K(\boldsymbol{x}, \boldsymbol{y})\, f(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}$$

- We shall soon see examples of high-dimensional kernels

Kernels

- The equivalent of a matrix for functions (i.e. a linear operator) is known as a kernel $K(x, y)$

$$g(x) = \mathcal{T}[f(x)] = \int_{y \in \mathcal{I}} K(x, y)\, f(y)\, \mathrm{d}y$$

- Our domain does not need to be one dimensional, e.g.

$$g(\boldsymbol{x}) = \mathcal{T}[f(\boldsymbol{x})] = \int_{\boldsymbol{y} \in \mathcal{I}} K(\boldsymbol{x}, \boldsymbol{y})\, f(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}$$

- We shall soon see examples of high-dimensional kernels

Kernels

- The equivalent of a matrix for functions (i.e. a linear operator) is known as a kernel $K(x, y)$

$$g(x) = \mathcal{T}[f(x)] = \int_{y \in \mathcal{I}} K(x, y)\, f(y)\, \mathrm{d}y$$

- Our domain does not need to be one dimensional, e.g.

$$g(\boldsymbol{x}) = \mathcal{T}[f(\boldsymbol{x})] = \int_{\boldsymbol{y} \in \mathcal{I}} K(\boldsymbol{x}, \boldsymbol{y})\, f(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}$$

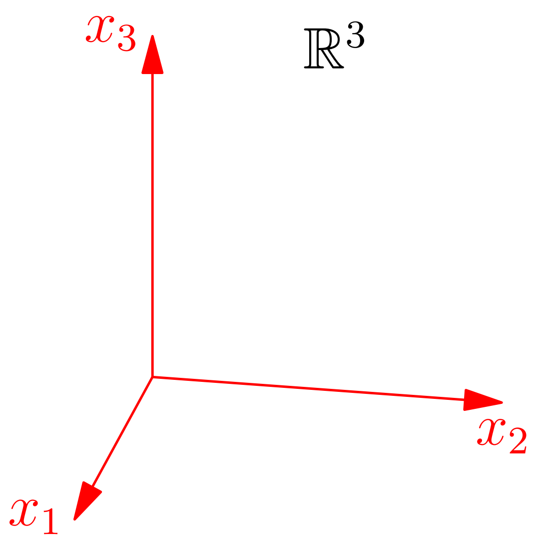
- We shall soon see examples of high-dimensional kernels

General Linear Mappings

- In general a linear operator will map vectors between different vector spaces
- E.g. $\mathbb{R}^3 \rightarrow \mathbb{R}^2$

