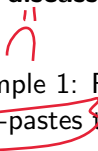


# CSCI 4022 Fall 2021

## Similarity and Distance

**Opening discussion:** which of the following represent violations of the course honor policy?

- 
- ▶ Example 1: For an assignment, Chris searches the internet for relevant codes and copy-pastes them into his Jupyter Notebook. He properly cites the source of the codes.
  - ▶ Example 2: For an assignment, Maciej and Felix work together to figure out how to implement the codes, but each works on their own computer and develops their own software.
  - ▶ Example 3: For an assignment, Rhonda has a plan for how to implement an algorithm, but isn't sure how to manipulate a Python list in a particular way that she needs to. She searches the internet, finds a fix, and implements it in her code without copying it.

# Announcements and To-Dos

## Announcements:

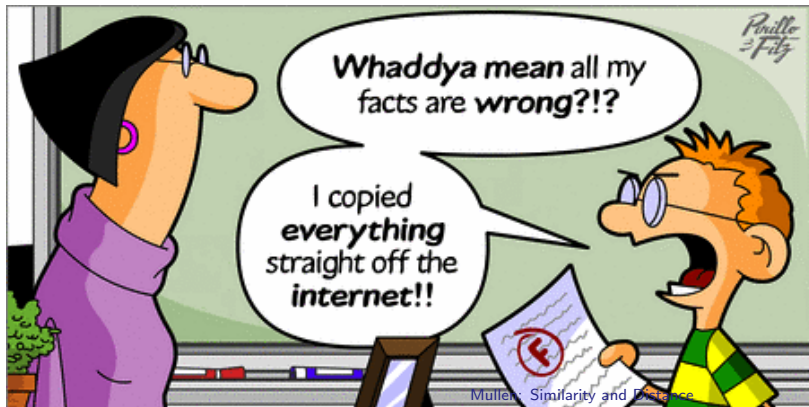
1. HW 1 posted by sometime this weekend... **ANSWER ZOOM POLL ABOUT DEADLINES**

## Before next class:

1. Make sure you can access the Canvas page and read the syllabus
2. Set up some way to back up your work
3. Install Anaconda (or other reliable Jupyter notebook method)
4. If you're new to P3, Numpy, or pandas, review and complete nb00 (intro to Python/Jupyter) and nb00a (a Numpy/Pandas tutorial)
5. If you're experienced with Python 3 and those packages, you can check out nb01, nb02 which we'll work on Friday!

## Academic Integrity

1. See the CU Academic Integrity Policy for more details. Here are some highlights.  
“Examples of cheating include: copying the work of another student during an examination or other academic exercise (includes computer programming)”
2. “Examples of plagiarism include: . . . copying information from computer-based sources”



## Integrity Examples

Example 1: For an assignment, Chris searches the internet for relevant codes and copy-pastes them into his Jupyter Notebook. He properly cites the source of the codes.

Example 2: For an assignment, Maciej and Felix work together to figure out how to implement the codes, but each works on their own computer and develops their own software.

Example 3: For an assignment, Rhonda has a plan for how to implement an algorithm, but isn't sure how to manipulate a Python list in a particular way that she needs to. She searches the internet, finds a fix, and implements it in her code without copying it.

## Integrity Examples

Example 1: For an assignment, Chris searches the internet for relevant codes and copy-pastes them into his Jupyter Notebook. He properly cites the source of the codes.

**Sol'n:** :( . Boo, Chris! Copy-pasting is still not your own work. Correct action: cite a resource, learn the pseudocode or general structure from it and re-create **from the ground up, yourself**.

