Lecture 4: Introduction to Data Science

Data Representation: How we represent the data might completely determine what information can be extracted from that data.

Data Representation Types

How can we represent data? There are many different ways:

Records

These are m-dimensional points or vectors, often represented as a tuple. For example, (name, age, balance) -> ("John", 20, 100). We can draw these vectors in their dimensional space.

Graphs

Another type of data representation is a graph. These show up in a number of use cases, like social networks. Here, these is some kind of connection between nodes - each node is an individual part of the network, and edges represent connections between those entities.

There are two common ways of representing a graph:

- Adjacency Matrix: n x n matrix, where a 1 in a (row,col) pair means there exists an edge between the node at row and the node at col, and zero means no edge. There is some redundancy in the adjacency matrix, since every (row/col) pair is listed again as its corresponding (col/row) pair.
- Adjacency list: For each node, just list the other nodes that said node is connected to.

Because of the redudancy in the space used in the adjacency matrix, we often use adjacency lists instead to represent graphs.

Images

Images are reprsented as pixels, each pixel has some color and location. Pixels may in grayscale (value between 0 and 1), or perhaps in rgb - how much of red, green, and blue each pixel contains. For simplcity, imagine that each pixel just encodes a number that indicates how much color it has.

Text

Textual data is another way of representing textual data. Usually, we're interested in the words inside the text, perhaps as a list of words. Sometimes, we'll see something a bit more advanced: like the root of the word instead of the word itself.

Strings

These are quite similar to texts, except that these are sequences of charactesrs instead of words, like DNA sequences. Here, hte characters themselves are of interest, instead of just the words

Time Series

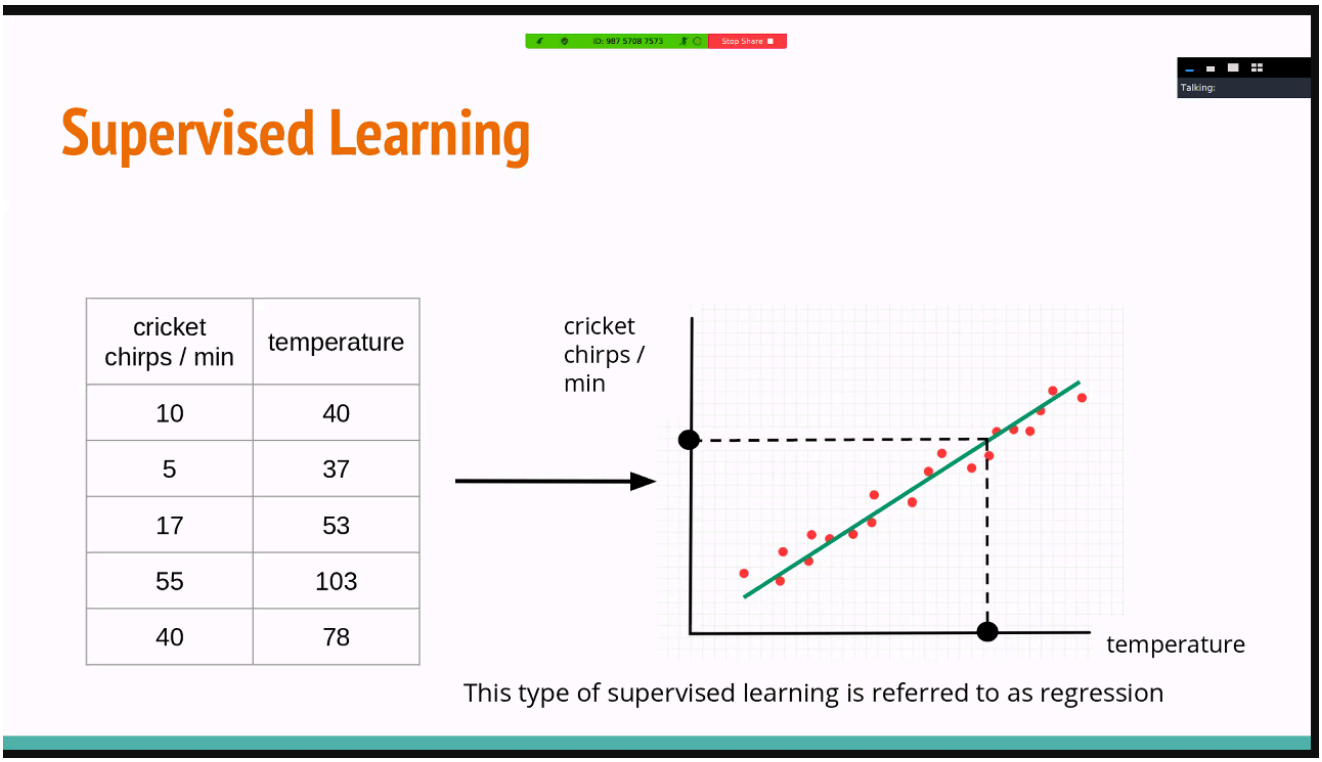
This is defined in a bit of a meta way, where it's data at a specific time interval. Videos - images at specific times, stock prices: value of a stock at each step in time, etc. It is effectively a list of data at specific intervals of time.