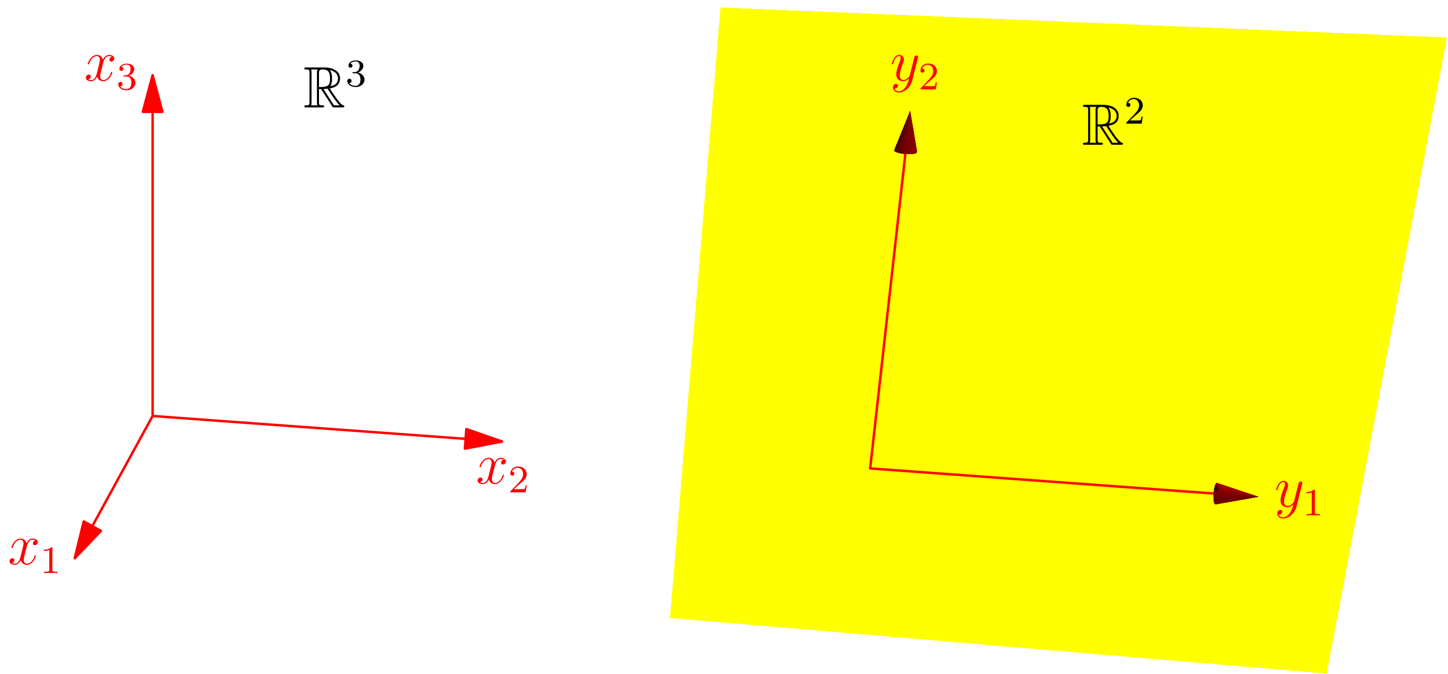
General Linear Mappings

- In general a linear operator will map vectors between different vector spaces
- E.g. $\mathbb{R}^3 \to \mathbb{R}^2$



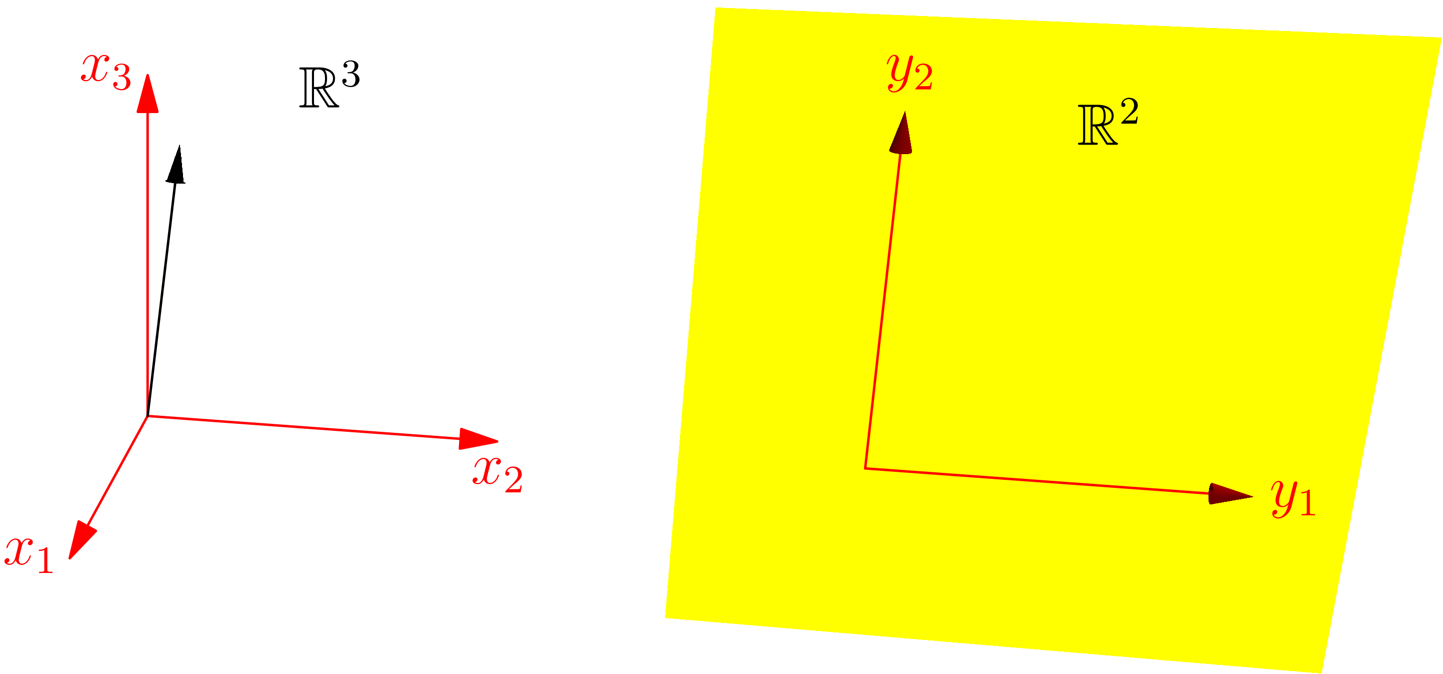
General Linear Mappings

- In general a linear operator will map vectors between different vector spaces
- E.g. $\mathbb{R}^3 \to \mathbb{R}^2$



