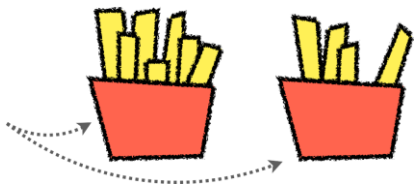
Data records: Similarity and distance

Lecture 17

Recap: tuples, attributes, datasets

- We define a set of attributes, and, for each entity, we record the values for each such attribute.



Dataset

Monday	Location A	21
Tuesday	Location B	24
Wednesday	Location A	19
Thursday	L:ocation C	25

Each observation is a tuple with three dimensions.

(“Monday”, “A”, 21)
(“Tuesday”, “B”, 24)
(“Wednesday”, “A”, 19)
(“Thursday”, “C”, 25)

Note that in this case not all tuple values are numeric.

Working with numeric attributes

- Now we are going to concentrate on data tuples that contain numbers only.
  - Each attribute represents a numeric dimension
  - Each record can be thought of as a point or a vector in a multi-dimensional space

Example of numeric tuples

- Imagine that we record a pair of coordinates for every coffee shop in town.
- The numeric space we are operating in is a plane: it has two dimensions.
- Numeric spaces are described using letter R. The 2D space is represented as R^2.

Store name	x_1	x_2
A	6	6
B	9	1
C	8	6
D	5	8
E	4	9

Numeric tuples

(6, 6)
(9, 1)
(8, 6)
(5, 8)
(4, 9)

Points in 2D

Vectors in 2D

$$\mathbf{v}=\begin{bmatrix}x_1\\x_2\end{bmatrix} \quad \mathbf{a}=\begin{bmatrix}6\\6\end{bmatrix} \quad \mathbf{b}=\begin{bmatrix}9\\1\end{bmatrix} \quad \mathbf{c}=\begin{bmatrix}8\\6\end{bmatrix}$$

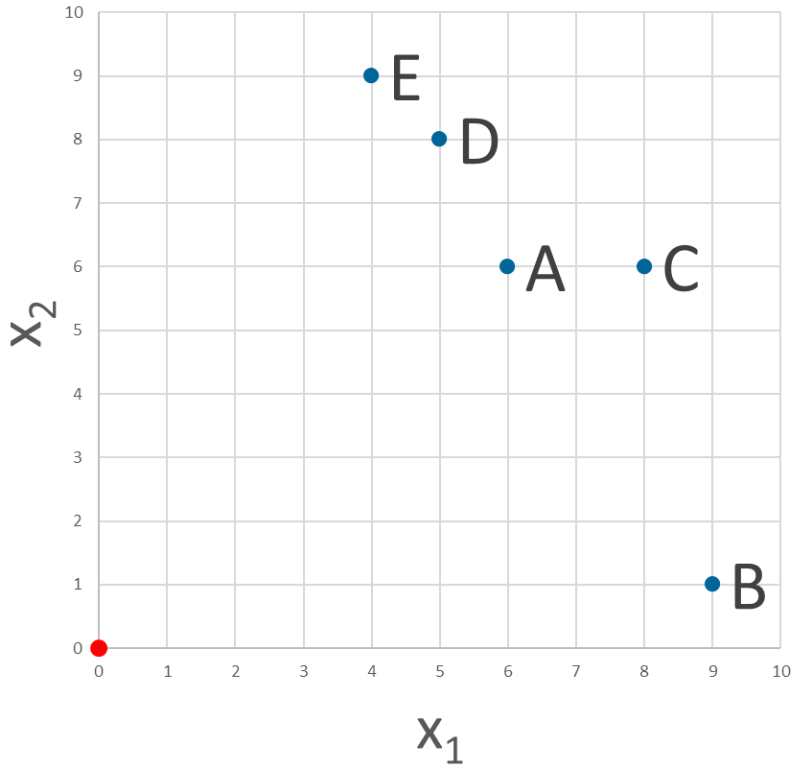
Distance between data records

Geometric interpretation

Plotting numeric 2D data on a plane

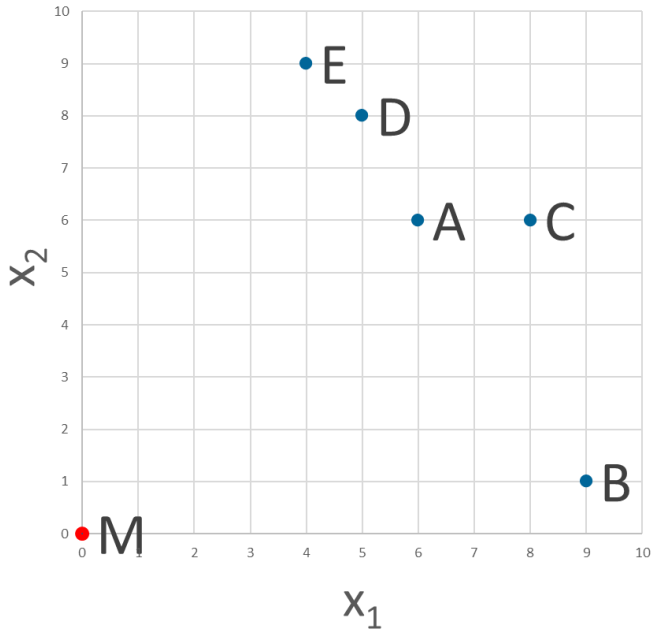
Store name	x_1	x_2
A	6	6
B	9	1
C	8	6
D	5	8
E	4	9

- We can plot our coffee dataset on a 2D plane.



The closest coffee shop

Store name	x_1	x_2
A	6	6
B	9	1
C	8	6
D	5	8
E	4	9

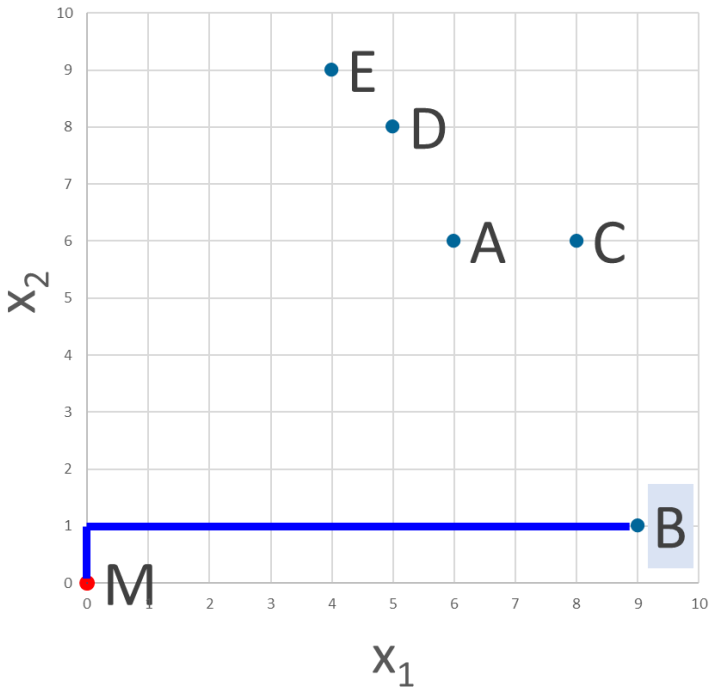


- If you are standing at point M(0, 0), which of the stores are closest to you?

- This depends on how we define *distance*

Manhattan distance

Store name	x_1	x_2	*manh (M,T)*
A	6	6	12
B	9	1	10
C	8	6	14
D	5	8	13
E	4	9	13



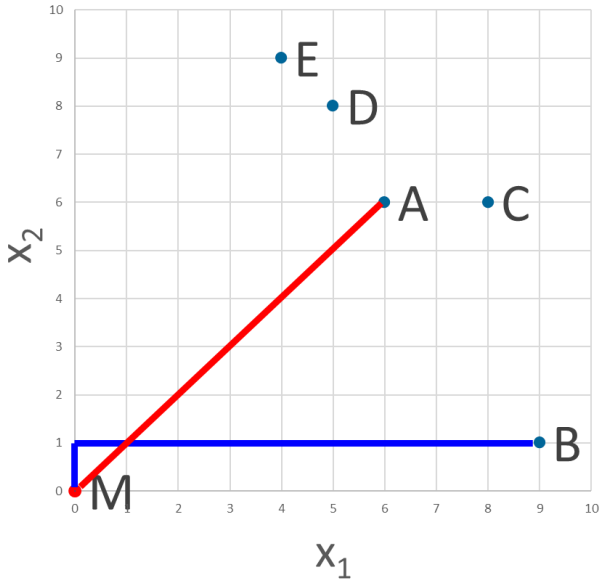
- If we can only move along the grid lines then to find the distance we add the number of blocks across each dimension.
- This is called a *Manhattan distance*.

$$manh(m,t) = |t_1 - m_1| + |t_2 - m_2|$$

- By Manhattan distance, the closest point to M is B.

Euclidean distance

Store name	x_1	x_2	*manh (M,T)*	*eucl(M,T)*
A	6	6	12	8.5
B	9	1	10	9.1
C	8	6	14	10.0
D	5	8	13	9.4
E	4	9	13	9.8

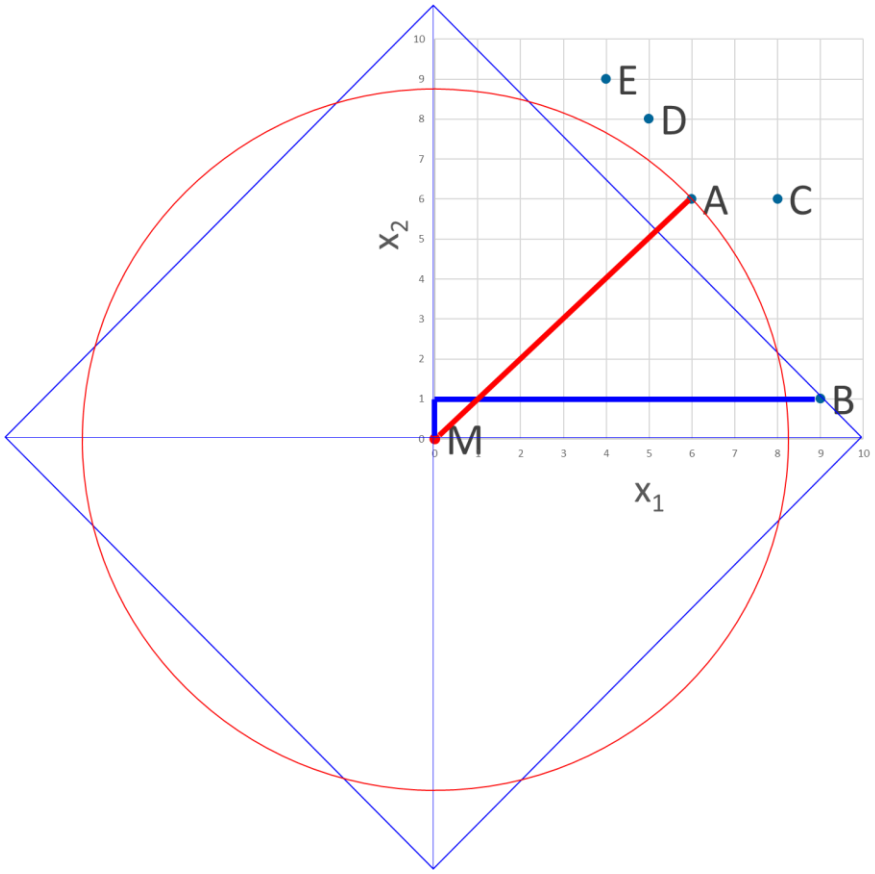


- If we are allowed to move directly from M to A ignoring any obstacles, then we can compute this distance using a Pythagorean theorem.
- This is called *Euclidean distance*.