Types of Learning

We'll talk about two different types of learning: supervised and unsupervised learning.

Supervised Learning

Imagine we have a list of records - the number of cricket chirps that appear every minute, and then we are recording the temperature. From the data, we can construct a linear model - this is called **regression** What's great about this model, is taht given a temperature, we can estimate how many cricket chirps there will be.



The goal here is pretty straightforward: estimate the number of cricket chirps given the temperature.

There is another type of supervised learning called **classification**. For example, we may be trying to decide whether or not a tumor is malignant or benign. Then, when we see a new tumor, we'd like to be able to say with a certain degree of confidence whether or not the new tumor is malignant or benign.

Unsupervised Learning

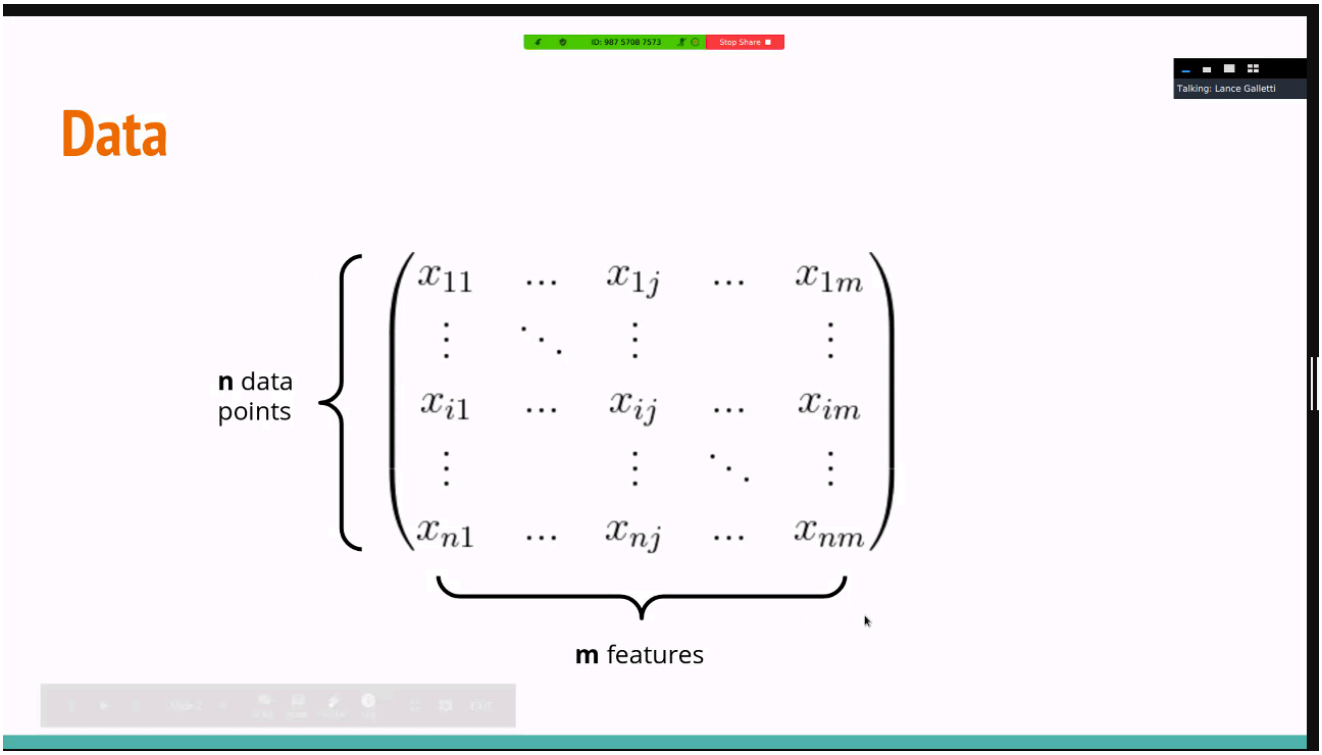
Here, there isn't necessarily a specific goal in mind. Instead, the goal is more meta: find interesting structure in the data. basically, look at the data, and see if we can find some interesting underlying structure to the data.