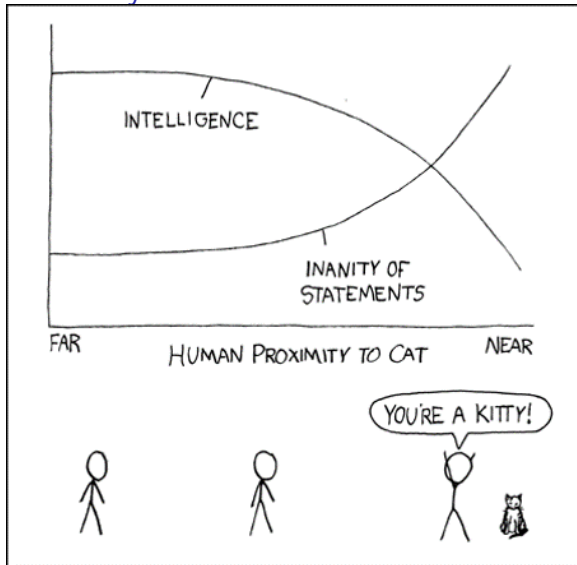
Example 2: For an assignment, Maciej and Felix work together to figure out how to implement the codes, but each works on their own computer and develops their own software.

**Sol'n:** Awesome! Work together, talk together, develop a theoretical solution together but don't copy the work.

Example 3: For an assignment, Rhonda has a plan for how to implement an algorithm, but isn't sure how to manipulate a Python list in a particular way that she needs to. She searches the internet, finds a fix, and implements it in her code without copying it.

**Sol'n:** Great!... but just to be safe, **cite** where you found the fix!

# Similarity and Distance



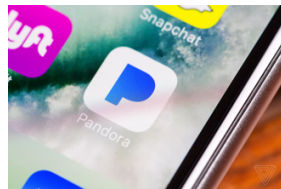
# Similarity

*Many* problems - both in this course and in general - can be expressed as the task of finding **similar** elements.

We often phrase this as finding “near-neighbors,” which may exist in **high-dimensional** spaces.

## Examples:

1. Documents with similar words/code
2. Users who watch similar movies
3. Songs that have similar attributes
4. Images with similar features
5. etc., etc.



## Similarity as distance

We can think of two elements that are quite similar as being *close* or *near* to one another. In other words, the *distance* between them is small.

Two elements that are not all similar are *far* or large *distances* apart. What is a distance?

**Definition:** A distance measure on a space is a function  $d(x, y)$  that takes two points in the space as arguments and produces a real number. It must satisfy the following axioms:

1. No negative distances.

$$d(x, y) \geq 0$$

2. Distances are only zero from a point to itself.

$$d(x, x) = 0$$

3. Distance is symmetric.

$$d(x, y) = d(y, x)$$

4. Distances satisfy the **triangle inequality**.



## Similarity as distance

We can think of two elements that are quite similar as being *close* or *near* to one another. In other words, the *distance* between them is small.

Two elements that are not all similar are *far* or large *distances* apart. What is a distance?

**Definition:** A *distance measure* on a space is a function  $d(x, y)$  that takes two points in the space as arguments and produces a real number. It must satisfy the following axioms:

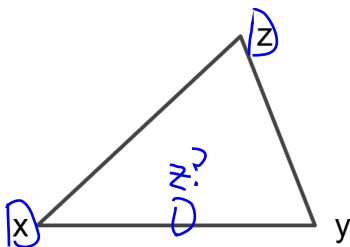
1. No negative distances.  $d(x, y) \geq 0$
2. Distances are only zero from a point to itself.  $d(x, y) = 0$  if and only if  $x = y$
3. Distance is symmetric.  $d(x, y) = d(y, x)$
4. Distances satisfy the **triangle inequality**.  $d(x, y) \leq d(x, z) + d(z, y)$

## Triangle Inequality

This is a common condition throughout many mathematical realms.

**Intuition:** Consider Alice, who walks straight from  $x$  to  $y$ . Also consider Bob, who walks from  $x$  to  $y$ , but makes a pit stop at  $z$  along the way.

