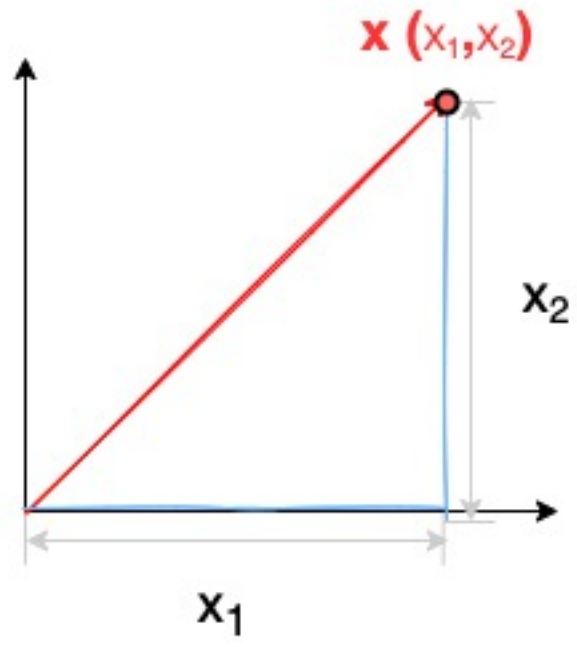
Numpy Tutorial and Cosine Similarity

Numpy

- Open notebook in github repo using google colab
  - Go to file ‘import notebook’ and pass in the github repo url
- Or create a virtual env in which you install jupyter and numpy
  - Python3 –m venv __name__
  - Source __name__/bin/activate
  - Pip install jupyter numpy
  - Deactivate (to leave env)
- You can also use the conda command but you need to install miniconda first
- Then you can do :
  - Conda create --name __name__ python=3.9
  - Conda activate __name__
  - Conda install jupyter numpy
  - conda deactivate (to leave env)

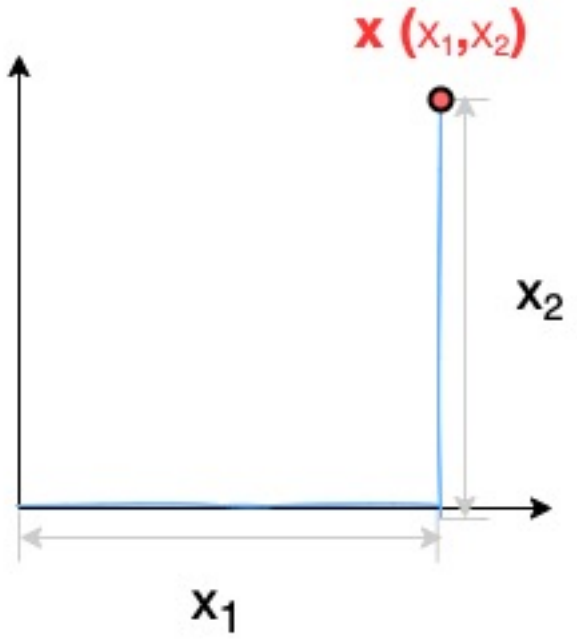
Vector Norm – Euclidian Norm

- How can we calculate the length of a vector ?
- $\| \boldsymbol{a} \| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}$
- Euclidian norm or L^2 norm
- Measures the shortest distance rom the origin



Manhattan Norm

- $\| \boldsymbol{a} \| = |a_1| + |a_2| + \cdots + |a_n|$



- L_1 norm or Manhattan norm

- Sum of the absolute values of the components of the vector

Law of Cosines and the Dot Product

- Now you know about vector norms, another way of expressing the dot product that we haven't seen is :

$$\mathbf{a} \cdot \boldsymbol{b} = \|\boldsymbol{a}\|\|\boldsymbol{b}\| cos\theta$$

- This comes from the law of cosines

- See here for proof !

New intuition about the dot product

- The result of the dot product is therefore impacted by
  - the vectors' magnitude
  - their direction/the angle between them

- When $\theta < 90°$ dot product is positive
- When $\theta = 90°$ dot product = 0
- When $90° < \theta < 180°$ dot product is negative

Cosine Similarity

- Cosine similarity measures the similarity between two vectors.
- It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction.
- It is often used to measure document similarity in text analysis.
- https://www.sciencedirect.com/topics/computer-science/cosine-similarity

Cosine Similarity

$$Similarity(\boldsymbol{a}, \boldsymbol{b}) = \cos\theta = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}$$

- Remember

$$\mathbf{a} \cdot \boldsymbol{b} = \|\boldsymbol{a}\|\|\boldsymbol{b}\| cos\theta$$

Transfroming sentences into vectors

- Now we've seen these fancy mathematical tools, we want to use them !
- But how de we transform text into numerical data…?
- One of the easiest ways to do this is to represent a sentence or document with attributes (or vector dimensions) which record the frequency of the words that appear in that particular sentence/doc !
- Each document is then represented by an object called a **term-frequency vector.**

Term-Frequency Vector Example

Document	team	coach	hockey	baseball	soccer	penalty	score	win	loss	season
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

https://www.sciencedirect.com/topics/computer-science/cosine-similarity

- Do you see any potential problems with such a representation ?

- These vectors are very *sparse* :
  - They are full of zeros !
  - And usually very long

- In order for the dot product to work, the vector representations must have the same number of dimensions, which typically equals the number of words found in the vocab used…

$$\mathbb{R}^{vocab}$$

Calculating the cosine similarity between 2 term-frequency vectors by hand

$$\boldsymbol{x} = \begin{bmatrix} 3 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

- How similar are the vectors $\boldsymbol{x}\ and\ \boldsymbol{y}$?

Implementing cosine similarity with numpy

- See the github repo