One way of doing this is called **clustering** - a bunch of data points are "close together," and therefore perhaps these data points are related in some manner.

For example, given a dataset that is a collection of articles - we can ask, do these articles cover the same topic?

Distance and Similarity

Here, we want to build up some sense of how "similar" different data points are, or perhaps how "far apart" they exist.



From the data that we get, we can generate a feature space of all possible values for the set of features in our data. For example, we may not really care about names (it's not really meaningful for what we're trying to do), but we may be interseted in age and balance. In this case, our feature space is the Euclidean plane.

in order to uncover interesting structure from our data, we need a way to compare data points. A **dissimilarity function** is a function that takes two objects (data points), and returns a large value if these objects are **dissimilar**. This can be any function really, as long as it adheres to this rule. With a dissimilar function, we can then begin to compare data points. This is in line with the distance function.

A distance function has three properties it needs to satisfy: **d** is a distance function *iff*

- $d(i,j) = 0$ if and only if $i = j$
- $d(i,j) = d(j,i)$
- $d(i,j) \leq d(i,k) + d(k,j)$ - this is also called the triangle inequality. That is, if you go through a third point, it must be at least as long as going from i to j directly. you can't magically go through a point k and have then traveled a lesser distance.

Why do we need a distance function? Before we said we needed a dissimilarity function! Well, the reason is that a distance function is *intuitive* - we can easily understand what a distance function means.

We're going to cover a number of different distance functions

Minkowski Distance

For x, y points in $d-$dimensional real space. For example, $x = [x_1, x_2, \ldots, x_d]$ and $y = [y_1, y_2, \ldots, y_d]$

Then, we can define a parameter $p \geq 1$, such that the minkowsky distance is:

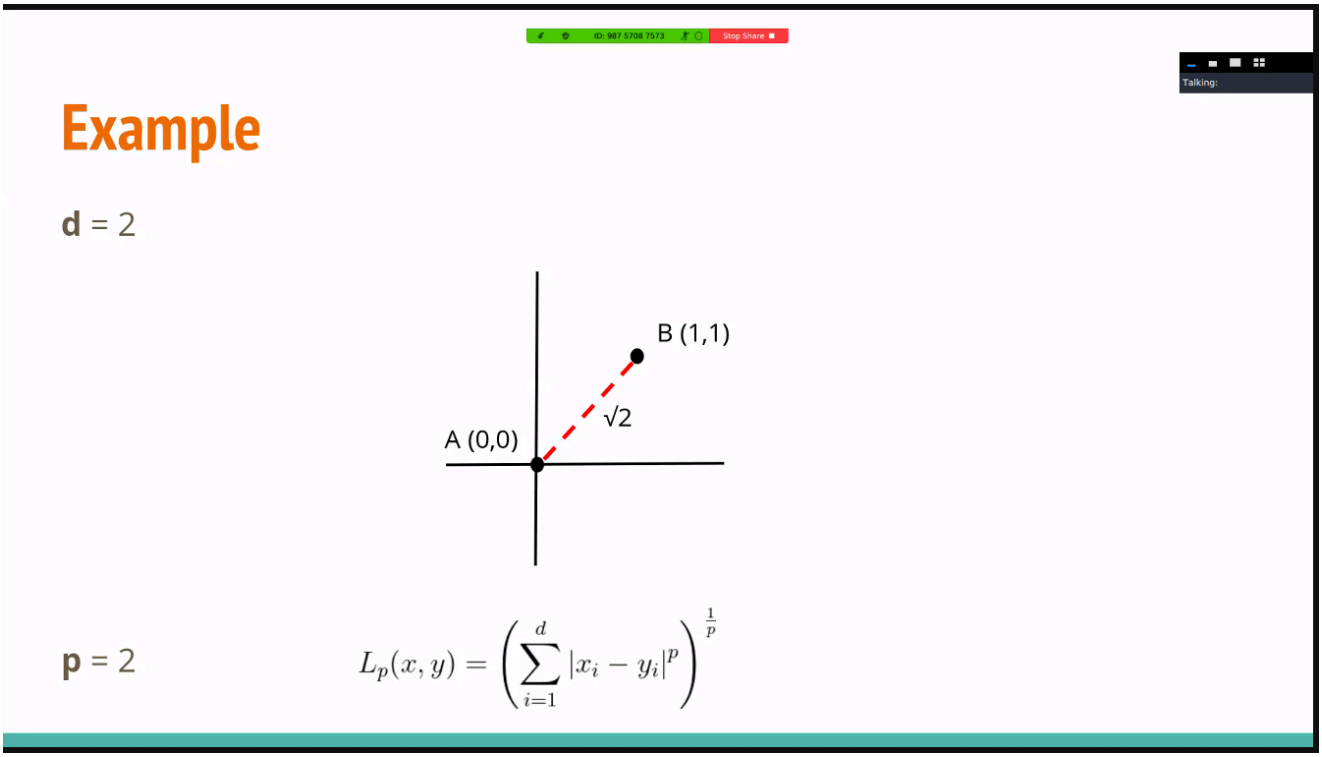
$$L_p(x,y) = (\Sigma_i^d |x_i - y_i|^p)^{1/p} \quad (1)$$