$$eucl(m,t) = \sqrt{(t_1 - m_1)^2 + (t_2 - m_2)^2}$$

- According to Euclidean distance the closest point to M is A

Distance metrics inspired by maps

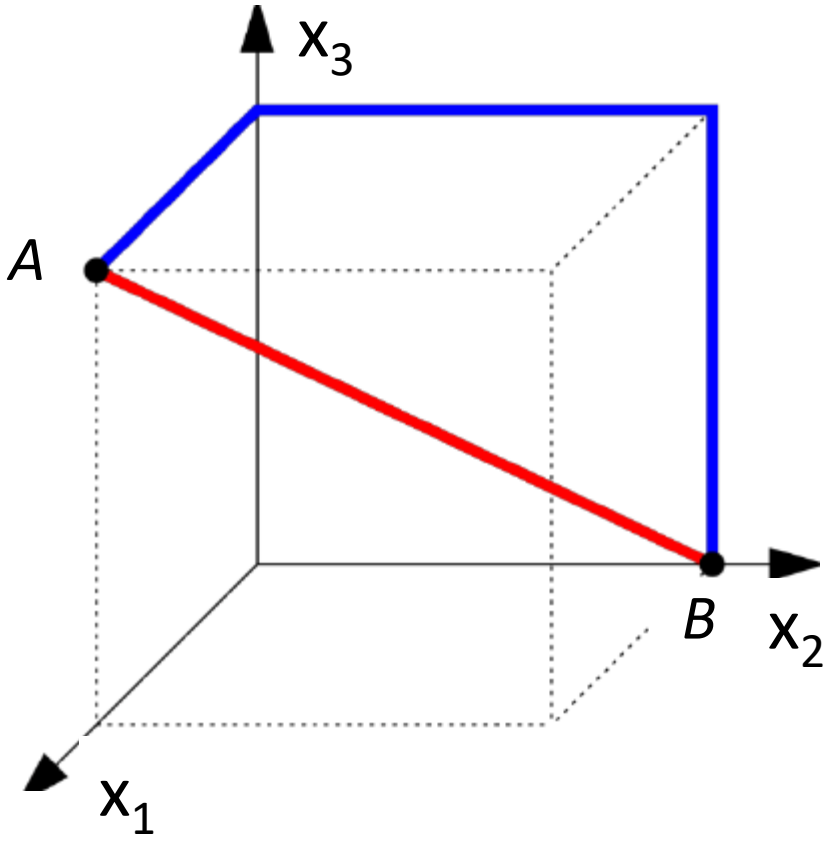


- *Euclidean* (circle radius) – best for natural, continuous spaces like geographic or physical data.
- *Manhattan* (diamonds) – works well for grid-based movement (e.g., city streets) and is more robust to outliers.

Works also for 3D spaces

- What if our numeric tuples have three dimensions: say, we added an elevation about the ground to our dataset?
- Same distance formulas

Store name	x1	x2	x3
A	6	6	4
B	9	1	1
C	8	6	3
D	5	8	1
E	4	9	2



$$euclid(A,B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2}$$

Works for any number of dimensions

$$eucl(A,B) = \sqrt{\sum_{i=1}^{N}(a_i - b_i)^2}$$

$$manh(A,B) = \sqrt{\sum_{i=1}^{N}|a_i - b_i|}$$

Distance between **records** in a *k*-dimensional space of *k* attributes

- Every data tuple with k numeric attributes can be thought of as a point in R^k.

- This is a useful idea, because we can now compute distance between a pair of tuples.

Example: movie rating dataset

- Consider a dataset of movie ratings, where we record the movie ratings of 5 movies by 6 of our friends.

	Spiderman	Beethoven	Star Wars	Shrek	Wish Dragon	The Batman
Friend A	4	3	5	2	3	3
Friend B	5	5	5	5	4	5
Friend C	3	4	2	4	3	2
Friend D	4	3	4	4	4	4
Friend E	2	3	2	3	2	2

Distance to my friends

	Spiderman	Beethoven	Star Wars	Shrek	Wish Dragon
Me	5	4	5	5	5

- *Me* also rated the first five movies from this list.

- Using geometric interpretation of data, we can think of each friend as a point in a 5-dimensional space
- We can compute the distance between *Me* and every one of my friends.

Most similar to *Me*

We can order the friends by their similarity to *Me*: the smaller the distance the greater the similarity in our movie preferences.

	Manh (Me, Friend i)	*Eucl* (Me, Friend i)
Friend A	7	3.9
Friend B	2	1.4
Friend C	8	4.2
Friend D	5	2.2
Friend E	12	5.7

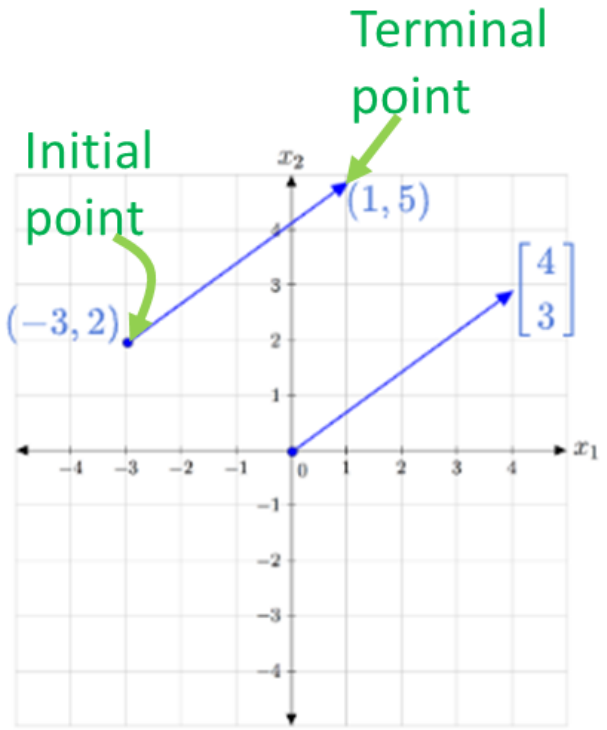
How can that be useful?

Distance between data records

Vector interpretation

Recap: Vectors

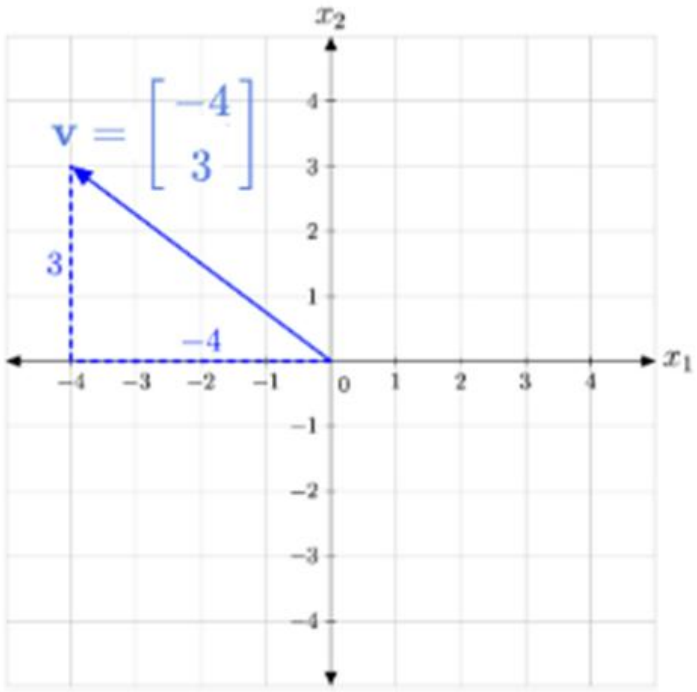
- A vector is a mathematical object that has both value (called magnitude of a vector), and the direction.
- In space, each vector can be described by the coordinates of its initial point (tail) and terminal point (head). (Vector is heading into the direction of the head, leaving behind the tail).



Because each vector is uniquely defined by direction and by how many steps to move, both vectors in the picture are identical: they both have the same magnitude and point in the same direction

Position vectors

- We can transpose any vector such that its initial point is aligned with the origin of the coordinate system.
- Such a vector is said to be in standard position and is called a *position vector*.



Vector components

- Each position vector in R^2 can be defined by the ordered pair of coordinates.

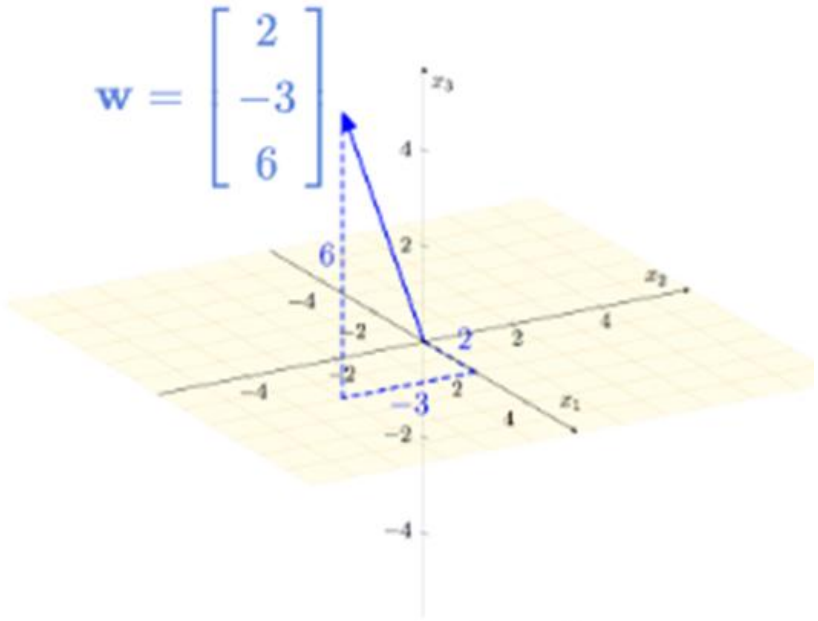
- In general, we can think of a vector in any space as an ordered list of scalar values, or as a *numerical tuple*.