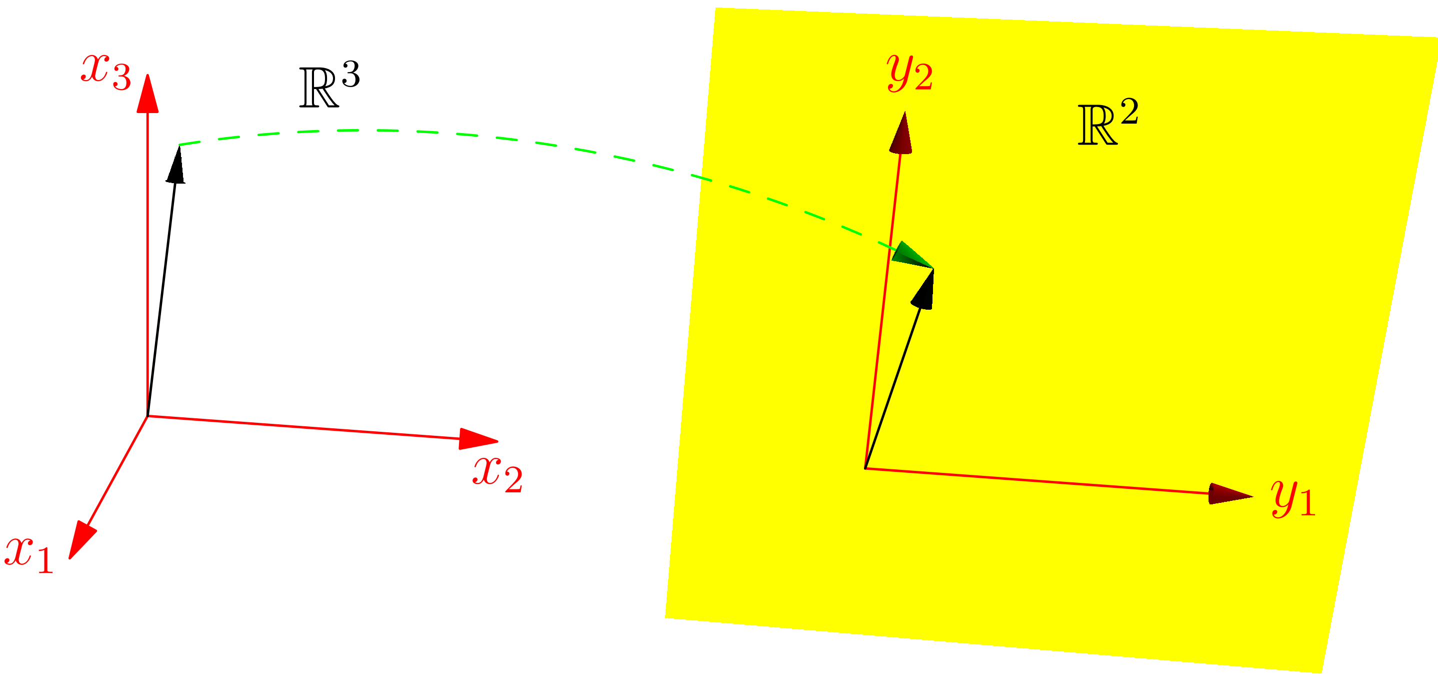
General Linear Mappings

- In general a linear operator will map vectors between different vector spaces
- E.g. $\mathbb{R}^3 \to \mathbb{R}^2$