The triangle inequality states that Bob's path with *any* such pit stop can never be shorter than Alice's direct path.



$$x = (1, 1)$$

$$z = (4, 3)$$

$$d(x, z) = \sqrt{(4-1)^2 + (3-1)^2}$$

## Euclidean Distance

recall:  $d(3, 7) = |7 - 3| = |3 - 7|$

You've seen Euclidean distance before! Suppose vectors  $x$  and  $y$  are in an  $n$ -dimensional Euclidean space ("normal" space, like  $\mathbb{R}$ )

**Generally:** we denote  $x = [x_1, x_2, \dots, x_n]$  and  $y = [y_1, y_2, \dots, y_n]$ .

$\mathbb{R}$

**Definition:** The most common distance is the  $L_2$ -norm, defined as:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} = \left( \sum_{i=1}^n (x_i - y_i)^2 \right)^{1/2}$$

**Example:** Suppose  $x = [1, 3, 0]$  and  $y = [2, -5, 10]$  as members of  $\mathbb{R}^3$ .

## Euclidean Distance Example

**Example:** Suppose  $x = [1, 3, 0]$  and  $y = [2, -5, 10]$  as members of  $\mathbb{R}^3$ .

Find the the  $L_2$ -norm distance between  $x$  and  $y$ .

$$\sqrt{(1-2)^2 + (3-(-5))^2 + (0-10)^2} = \sqrt{1^2 + 8^2 + 10^2}$$

## Euclidean Distance Example

**Example:** Suppose  $x = [1, 3, 0]$  and  $y = [2, -5, 10]$  as members of  $\mathbb{R}^k$ .

Find the the  $L_2$ -norm distance between  $x$  and  $y$ .

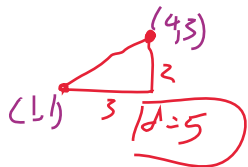
$$\sqrt{(1-2)^2 + (3-(-5))^2 + (0-10)^2} = \sqrt{165}$$

the  $L_1$ -distance:

$$(|1-2| + |3-(-5)| + |0-10|)^{1/1}$$

## Other Norms

It turns out some other distances can at times be quite handy.



**Definition:** For  $r \geq 1$ , the  $L_r$ -**norm** is defined as:

$$d(x, y) = \left( \sum_{i=1}^n (x_i - y_i)^r \right)^{1/r}$$

You may have seen one of the other common and special cases.

**Definition:** The  $L_1$ -norm is also called **Manhattan Distance**, given by:

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

element-wise  
difference

You may have seen this when computing distances on a rectangular grid (or how far apart *indices* of e.g. a matrix/array might be). It can be thought of as the sum of the distance between each index in the array.

## Other Norms

→  $L_1$  fails triangle inequality

It turns out  $r = 1$  is as low as the  $L_r$  norms make sense. But we can let  $r$  grow all the way until we ask about what  $r \rightarrow \infty$  looks like.

$$d(x, y) = \left( \sum_{i=1}^n (x_i - y_i)^r \right)^{1/r}$$

1. Result: the component  $i$  for which  $|x_i - y_i|$  is *largest* will dominate the sum.
2. So the sum looks "more and more" like  $(x_i - y_i)^r$  for that maximal  $i$  component
3. Then, when we take the  $1/r$  power or  $r$ th root, we just get back  $(x_i - y_i)$  for that maximal component

diff

$$\frac{1}{2^{100}} + \frac{3}{8^{100}} + \frac{0}{10^{100}}$$

looks like  $10^{100}$

$$(10^{100})^{1/100} \approx 10$$

## Other Norms

It turns out  $r = 1$  is as low as the  $L_r$  norms make sense. But we can let  $r$  grow all the way until we ask about what  $r \rightarrow \infty$  looks like.

$$d(x, y) = \left( \sum_{i=1}^n (x_i - y_i)^r \right)^{1/r}$$

1. Result: the component  $i$  for which  $|x_i - y_i|$  is *largest* will dominate the sum.
2. So the sum looks “more and more” like  $(x_i - y_i)^r$  for that maximal  $i$  component
3. Then, when we take the  $1/r$  power or  $r$ th root, we just get back  $(x_i - y_i)$  for that maximal component