- Each scalar value x_i is called a *component* of the vector: it specifies the coordinate for a specific dimension *i*.

$$v = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

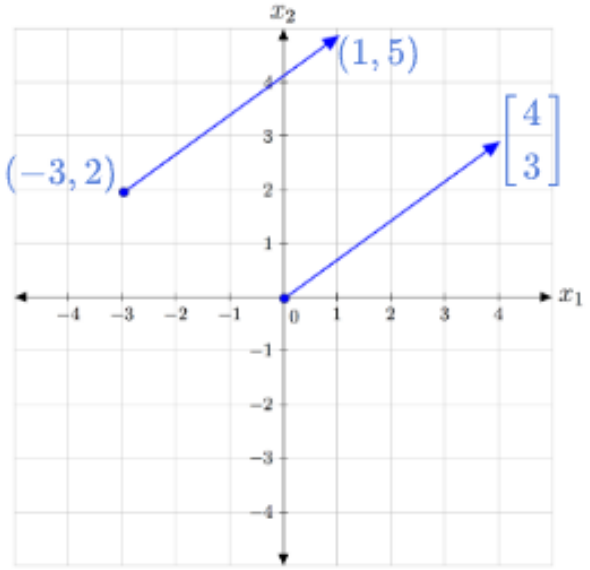
$$(x_1, x_2, \ldots, x_n)$$

Example of 3 components in R^3



Vector: $\mathbf{w} = \begin{bmatrix} 2 \\ -3 \\ 6 \end{bmatrix}$

Computing a position vector: example in 2D

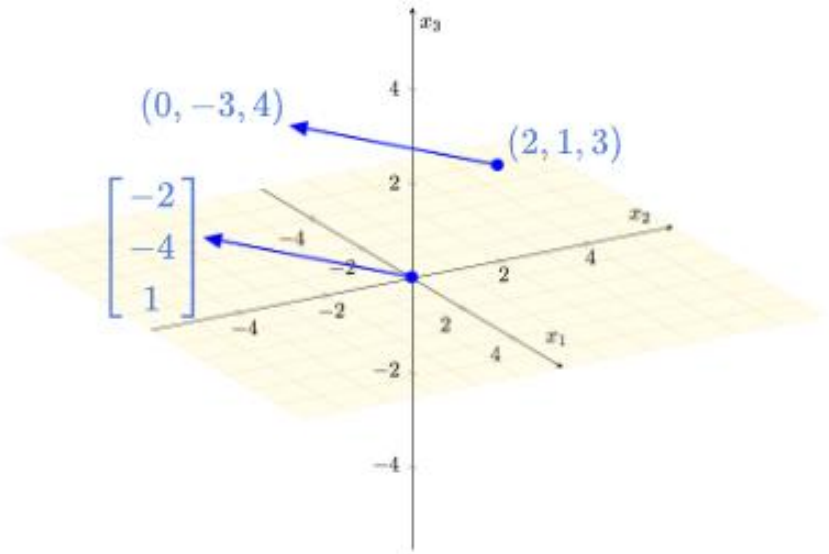


Position vector:

$$\begin{bmatrix} b_1 - a_1 \\ b_2 - a_2 \end{bmatrix} = \begin{bmatrix} 1-(-3) \\ 5-2 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

- Given the initial point A and the terminal point B of a vector, the end of the position vector can be found by subtracting corresponding components: i.e. subtract coordinates of beginning from the coordinates of the end across each dimension *i*: $b_i - a_i$

Computing a position vector: example in 3D



Position vector:

$$\begin{bmatrix} b_1 - a_1 \\ b_2 - a_2 \\ b_3 - a_3 \end{bmatrix} = \begin{bmatrix} 0 - 2 \\ -3 - 1 \\ 4 - 3 \end{bmatrix} = \begin{bmatrix} -2 \\ -4 \\ 1 \end{bmatrix}$$

- A position vector in R^3 can be computed using similar process.

Position vector: exercise

- A vector in R^2 is described by an initial and a terminal point:

Initial point: (-1,-2)

Terminal point: (-5,-3)

- What is the corresponding position vector?

Position vector: exercise

- A vector in R^2 is described by an initial and a terminal point:

Initial point: (-1,-2)

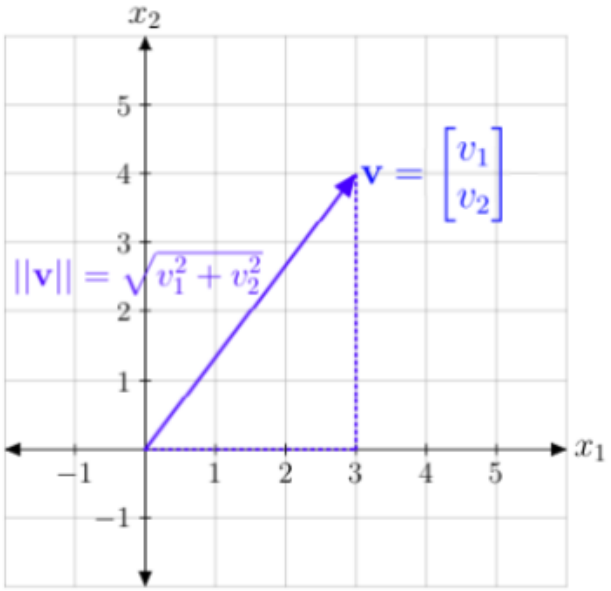
Terminal point: (-5,-3)

- What is the corresponding position vector?

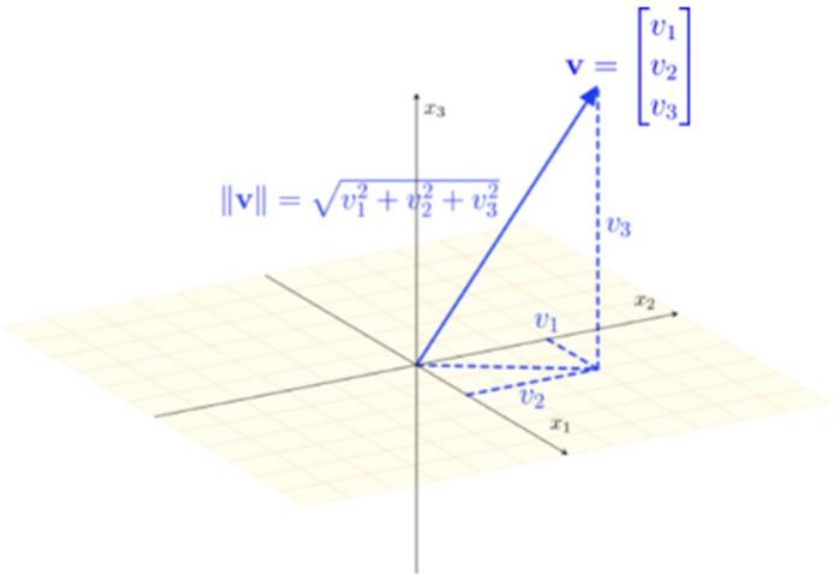
(-4,-1)

Magnitude

- The *magnitude* of vector **v**, denoted by |**v**| (or v), is the length of vector **v**
- Magnitude can be found using Pythagorean theorem.



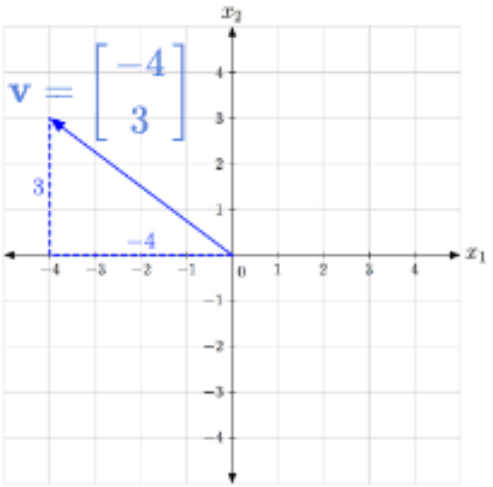
Magnitude of a vector in R^2.



Magnitude of a vector in R^3.

Magnitude: example in R^2

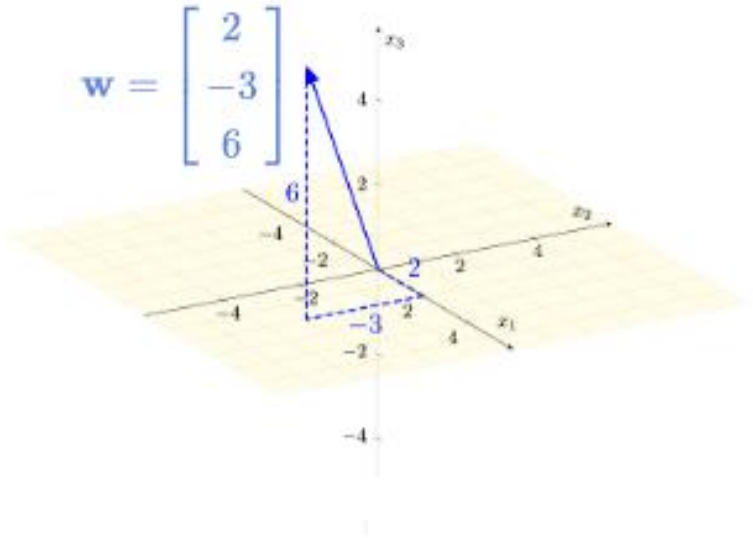
$$\mathbf{v} = \begin{bmatrix} -4 \\ 3 \end{bmatrix}$$



$$\begin{aligned} \|\mathbf{v}\| &= \sqrt{(-4)^2 + 3^2} \\ &= \sqrt{25} \\ &= 5 \end{aligned}$$