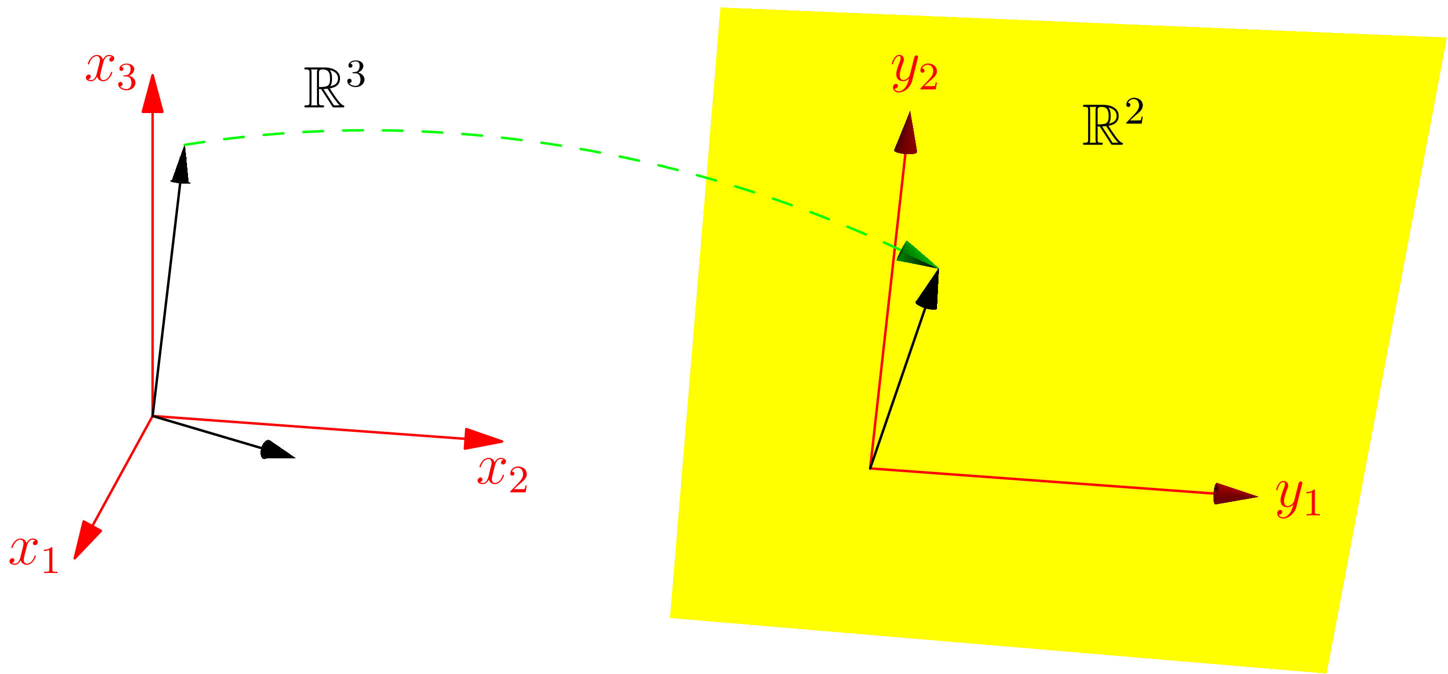
General Linear Mappings

- In general a linear operator will map vectors between different vector spaces
- E.g. $\mathbb{R}^3 \to \mathbb{R}^2$