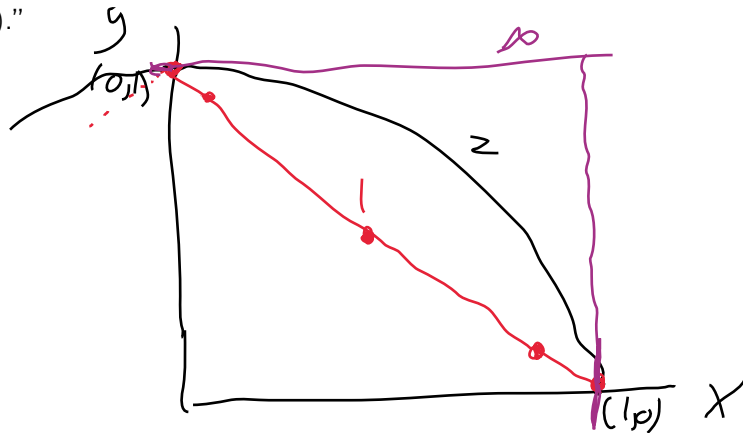
**Definition:** The  $L_\infty$ -norm or “max” norm is given by:

$$d(x, y) = \max_i |x_i - y_i|$$



## Other Norms

We can visualize these by inspecting the *unit circle*. Classically, we define the unit circle in  $\mathbb{R}^2$  by  $x^2 + y^2 = 1$ , but we can also think of it as “the set of all points of distance exactly 1 from  $(0,0)$ .”



~~—~~  $r=2$

~~—~~  $r=1$   
 $|x-0| + |y-0| = 1$   
 $|x| + |y| = 1$

~~—~~  $r=\infty$   
 $\max(|x-0|, |y-0|) = 1$

## Euclidean Distances

**Example:** Suppose you have created a video streaming service and, as your service is still in the early stages, there are only 5 movies that you provide.

You want to evaluate how similar two users are, based on the ratings they have given to these 5 movies (0 = didn't watch, 1 = worst, 5 = best). Use the  $L_1$ ,  $L_2$  and  $L_\infty$ -norms to evaluate the distances between the users.

col = people

	Zach	Tony
Cliffhanger	5	4
Emoji Movie	3	1
Rocky IV	3	5
Sharknado	2	3
High School High	2	5

$(Z:-T_i)$   
↓  
 $L_1 : d(Z, T) =$

$L_2 : d(Z, T) =$

row = products

$L_\infty : d(Z, T) =$

## Euclidean Distances

**Example:** Suppose you have created a video streaming service and, as your service is still in the early stages, there are only 5 movies that you provide.

You want to evaluate how similar two users are, based on the ratings they have given to these 5 movies (0 = didn't watch, 1 = worst, 5 = best). Use the  $L_1$ ,  $L_2$  and  $L_\infty$ -norms to evaluate the distances between the users.

$$L_1 : d(Z, T) = 1 + 2 + 2 + 1 + 3 = 9$$

	Zach	Tony
Cliffhanger	5	4
Emoji Movie	3	1
Rocky IV	3	5
Sharknado	2	3
High School High	2	5

$(Z_i - T_i)$

$$L_2 : d(Z, T) = \sqrt{1^2 + 2^2 + 2^2 + 1^2 + 3^2} = \sqrt{19}$$

$$L_\infty : d(Z, T) = 3$$

## Euclidean Distances

**Example:** Suppose you have created a video streaming service and, as your service is still in the early stages, there are only 5 movies that you provide.

You still want to evaluate how similar two users are, but what if we didn't all watch everything yet?

**Consider instead:** What could we do if there are movies that one person has seen but not the other?

	Zach	Tony
Cliffhanger	5	4
Emoji Movie	0	0
Rocky IV	3	5
Sharknado	0	3
High School High	2	5

## Setting Up Other Distances

The Euclidean distances are good for numerical data, but not great for non-numerical data like a binary classification of whether or not we *have seen* a movie may want something else.

**Example:** say we want to evaluate how similar Zach and Tony are based on which movies each of them have seen, and not their actual ratings, at this point.