Magnitude: example in R^3

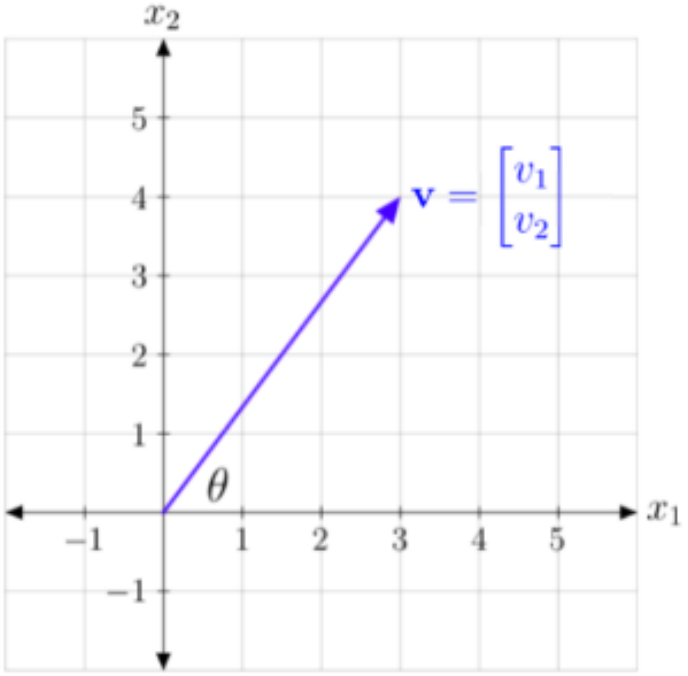
$$\mathbf{w} = \begin{bmatrix} 2 \\ -3 \\ 6 \end{bmatrix}$$



$$\begin{aligned} \|\mathbf{w}\| &= \sqrt{(2)^2 + (-3)^2 + 6^2} \\ &= \sqrt{49} \\ &= 7 \end{aligned}$$

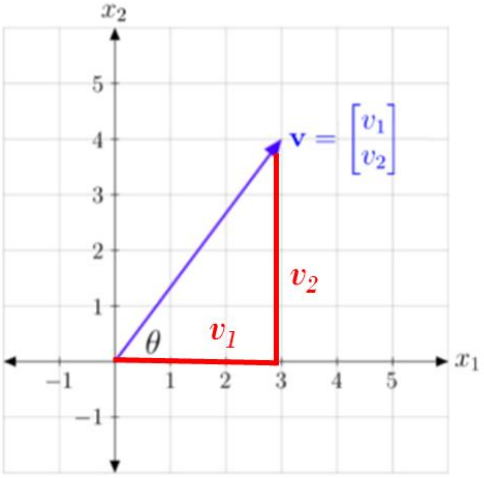
Direction

- The direction of a vector in R^2 is given by the angle θ the position vector makes with respect to the horizontal axis, or **positive x_1-axis**.



Computing an angle from coordinates

- We are computing an angle between two vectors: vector (1,0) and the given vector ***v***.
- If we know two sides of the right triangle (coordinates v_1 and v_2), we can compute tangent of θ, and from *tan* the angle θ itself:



$$tan\,\theta = \frac{v_2}{v_1}$$

$$\theta = tan^{-1}\left(\frac{v_2}{v_1}\right)$$

Finding direction: example

- Find the direction of the vector with initial point P(-8, 1) and terminal point Q(-2,-5)

Finding direction: example

- Solution

1. First find the position vector

$$\mathbf{u} = \begin{bmatrix} -2-(-8) \\ -5-1 \end{bmatrix} = \begin{bmatrix} 6 \\ -6 \end{bmatrix}$$

Finding direction: example

- Solution

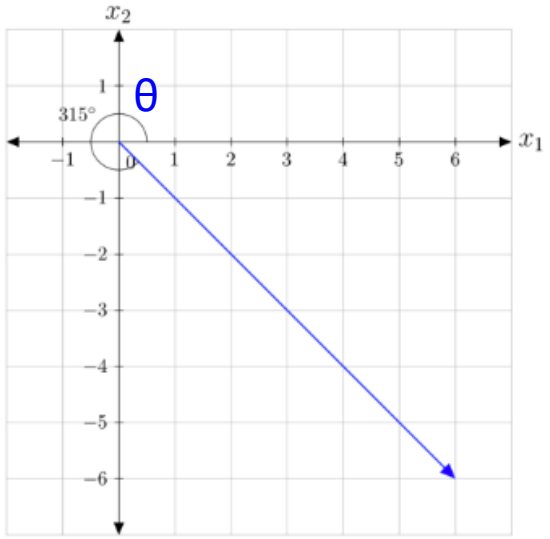
1. First find the position vector $\mathbf{u} = \begin{bmatrix} 6 \\ -6 \end{bmatrix}$
2. The direction is given by the angle θ.

$$\begin{aligned} \tan\theta &= \frac{-6}{6} \\ &= -1 \\ \theta &= \tan^{-1}(-1) \\ &= -45^\circ \end{aligned}$$

Finding direction: example

- Solution

1. First find the position vector $\mathbf{u} = \begin{bmatrix} 6 \\ -6 \end{bmatrix}$
2. The direction is given by the angle θ: θ = -45°
3. The angle terminates in the fourth quadrant, so 360° is added to obtain a positive angle: −45°+360°=315°.



Answer: θ = 315°

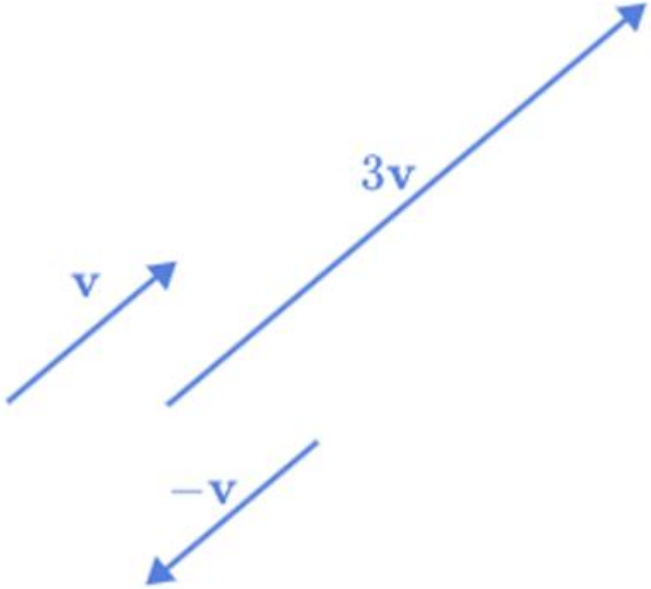
Vector: magnitude + direction

- Now each vector, previously defined by the coordinates of its terminal point, can also be defined as a combination of magnitude and direction.

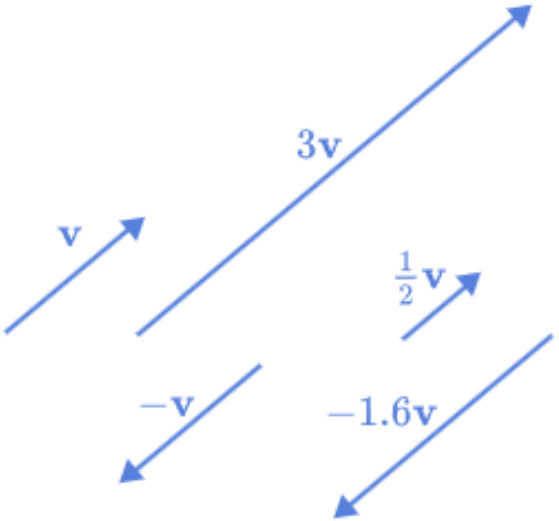
Vector operations

Multiplication by scalar

- A *scalar* is any real number.
- *Scalar multiplication* is a vector operation defined by multiplying a vector by a number: each component is rescaled by the same amount
- Geometrically, multiplication by a positive scalar multiplies the magnitude of the vector, and the resulting vector still points to **the same direction**.
- Multiplying by a negative scalar works the same way but -1 reverses the direction of a vector.



Scalar multiplication



- Any vector can be scaled to any magnitude without changing its direction.

Addition/subtraction

Given the vectors $\mathbf{m} = \begin{bmatrix} 3 \\ -2 \\ 5 \end{bmatrix}$ and $\mathbf{n} = \begin{bmatrix} -6 \\ 0 \\ 1 \end{bmatrix}$.

$$\mathbf{m} + \mathbf{n} = \begin{bmatrix} 3 \\ -2 \\ 5 \end{bmatrix} + \begin{bmatrix} -6 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 + (-6) \\ -2 + 0 \\ 5 + 1 \end{bmatrix} = \begin{bmatrix} -3 \\ -2 \\ 6 \end{bmatrix}$$

- Vectors are added componentwise.
- That means that the corresponding components are added to each other resulting in a new vector with the same number of components.
- Vectors are subtracted componentwise as well.

Combining operations

- Vector addition and subtraction can be combined with scalar multiplication:

Given the vectors $\mathbf{m} = \begin{bmatrix} 3 \\ -2 \\ 5 \end{bmatrix}$ and $\mathbf{n} = \begin{bmatrix} -6 \\ 0 \\ 1 \end{bmatrix}$.

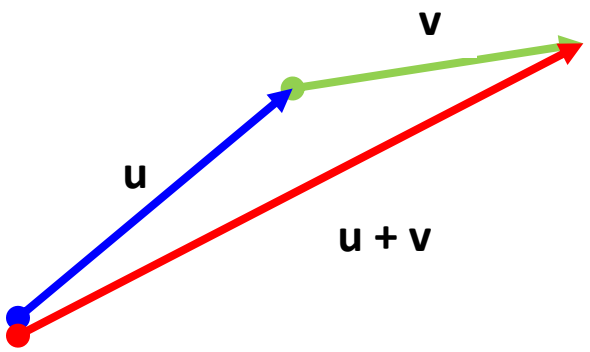
$$\mathbf{m} + \mathbf{n} = \begin{bmatrix} 3 \\ -2 \\ 5 \end{bmatrix} + \begin{bmatrix} -6 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 + (-6) \\ -2 + 0 \\ 5 + 1 \end{bmatrix} = \begin{bmatrix} -3 \\ -2 \\ 6 \end{bmatrix}$$

$$\mathbf{m} - \mathbf{n} = \begin{bmatrix} 3 \\ -2 \\ 5 \end{bmatrix} - \begin{bmatrix} -6 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 - (-6) \\ -2 - 0 \\ 5 - 1 \end{bmatrix} = \begin{bmatrix} 9 \\ -2 \\ 4 \end{bmatrix}$$

$$2\mathbf{m} - 4\mathbf{n} = 2\begin{bmatrix} 3 \\ -2 \\ 5 \end{bmatrix} - 4\begin{bmatrix} -6 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ -4 \\ 10 \end{bmatrix} - \begin{bmatrix} -24 \\ 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 30 \\ -4 \\ 6 \end{bmatrix}$$