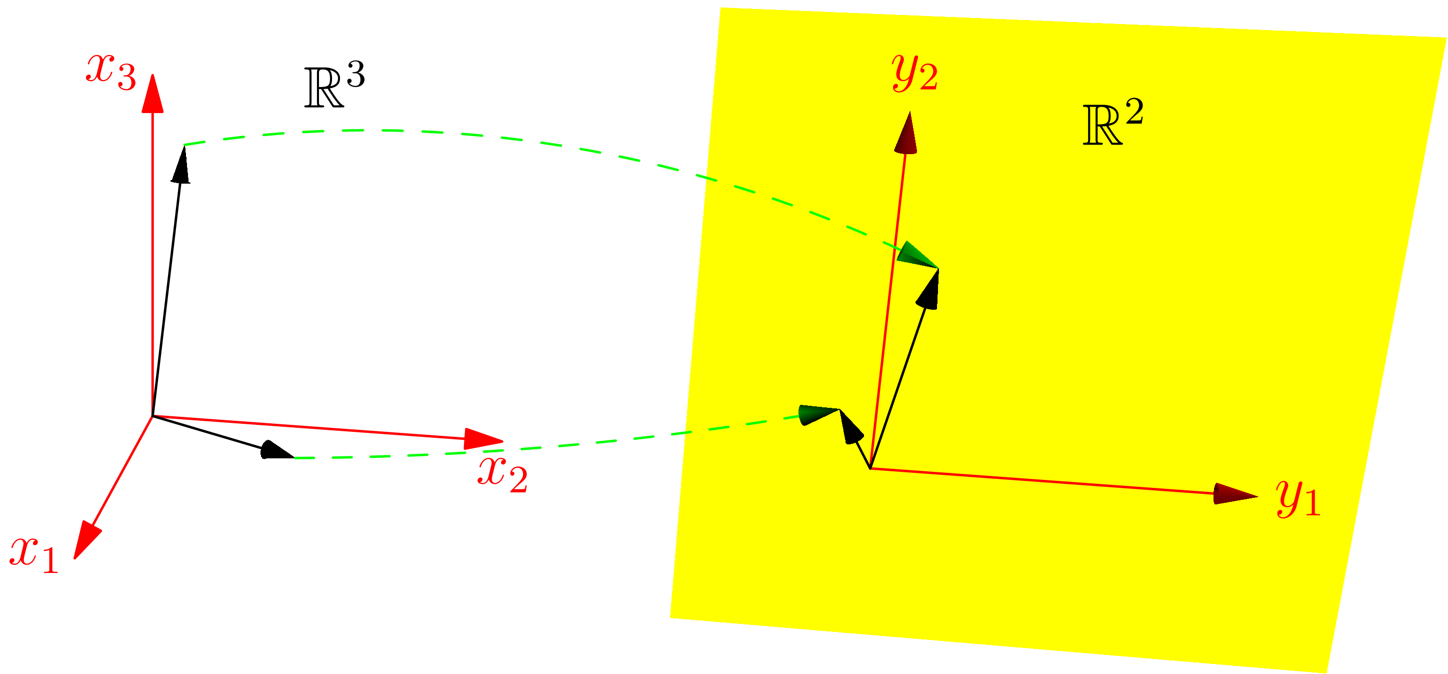
General Linear Mappings

- In general a linear operator will map vectors between different vector spaces
- E.g. $\mathbb{R}^3 \to \mathbb{R}^2$



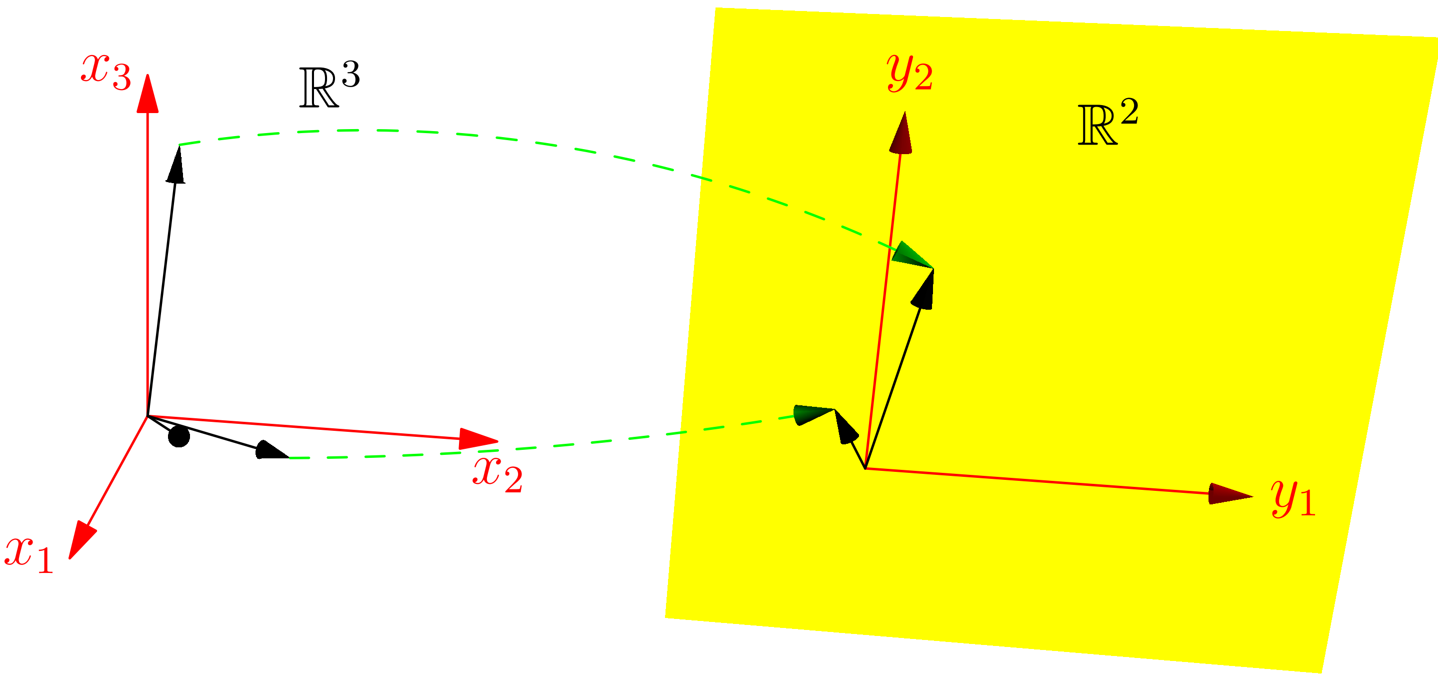
General Linear Mappings

- In general a linear operator will map vectors between different vector spaces
- E.g. $\mathbb{R}^3 \to \mathbb{R}^2$