We do each x minus each y, raise that to the p power, sum that, then take all of that to the p^{th} root.

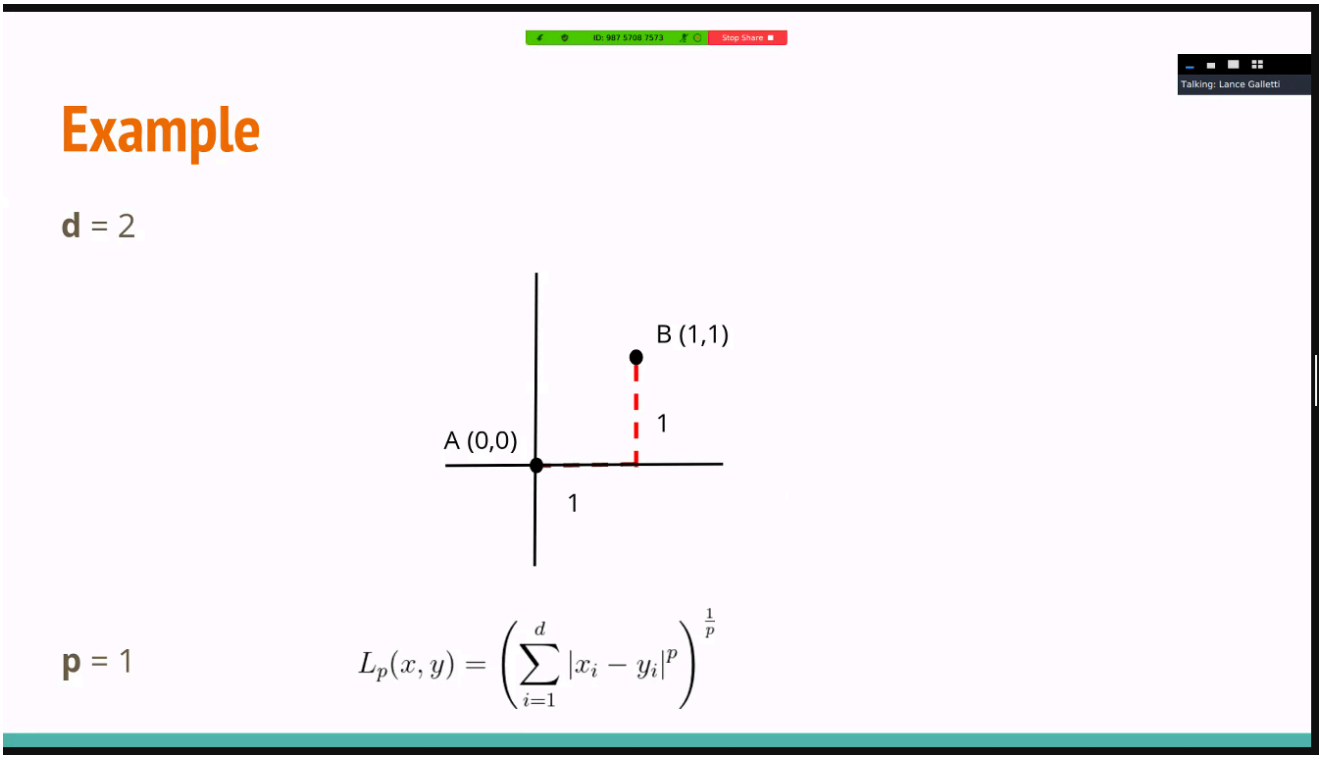
There are a couple special cases:

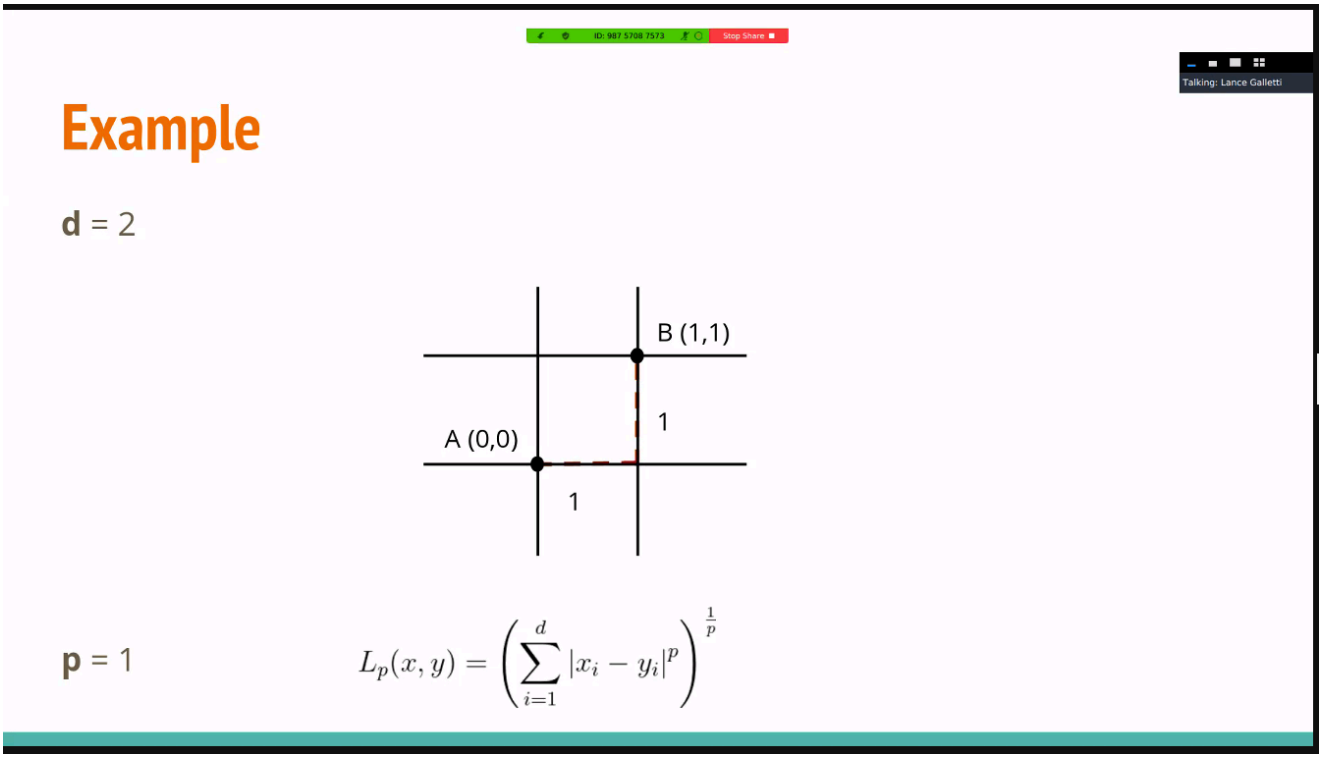
- When $p = 2$ we have the Euclidean distance (should be recognizable - for two points in Euclidean space on a triangle, this is just the Pythagorean theorem)
- When $p = 1$ we have the Manhattan distance

Example when we are in 2-dimensional space, with $p = 2$



What happens when $p = 1$?





It's called the manhattan distance because to get from one point to another, you have to follow a grid. In two-dimensional space, this is like a grid, where you can only travel vertically or horizontally, but not diagonally.