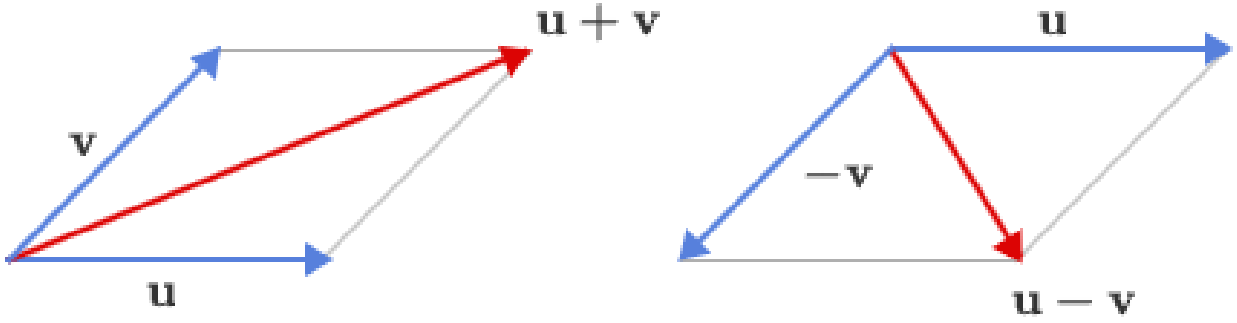
Addition: geometric interpretation

- Geometrically you can think of this as adding initial point of the second vector **v** to the terminal point of the first vector **u**
- This is as if you first follow the steps defined by vector u, and then continue by the directives of vector v.
- The result is a vector that brings us exactly into the same place as these two steps.



Addition: parallelogram rule

- The sum of two vectors is given by the diagonal of the parallelogram created by using the two vectors as adjacent sides.



Notations for sum and product

- Before we continue let’s see how we express operations on vectors in a space with an arbitrary number of dimensions.

Capital Sigma

The summation over the collection of *n* elements $X = \{x_1, x_2, \ldots, x_{n-1}, x_n\}$ is denoted using capital Sigma:

$$\sum_{i=1}^{n} x_i \overset{\text{def}}{=} x_1 + x_2 + \cdots + x_n$$

Value of a vector component across dimension *m*

The same applies to the *m* components of a vector $\mathbf{x} = [x^{(1)}, x^{(2)}, \ldots, x^{(m-1)}, x^{(m)}]$ in R^m.

The notation “def” means “is defined as”.

Capital Pi

A notation analogous to capital Sigma is the capital Pi notation. It denotes a product of elements in a collection or components of a vector.

$$\prod_{i=1}^{n} x_i \overset{\text{def}}{=} x_1 \cdot x_2 \ldots x_n$$

Dot product

- The *dot product* of two vectors **u** and **v** in the same n-dimensional space, denoted by **u**·**v**, is the sum of the products of corresponding components.
- Symbolically, given vectors $\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$; $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$,
- The dot product is defined as:

$$\mathbf{u} \cdot \mathbf{v} \overset{\text{def}}{=} \sum_{i=1}^{n} u_i v_i = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n$$

Dot product: example

$$\begin{bmatrix} 5 \\ 0 \\ -2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -3 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} 5 \\ 0 \\ -2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -3 \\ 4 \end{bmatrix} = 5(1) + 0(-3) + (-2)(4)$$

$$= 5 + 0 + (-8)$$

$$= -3$$

Algebraic laws for vectors

- Vectors obey algebraic laws similar to the algebraic laws on scalars

For vector addition:

For vectors $\vec{u}, \vec{v}, \vec{w} \in R^k$ and scalars *p* and *q* we have:

The Commutative Law of Addition::

$$\vec{u} + \vec{v} = \vec{v} + \vec{u} \qquad (1)$$

The Associative Law of Addition:

$$(\vec{u} + \vec{v}) + \vec{w} = \vec{u} + (\vec{v} + \vec{w}) \qquad (2)$$

The Associative Law for Scalar Multiplication
The Distributive Law over Vector Addition
The Distributive Law over Scalar Addition

For dot product:

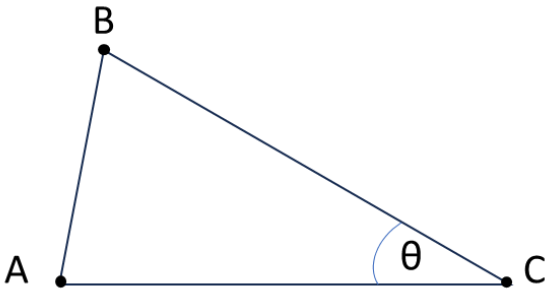
Commutative property
Distributive property
Associative property

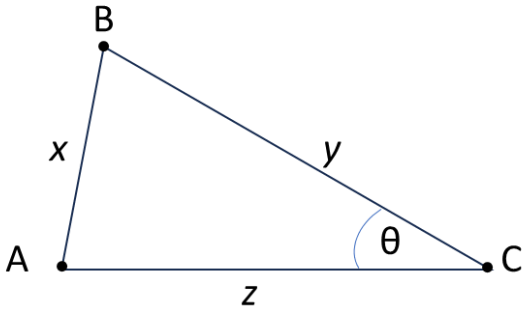
Angle between two vectors
Trigonometry/geometry snippet

Theorem (Cosine rule=Law of cosine): the square of the length of any side of a given triangle is equal to the sum of the squares of the length of two other sides minus twice the product of their lengths multiplied by the cosine of an angle between them.

- Let ABC be an arbitrary triangle. Let |AB|=x, |BC|=y, and |AC}=z. Also let the angle ∠BCA=θ.
- We want to show that:

$$x^2 = y^2 + z^2 - 2yz \cos\theta$$





Cosine rule

$x^2 = z^2 + y^2 - 2yz \times \cos\theta.$

- We will use the Cosine Rule to compute the cosine of an angle between two vectors.

Angle between two vectors

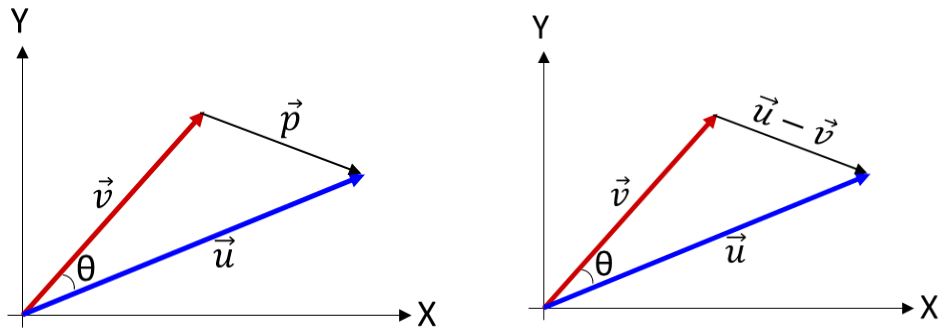
Let $\vec{v}$ and $\vec{u}$ be two arbitrary vectors and let their magnitudes be v and u respectively.

Then the cosine of an angle between them can be computed as the dot product of $\vec{v}$ and $\vec{u}$ divided by the product of their magnitudes:

$$cos\,\theta = \frac{\vec{v} \cdot \vec{u}}{vu}$$

- We will show that this is true in R^2, but the same holds for any number of dimensions.

Computing angle between two vectors: 2/2



So we have two expressions for the same dot product $\vec{p} \bullet \vec{p}$: (1) and (2). We write an equation:

$$u^2 + v^2 - 2\vec{u} \bullet \vec{v} = u^2 + v^2 - 2uv \cdot \cos\theta$$

This gives:

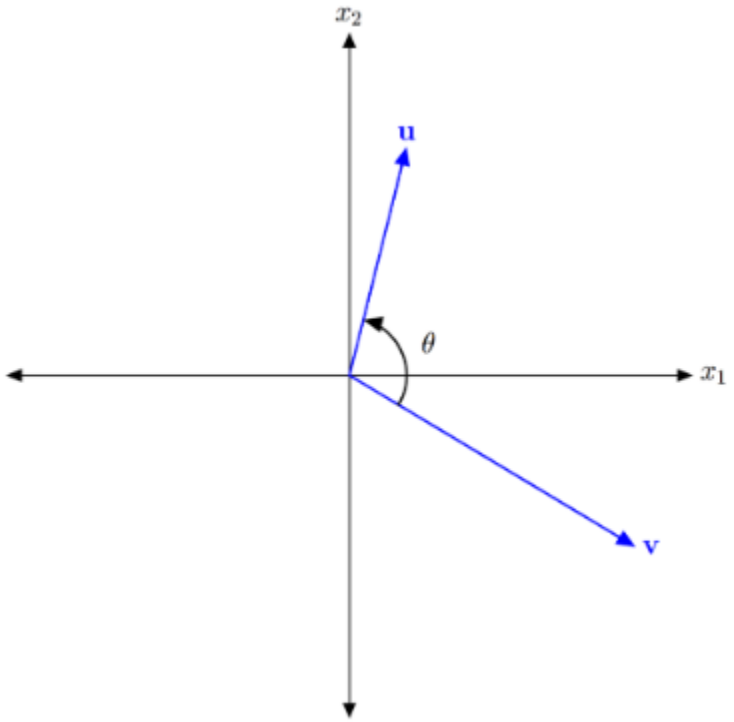
$$\vec{u} \bullet \vec{v} = uv \cdot \cos\theta$$