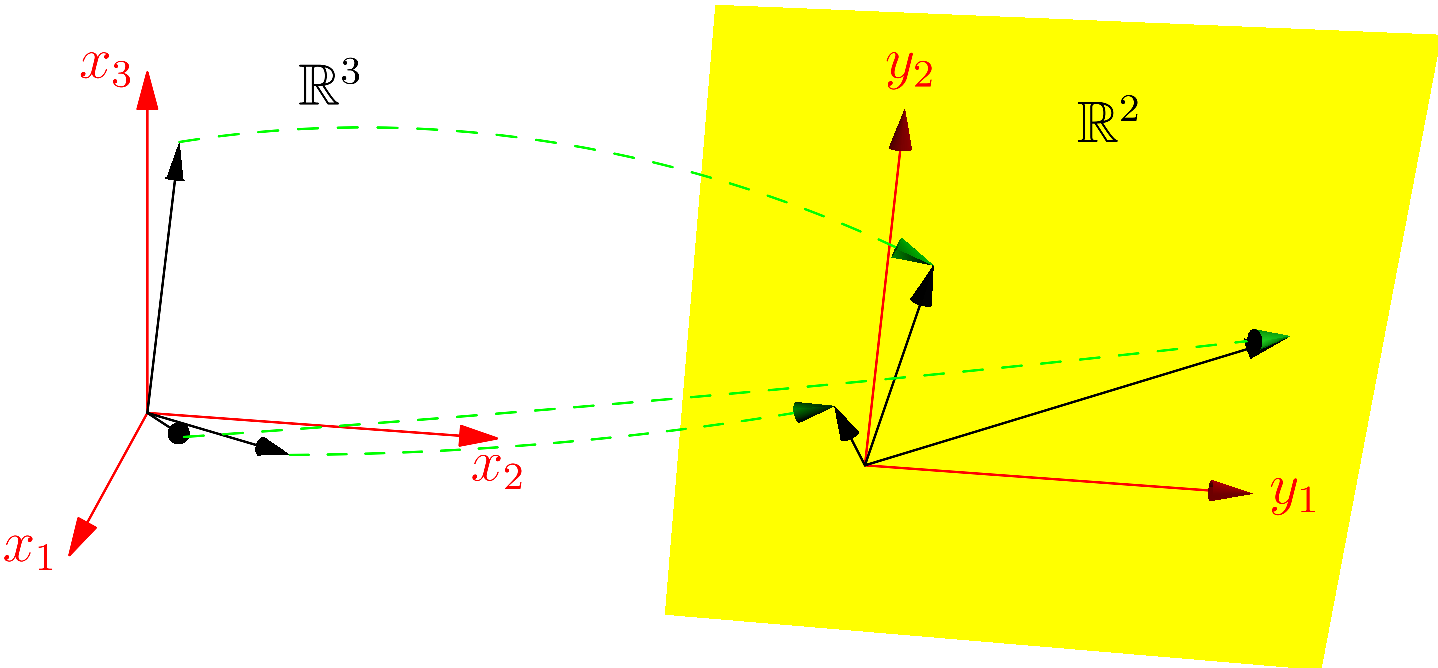
General Linear Mappings

- In general a linear operator will map vectors between different vector spaces
- E.g. $\mathbb{R}^3 \to \mathbb{R}^2$