Missed the last slide :/

Cosine Similarity

this is another type of distance function. First, we need to define a **similarity function**: a function that returns a large value if two objects are similar.

$$s(x, y) = cos(\theta) \tag{2}$$

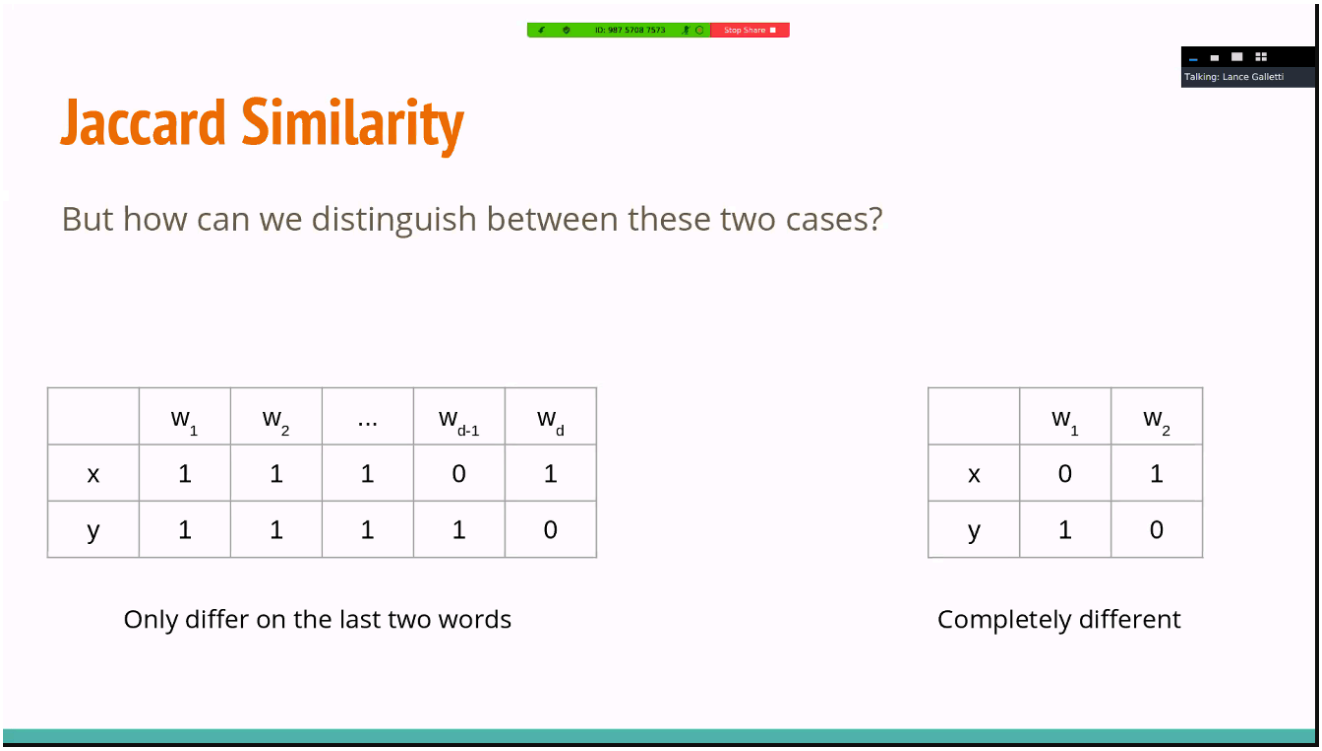
Where θ is the angle betwen x and y.

to get a corresponding dissimilarity function, we can try $d(x, y) = 1/s(x, y)$ or $d(x, y) = k - s(x, y)$ for some value k. Meaning, we can either divide 1 by the result (so a large value would result in a small value, and vise versa), or by simply subtracting the result from some number k.

When should we use cosine (dis)similarity over Euclidean distance? Well, it's when we're not interested in the magnitude of our vectors. For example, the abstract of a paper and teh content should be very similar, as they talk about similar topics, but their magnitudes are quite different because the abstract is small while the paper is big.

Jaccard Similarity

Imagine that x and y are two different documents. Then, if a word is present, we give x or y a 1, otherwise a 0. One way we can find the distance between x and y is to use the Manhattan distance, because this would return the size of the set difference between x and y (if they have the same word, the difference is 0, otherwise 1, so it's basically a count of the number of different words). However, consider the following case:



	w_1	w_2	...	w_{d-1}	w_d
x	1	1	1	0	1
y	1	1	1	1	0

	w_1	w_2
x	0	1
y	1	0

In each of these two cases, the Manhattan distance is 2. However, it is clear that these two instances are completely different! We need to also account for the size of the intersection. This is where the Jaccard similalrity comes in:

$$JSim(x, y) = \frac{|x \cap y|}{|x \cup y|} \tag{3}$$

This is the ratio between the intersection of x and y to their union. We can then define the Jaccard distance as:

$$JDist(x, y) = 1 - \frac{|x \cap y|}{|x \cup y|} \tag{4}$$