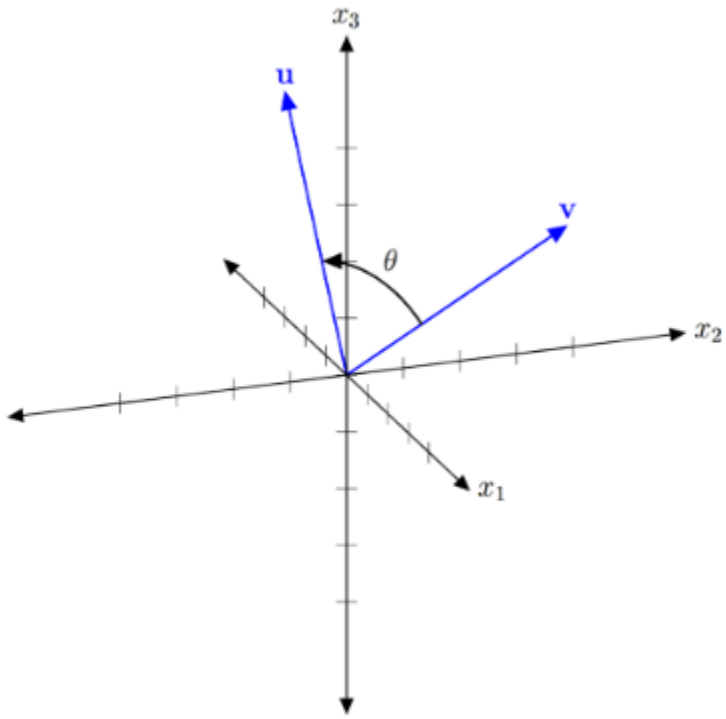
And we can compute θ as dot product divided by the magnitudes of vectors $\vec{u}$ and $\vec{v}$:

$$\cos\theta = \frac{\vec{u} \bullet \vec{v}}{uv}$$ ■

The cosine of the angle between vectors

$$\cos\theta = \frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}.$$





Angle between two vectors: example

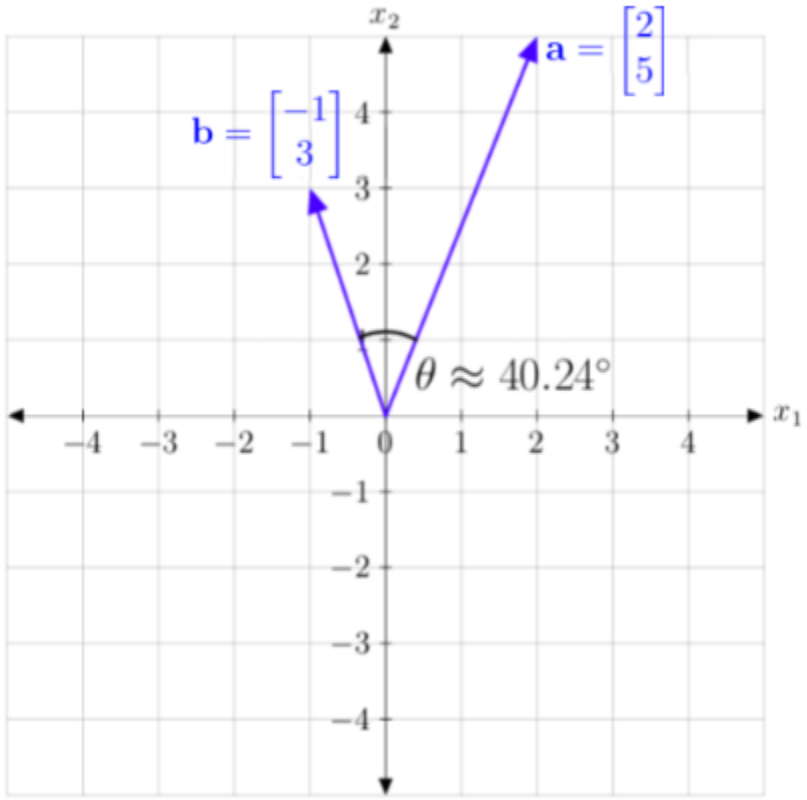
$$\mathbf{a} = \begin{bmatrix} 2 \\ 5 \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} &= 2(-1) + 5(3) \\ &= 13 \end{aligned}$$

$$\begin{aligned} \|\mathbf{a}\| &= \sqrt{2^2 + 5^2} \\ &= \sqrt{29} \end{aligned} \qquad \begin{aligned} \|\mathbf{b}\| &= \sqrt{(-1)^2 + 3^2} \\ &= \sqrt{10} \end{aligned}$$

$$\cos\theta = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} = \frac{13}{\sqrt{29}\sqrt{10}}$$

$$\theta = \cos^{-1}\left(\frac{13}{\sqrt{29}\sqrt{10}}\right) \approx 40.24^\circ$$



Orthogonal vectors

- The angle between orthogonal vectors is 90^o.
- Cosine of 90^o is 0.
- Two vectors are orthogonal if and only if the dot products of the vectors is 0.

Orthogonal vectors: example

- Given vectors **u** and **v** below, find the value of *a* such that **u** and **v** become orthogonal.

$$\mathbf{u} = \begin{bmatrix} 20 \\ -4 \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} a \\ 5 \end{bmatrix}$$

Solution

$$\begin{aligned} \mathbf{u} \cdot \mathbf{v} &= 0 \\ 20a + (-4)(5) &= 0 \\ 20a &= 20 \\ a &= 1 \end{aligned}$$

Generalization to multi-dimensional spaces

- We learned how to put vectors into a standard position, how to find vector magnitude and direction, and how to compute an angle between two vectors.

- We were mostly using R^2 but the same ideas hold for any number of dimensions.

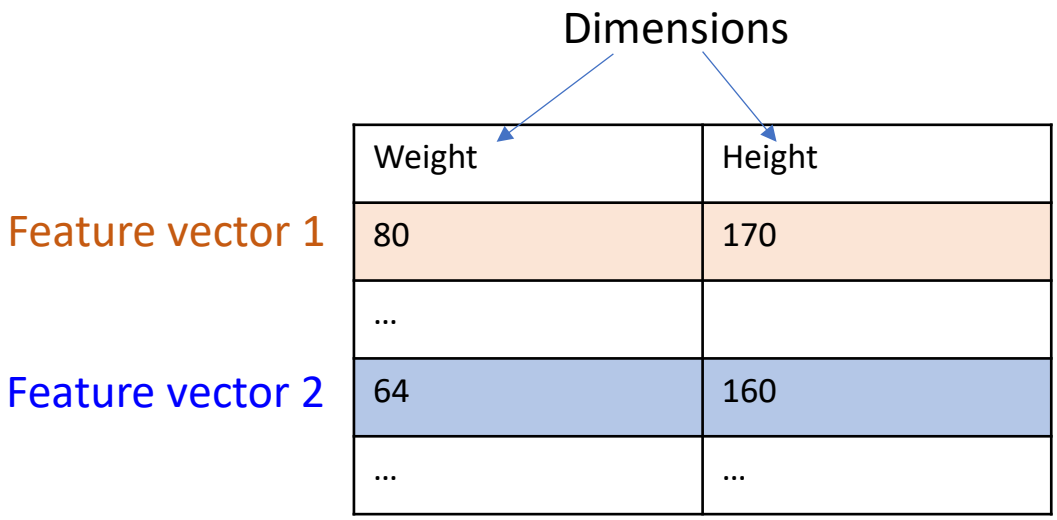
Distance between data records

Vector interpretation

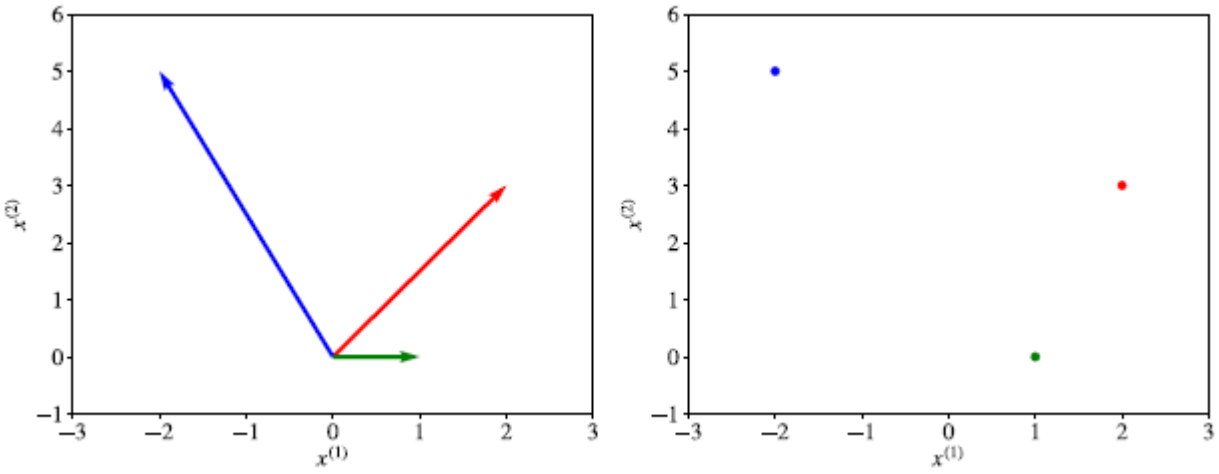
Data as feature vectors

- We see that “vector” is essentially just a list of *k* scalars that can be thought of as representing a positional vector in space R^k
- In machine learning, the vectors we speak of are “feature vectors”: the list of numbers we work with is a **row** in a dataset, and each row is a vector whose components are the values along each dimension (= feature = column = attribute).

Feature vectors: example



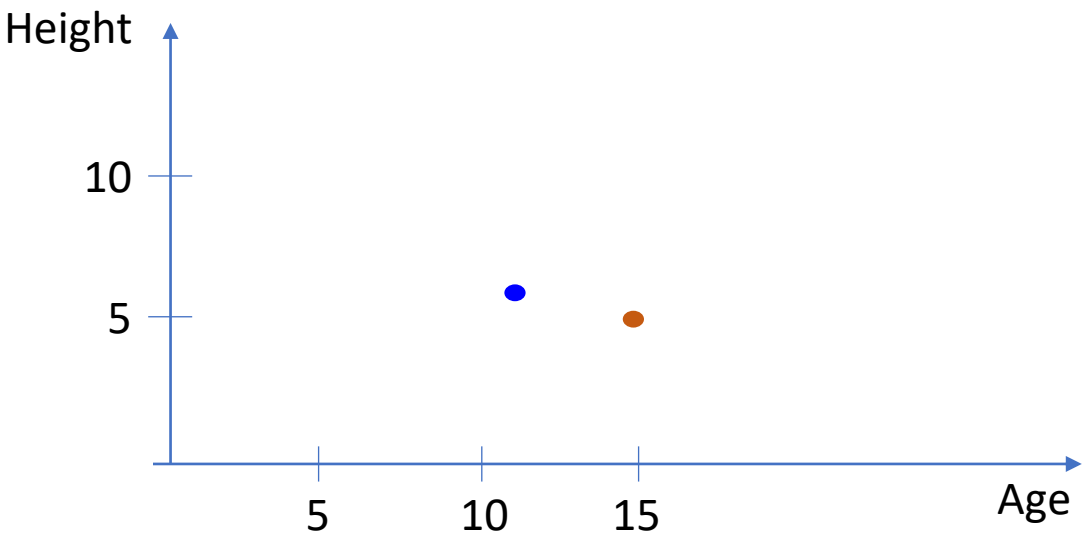
Vectors can be visualized as positional vectors (with direction) or as points in R^n