We can express Zach's and Tony's movies as **sets** Z and T, respectively.

$$Z = \{C, R, HSH\}$$

$$T = \{C, R, S, HSH\}$$

	Zach	Tony
Cliffhanger	5	4
Emoji Movie	0	0
Rocky IV	3	5
Sharknado	0	3
High School High	2	5

## Setting Up Other Distances

The Euclidean distances are good for numerical data, but not great for non-numerical data like a binary classification of whether or not we *have seen* a movie may want something else.

**Example:** say we want to evaluate how similar Zach and Tony are based on which movies each of them have seen, and not their actual ratings, at this point.

We can express Zach's and Tony's movies as **sets** Z and T, respectively.

$Z = \{\text{Cliffhanger, Rocky IV, High School High}\}$

$T = \{\text{Cliffhanger, Rocky IV, Sharknado, High School High}\}$

	Zach	Tony
Cliffhanger	5	4
Emoji Movie	0	0
Rocky IV	3	5
Sharknado	0	3
High School High	2	5

## Jaccard Similarity

**Goal:** quantify the similarity of two sets.

Assume there is some **universal set** or domain  $U$ . Here, it is the set of all the movies in your rinky-dink movie platform.

$$U = \{C, E, S, A, HSH\}$$

$$Z = \{C, A, HSH\}$$

$$T = \{C, A, HSH, S\}$$

**Definition:** The *Jaccard Similarity* of sets  $S$  and  $T$  is

$$\frac{|S \cap T|}{|S \cup T|} \quad \text{"AND" / "OR"}$$

## Jaccard Similarity

**Goal:** quantify the similarity of two sets.

Assume there is some **universal set** or domain  $U$ . Here, it is the set of all the movies in your rinky-dink movie platform.

$U = \{\text{Cliffhanger, Emoji Movie, Rocky IV, Sharknado, High School High}\}$

$Z = \{\text{Cliffhanger, Rocky IV, High School High}\}$

$T = \{\text{Cliffhanger, Rocky IV, Sharknado, High School High}\}$

**Definition:** The *Jaccard Similarity* of sets  $S$  and  $T$  is



## Jaccard Similarity

**Goal:** quantify the similarity of two sets.

Assume there is some **universal set** or domain  $U$ . Here, it is the set of all the movies in your rinky-dink movie platform.

$U = \{\text{Cliffhanger, Emoji Movie, Rocky IV, Sharknado, High School High}\}$

$Z = \{\text{Cliffhanger, Rocky IV, High School High}\}$

$T = \{\text{Cliffhanger, Rocky IV, Sharknado, High School High}\}$

**Definition:** The *Jaccard Similarity* of sets  $S$  and  $T$  is

$$\text{sim}(S, T) = \frac{|S \cap T|}{|S \cup T|}$$

## Jaccard Similarity

**Definition:** The *Jaccard Similarity* of sets  $S$  and  $T$  is

$$\text{sim}(S, T) = \frac{|S \cap T|}{|S \cup T|}$$

**Example:** For our example, find the similarity of  $Z$  and  $T$ .

$Z = \{\text{Cliffhanger, Rocky IV, High School High}\}$   $T = \{\text{Cliffhanger, Rocky IV, Sharknado, High School High}\}$

## Jaccard Similarity

**Definition:** The *Jaccard Similarity* of sets  $S$  and  $T$  is

$$\text{sim}(S, T) = \frac{|S \cap T|}{|S \cup T|}$$

**Example:** For our example, find the similarity of  $Z$  and  $T$ .

$Z = \{\text{Cliffhanger, Rocky IV, High School High}\}$   $T = \{\text{Cliffhanger, Rocky IV, Sharknado, High School High}\}$

**Solution:**

$$\text{sim}(Z, T) = \frac{|Z \cap T|}{|Z \cup T|} = \frac{|\{\text{Cliffhanger, Rocky IV, High School High}\}|}{|\{\text{Cliffhanger, Rocky IV, Sharknado, High School High}\}|} = \frac{3}{4}$$

## Jaccard distance

Is Jaccard similarity a distance measure?

## Jaccard distance

Is Jaccard similarity a distance measure?