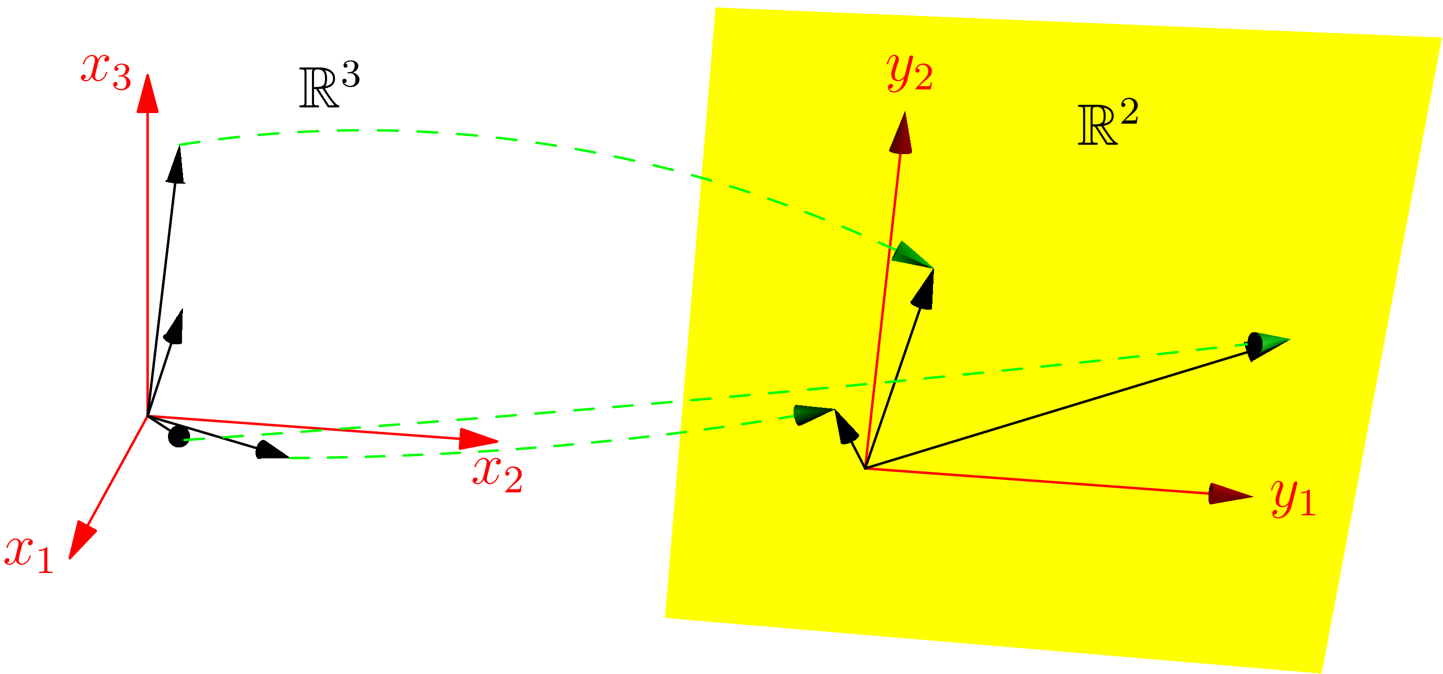
General Linear Mappings

- In general a linear operator will map vectors between different vector spaces
- E.g. $\mathbb{R}^3 \rightarrow \mathbb{R}^2$



General Linear Mappings

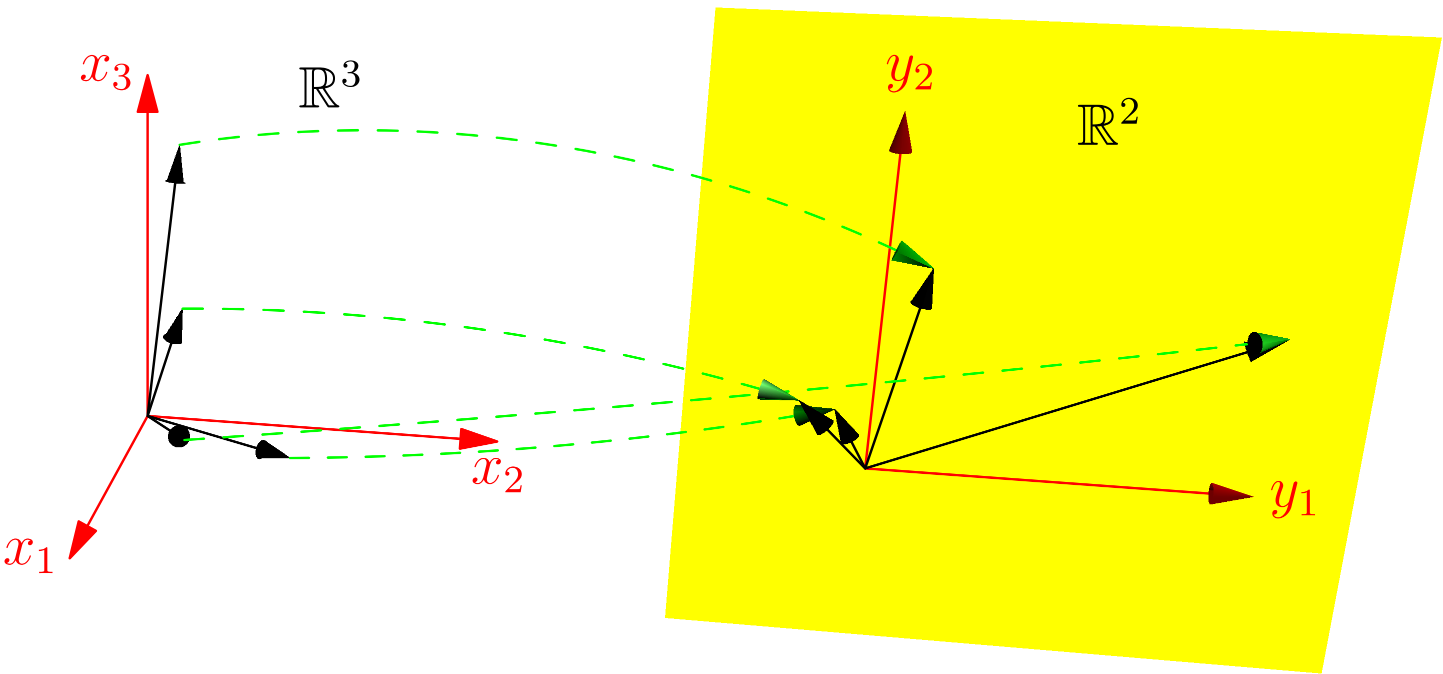
- In general a linear operator will map vectors between different vector spaces
- E.g. $\mathbb{R}^3 \to \mathbb{R}^2$