Feature vectors: visualized as points

Dimensions

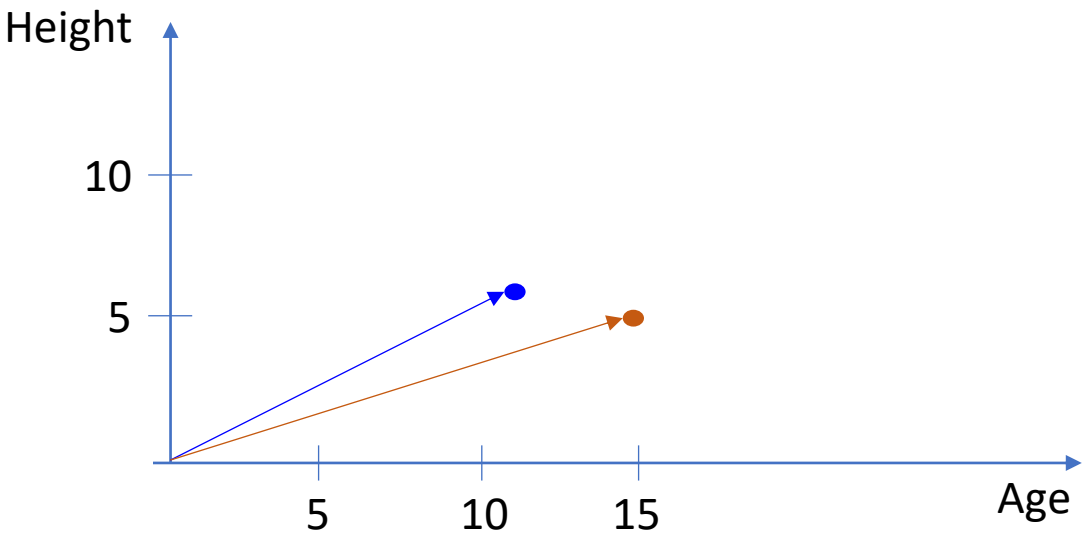
	Age	Height
Feature vector 1	15	5
	…	
Feature vector 2	12	6
	…	…



Feature vectors: visualized as position vectors

Dimensions

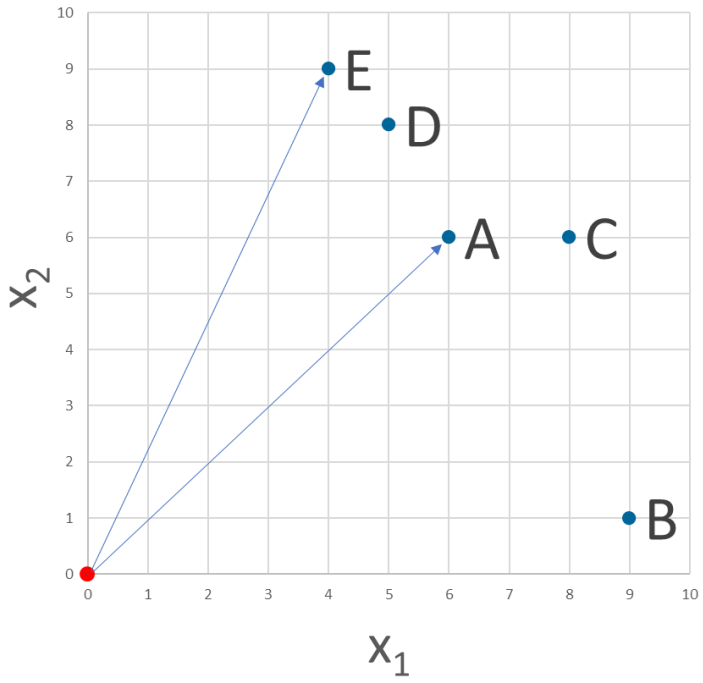
	Age	Height
Feature vector 1	15	5
	…	
Feature vector 2	12	6
	…	…



Coffee shops can also be visualized as vectors

- Each shop is a terminal point of the corresponding position vector.

Store name	x_1	x_2
A	6	6
B	9	1
C	8	6
D	5	8
E	4	9



Coffee shop vectors in 3D

Store name	x1	x2	x3
A	6	6	4
B	9	1	1
C	8	6	3
D	5	8	1
E	4	9	2

$$v=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \mathbf{a}=\begin{bmatrix} 6 \\ 6 \\ 4 \end{bmatrix} \quad \mathbf{b}=\begin{bmatrix} 9 \\ 1 \\ 1 \end{bmatrix} \quad \mathbf{c}=\begin{bmatrix} 8 \\ 6 \\ 3 \end{bmatrix}$$

New distance metric: Cosine similarity

- Sometimes it makes more sense to consider two records closely associated because of similarities in the way the attributes *within each record are related*

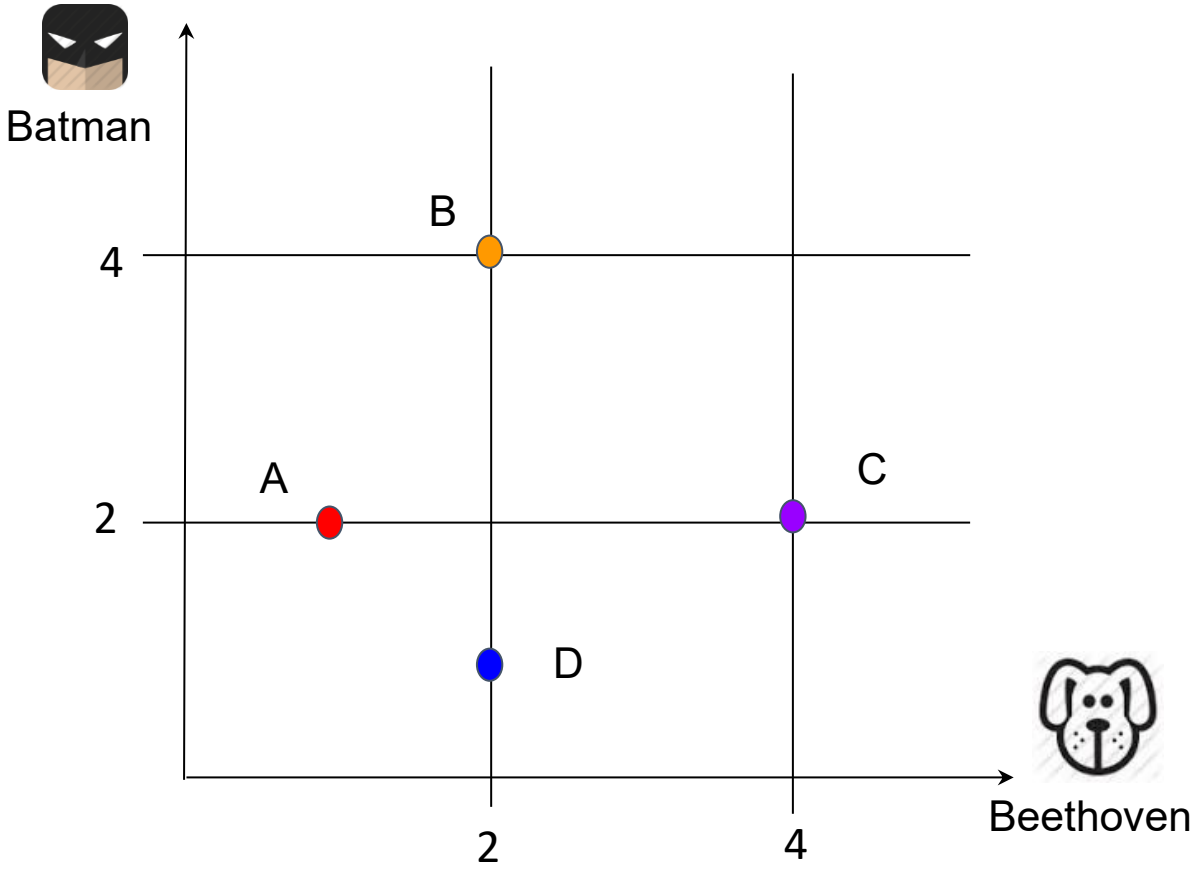
Movie rating dataset

	Spiderman	Beethoven	Star Wars	Shrek	Wish Dragon	The Batman
Friend A	4	3	5	2	3	3
Friend B	5	5	5	5	4	5
Friend C	3	4	2	4	3	2
Friend D	4	3	4	4	4	4
Friend E	2	3	2	3	2	2

- Note how some people tend to give more generous ratings than the others.

- This is an example of grade/rating inflation

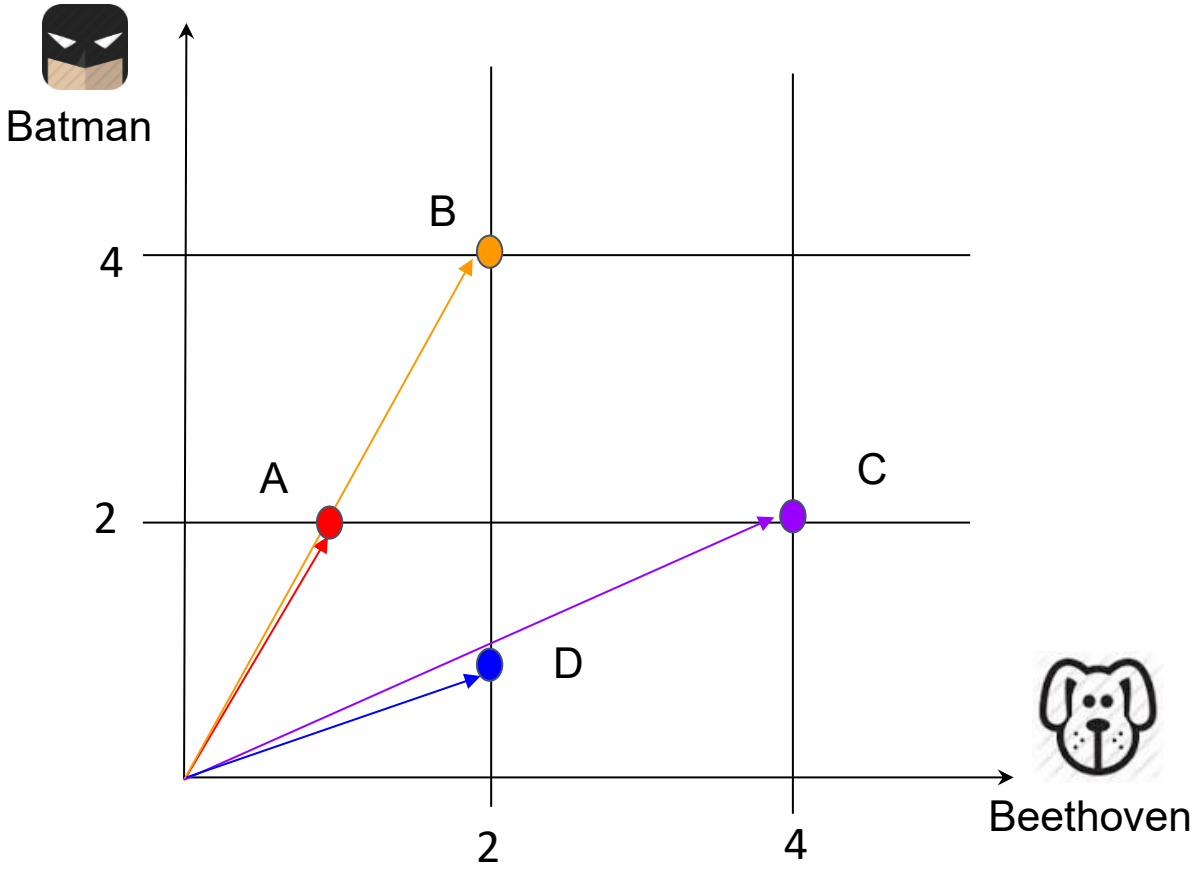
Example: Coordinate-based similarity
Euclidean distance