No way!. It doesn't even pass an intuition test.

1. If two sets  $S$  and  $T$  are similar, then  $d(S, T)$  should be **small**.
2. ...but  $\text{sim}(S, T)$  is **large** (near 1) for similar sets and **small** (approaches zero) for non-overlapping sets.
3. In fact... this sounds like the *opposite of a distance*

## Jaccard distance

Is Jaccard similarity a distance measure?

**Definition:** The **Jaccard distance** of sets  $S$  and  $T$  is

$$d(S, T) = 1 - \text{sim}(S, T)$$

## Matrix Representation of Sets

We can retain the numerical ideas from Euclidean distances with set similarities.

For our sets:

$Z = \{\text{Cliffhanger, Rocky IV, High School High}\}$

$T = \{\text{Cliffhanger, Rocky IV, Sharknado, High School High}\}$

**Definition:** the characteristic matrix of sets  $Z$  and  $T$  is the matrix whose columns correspond to the sets  $Z$  and  $T$  and whose rows correspond to all elements of the universal set  $U$ .

There is a 1 in row  $r$  and column  $c$  if the element for  $r$  is a member of the set in column  $c$ . Else, there is a 0 at position  $(r, c)$ .

	Zach	Tony
Cliffhanger	1	1
Emoji Movie	0	0
Rocky IV	1	1
Sharknado	0	1
High School High	1	1

## Matrix Representation of Sets

- ▶ In reality, such a table might be **massive**. We tend to both have lots of users (columns) and lots of choices (rows).
- ▶ We almost **never** want to actually store the entire characteristic matrix. It is usually *sparse*: there are **far more zeros than nonzeros**. So we often just store the locations of the 1s.
- ▶ Consider storing smaller such characteristic matrices as collections of (vector) bitstrings:  
 $Z = \{1, 0, 1, 0, 1\}$ ,  $T = \{1, 0, 1, 1, 1\}$ . This may give us a sense of how to extend other notions of distance.

$$10^6 \text{ rows} \\ 10^6 \text{ col} \\ = 10^{12} = \text{ trillion}$$

	Zach	Tony
Cliffhanger	1	1
Emoji Movie	0	0
Rocky IV	1	1
Sharknado	0	1
High School High	1	1



## Other non-Euclidean distances

**Definition:** the *edit distance* (LCS) between two strings  $x = x_1x_2x_3 \dots x_n$  and  $y = y_1y_2y_3 \dots y_n$  is the smallest number of *insertions and deletions* of single characters that will convert  $x$  to  $y$ .

**Definition:** the *Hamming distance* between two vectors or strings is the number of components in which they differ. It is also known as the *substitution distance*.

NB: we're not including substitutions in “edit distance” here, but many sources do. A variant of edit distance that explicitly allows substitutions, insertions, or deletions is called *Levenshtein distance*.

**Example:** With  $Z = 10101$  and  $T = 10111$ , what are the edit and Hamming distances between them?

**Example:** What about the edit distance for  $x = abcde$  and  $y = acfdeg$ ?

## Other non-Euclidean distances

**Example:** With  $Z = 10101$  and  $T = 10111$ , what are the edit and Hamming distances between them?

**Example:** What about the edit distance for  $x = abcde$  and  $y = acfdeg$ ?

## Other non-Euclidean distances

**Example:** With  $Z = 10101$  and  $T = 10111$ , what are the edit and Hamming distances between them?

**Example:** What about the edit distance for  $x = abcde$  and  $y = acfdeg$ ?

**Solution:** For  $Z$  and  $T$ , the hamming distance is 1 since they differ only in the 4th component. This is an edit distance of 2, however, as it requires deleting and re-inserting a new 4th term to convert one to the other.

For  $x$  and  $y$ , we have an edit distance of 3: from  $x$  we delete the  $b$ , insert  $f$  between  $c$  and  $d$ , and finally insert  $g$  after  $e$ .