General Linear Mappings

- In general a linear operator will map vectors between different vector spaces

- E.g. $\mathbb{R}^3 \to \mathbb{R}^2$



Square Matrices

- We will spend a lot of time on operators that map from a vector space onto itself $\mathcal{T} : \mathcal{V} \to \mathcal{V}$
- For vectors in $\mathbb{R}^n$ such linear operators are represented by square matrices
- When there is a one-to-one mapping then we have a unique inverse
- We will study such mappings in detail in the next lecture

Square Matrices

- We will spend a lot of time on operators that map from a vector space onto itself $\mathcal{T} : \mathcal{V} \to \mathcal{V}$
- For vectors in $\mathbb{R}^n$ such linear operators are represented by square matrices
- When there is a one-to-one mapping then we have a unique inverse
- We will study such mappings in detail in the next lecture

Square Matrices

- We will spend a lot of time on operators that map from a vector space onto itself $\mathcal{T} : \mathcal{V} \to \mathcal{V}$
- For vectors in $\mathbb{R}^n$ such linear operators are represented by square matrices
- When there is a one-to-one mapping then we have a unique inverse
- We will study such mappings in detail in the next lecture

Square Matrices

- We will spend a lot of time on operators that map from a vector space onto itself $\mathcal{T} : \mathcal{V} \to \mathcal{V}$
- For vectors in $\mathbb{R}^n$ such linear operators are represented by square matrices
- When there is a one-to-one mapping then we have a unique inverse
- We will study such mappings in detail in the next lecture

Summary

- We haven't covered much machine learning as such
- But mathematics is the language of machine learning and you have to get used to it
- Mathematics is like programming, if you don't understand the syntax and you can't write it down then its meaningless
- We've taken a high level view of vector spaces, this will pay us back later as we look at kernel methods

Summary

- We haven't covered much machine learning as such—sorry
- But mathematics is the language of machine learning and you have to get used to it
- Mathematics is like programming, if you don't understand the syntax and you can't write it down then its meaningless
- We've taken a high level view of vector spaces, this will pay us back later as we look at kernel methods

Summary

- We haven't covered much machine learning as such—sorry
- But mathematics is the language of machine learning and you have to get used to it
- Mathematics is like programming, if you don't understand the syntax and you can't write it down then its meaningless
- We've taken a high level view of vector spaces, this will pay us back later as we look at kernel methods

Summary

- We haven't covered much machine learning as such—sorry
- But mathematics is the language of machine learning and you have to get used to it
- Mathematics is like programming, if you don't understand the syntax and you can't write it down then its meaningless
- We've taken a high level view of vector spaces, this will pay us back later as we look at kernel methods

Summary

- We haven't covered much machine learning as such—sorry
- But mathematics is the language of machine learning and you have to get used to it
- Mathematics is like programming, if you don't understand the syntax and you can't write it down then its meaningless
- We've taken a high level view of vector spaces, this will pay us back later as we look at kernel methods