A: (1,2)
B: (2,4)
C: (4,2)
D: (2,1)

Movie rankings in 2D

Example: Vector-based similarity angle between vectors



A: (1,2)
B: (2,4)
C: (4,2)
D: (2,1)

Movie rankings in 2D

Cosine similarity as a distance metric

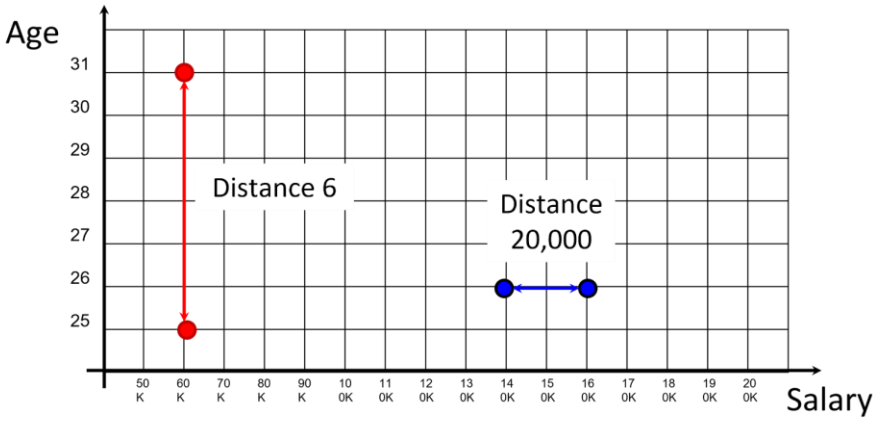
- To evaluate the similarity of two feature vectors we can use the angle between them.
  - If the angle is smaller, then the two vectors point into more or less the same direction.
  - If the angle is large, then these vectors represent a set of attributes that are radically different.

- We can define the distance using a concept of *Cosine similarity*.
  - If we compute a cosine of an angle between two vectors, then the closer cosine is to 1, the **more similar** these two vectors are, and the **smaller** is the **distance** between two data vectors.

Scaling numeric dimensions

Mapping all dimensions to a common range

- To place feature vectors in a common space, we need to be careful about the scale of different dimensions.
  - Example: Difference in 1 dollar = difference in 1 child?



- Min-max scaling: map all variables to a common range 0-1:

$$a_i = \frac{v_i - \min(all\ v)}{\max(all\ v) - \min(all\ v)}$$

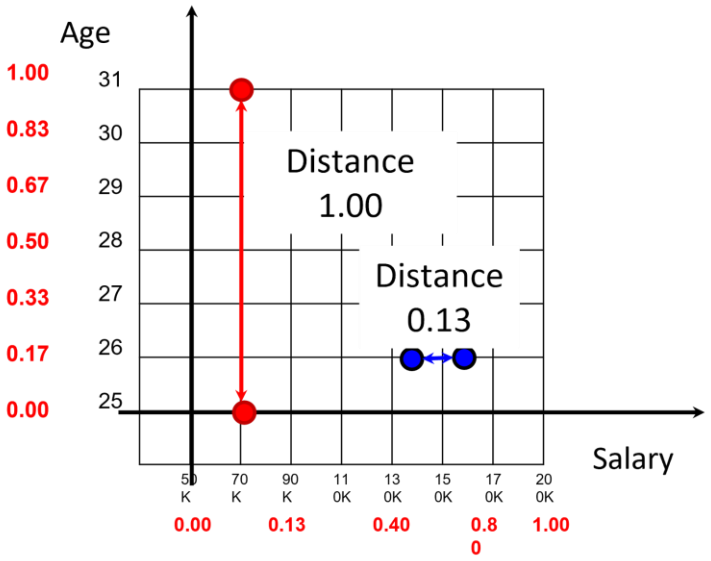
Same dimensions after scaling

$$a_i = \frac{v_i - \min(all\ v)}{\max(all\ v) - \min(all\ v)}$$

For Age: $a_i=(v_i-25)/(31-25)$

For Salary: $a_i=(v_i-50{,}000)/(200{,}00-50{,}00)$

- All values are now between 0.00 and 1.00



Dealing with non-numeric attributes

- Data tuple ≠ data vector: Not all values in a data tuple are expected to be numeric.

Recap: types of attributes

- True measures = numeric = quantitative can take on a numeric value that can be measured and ordered:
  - Discrete - finite within a range, typically integers
  - Continuous - infinite along a continuum of values within a range, typically real numbers

- True measures measure the value from a meaningful “0” point
- The ratio between values is meaningful
  - Examples: age, weight, length

- These are easy to deal with for computing distance

Ordinal attributes

- Ranks = ordinal. Take on a value from a set of categories, but these categories have an ordering (ex. disagree, neutral, and agree).
- These values have an order, but the distance between different ranks is not well defined
  - Example: quality of a product : {poor, fair, OK, good, wonderful}
- Order is important, but exact difference between values is undefined
- We could map the ordinal values to successive integers:
  - {poor=0, fair=1, OK=2, good=3, wonderful=4}

- This is the best we can do

Nominal attributes

- Categorical = nominal = qualitative can take on the value (usually a label) of one of several categories.
- They have no ordering, existing in name only
  - ex. apples, oranges, and grapes
- Each value is one of a set of unordered categories.
- We can only tell that X≠Y, but not how much X is greater than Y.

- The only solution: one hot encoding
  - To convert categorical attributes into numeric we add as many new attributes as there are different categories
  - For each tuple we enter either 1(True) or 0(False) for the corresponding attribute
  - If these are combined with other attributes because they are now numeric, we can use one of the three distances for numeric attributes.

Binary attributes

- Sometimes values of attributes are represented as binary: either *yes* or *no*.
  - Example: like/dislike. Presence/absence of words.

- We can encode 1 for *yes* and 0 for *no* and treat them as numeric

Distance between data records

Set interpretation

All attributes are binary

- If all attributes are binary, then we can use a different metric for similarity: similarity based on sets.

Matching coefficients. All attributes are binary

	Y	Y
X	M_{11}	M_{10}
X	M_{01}	M_{00}

M_{11}: number of attributes with value 1 in both X and Y
M_{10}: number of attributes with value 1 in X and 0 in Y
M_{01}: number of attributes with value 0 in X but 1 in Y
M_{00}: number of attributes with value 0 in both X and Y

Matching coefficients and Jaccard index

	Y	Y
X	M_{11}	M_{10}
X	M_{01}	M_{00}

Jaccard index is used for asymmetric binary attributes, where only presence (1) is important

Simple Matching Coefficient

SMC = number of matches / number of all attributes (dimensions)

= $(M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

Jaccard Index

J = number of M_{11} matches / number of not-both-zero attribute values

= $(M_{11}) / (M_{01} + M_{10} + M_{11})$

SMC and Jaccard example

x=(	1	0	0	0	0	0	0	0	0	0)
y=(	0	0	0	0	0	0	1	0	0	1)

$SMC = (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00}) = (0+7)/10=0.7$

$J = M_{11} / (M_{01} + M_{10} + M_{11}) = (0)/3=0.0$

The choice is application-dependent

SMC and Jaccard example

x=(	1	0	0	0	0	0	0	0	0	0)
y=(	0	0	0	0	0	0	1	0	0	1)

$SMC = (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00}) = (0+7)/10=0.7$

$J = M_{11} / (M_{01} + M_{10} + M_{11}) = (0)/3=0.0$

The choice is application-dependent

Which measure to choose for:

- Comparing documents by common words?
- Comparing transactions by common items?
- Comparing students by knowledge of 10 topics?

Tanimoto similarity coefficient

- Jaccard index is defined as the number of attributes with value 1 in both records, divided by the total number of records for which there is at least one 1 value:

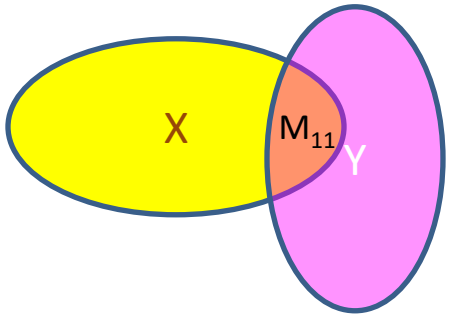
$J = M_{11} / (M_{01} + M_{10} + M_{11})$

- Tanimoto coefficient is similar but is defined in terms of set operations: it is an intersection over union of all attribute values without attributes for which both binary values are False(0):

$T = M_{11} / (M_{-1} + M_{1-} - M_{11})$

The formulas show that Jaccard and Tanimoto are **exactly the same**!

	Y	Y
X	M_{11}	M_{10}
X	M_{01}	M_{00}



M_{11} ← intersection

$(M_{01} + M_{10} + M_{11}) = (M_{-1} + M_{1-} - M_{11})$

union

Summary of proximity metrics

- We now have several distance (similarity) metrics for two tuples in a multi-dimensional space defined by their attributes.
  - Manhattan distance
  - Euclidean distance
  - Cosine similarity
  - Matching coefficients: symmetric and asymmetric.

- We have learned that no matter the type, we can convert any tuple to a feature vector, and to compute a distance between any pair.

- Thinking of data tuples as vectors in multidimensional space is a very fruitful approach.