## Other non-Euclidean distances

**Example:** prove that the edit distance is a proper distance measure.

## Other non-Euclidean distances

**Example:** prove that the edit distance is a proper distance measure.

**Solution:**

1. No negative distances.
2. Distances are only zero from a point to itself.
3. Distance is symmetric.
4. Distances satisfy the **triangle inequality**.

## Other non-Euclidean distances

**Example:** prove that the edit distance is a proper distance measure.

**Solution:**

1.  $d(x, y) \geq 0$ : ☒. We're counting operations: obviously can't be negative!
2.  $d(x, y) = 0 \leftrightarrow x = y$ : ☒. Two statements: if  $x = y$ , clearly the edit distance is zero. On the other hand, if we don't edit a string  $x$  then we get the same string, so no edit distance also implies  $x = y$ .
3.  $d(x, y) = d(y, x)$  ☒ Given a sequence of edits to turn  $x$  to  $y$ , we can turn this into a same-length sequence of edits to get  $y$  to  $x$  by changing each insertion to a deletion and vice versa. (in other words: each operation is invertible!)
4.  $d(x, y) \leq d(x, z) + d(z, y)$  ☒. If we turn  $x$  to  $y$  by first turning  $x$  to  $z$ , then it the number of edits can't be less than going directly to  $y$ .

So edit distance is a proper distance measure.

## Better Proofs

To be quite honest, the argument on the prior slide for the **triangle inequality** was a little hand-wavy. What's a more formal proof for  $d(x, y) \leq d(x, z) + d(z, y)$ ?

### Proof by Contradiction (for a $p \rightarrow q$ proof)

1. Assume  $p$  and  $\neg q$ : "Suppose edit distance is distance measure ( $p$ ) that so that  $d(x, y) > d(x, z) + d(z, y)$ . (*not*  $q$ )"
2. By definition of edit distance, the *minimum* edits to get from string  $x$  to string  $z$  is  $d(x, z)$ , and the *minimum* edits to get from string  $z$  to string  $y$  is  $d(z, y)$ .
3. We can get from string  $x$  to string  $y$  by changing  $x$  to  $z$  and then  $z$  to  $y$ , so **one** way to get from  $x$  to  $y$  is via  $z$ , at a total distance of  $d(x, z) + d(z, y)$ .
4. Because edit distance is the *minimum* edits necessary, the actual distance  $d(x, y)$  can not be more than the *possible* distance  $d(x, z) + d(z, y)$ . In other words,  $d(x, y) \leq d(x, z) + d(z, y)$  which contradicts our assumptions of  $p$  and  $\neg q$ , so we must conclude that  $p \rightarrow q$  (or if the metric is edit distance, it satisfies the triangle inequality.)

## Matrix Representation of Sets: Size concerns

- ▶ In reality, such a table might be **massive**. We tend to both have lots of users (columns) and lots of choices (rows).
- ▶ Lots of rows: imagine each row represents some combination of words or characters in a document
- ▶ Lots of columns: each column could represent a document, e.g. an e-mail.
- ▶ We will want Jaccard similarity, but looping over all possible pairs of documents and over all rows would take a very long time...

	Zach	Tony
Cliffhanger	1	1
Emoji Movie	0	0
Rocky IV	1	1
Sharknado	0	1
High School High	1	1

**Recall:** How many comparisons is comparing each pair of  $n$  users? What if each comparison has  $m$  movies in the list?

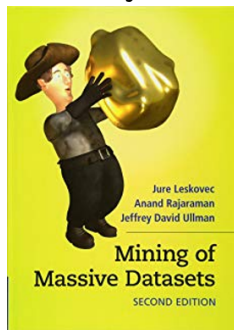
$$\binom{n}{2} = \frac{n(n-1)}{2} \cdot m$$

**Next time:** minhashing and document similarity.



## Acknowledgments

Some material is adapted/adopted from Mining of Massive Data Sets, by Jure Leskovec, Anand Rajaraman, Jeff Ullman (Stanford University) <http://www.mmds.org>



Special thanks to Tony Wong for sharing his original adaptation and adoption of slide material.