COSC 3337 : Data Science I

N. Rizk

College of Natural and Applied Sciences

Department of Computer Science

University of Houston

What is similarity?

Similarity is hard to define, but…

"*We know it when we see it*"

Detecting similarity is a typical task
in machine learning

Definition of similarity

The definition of **similarity or dissimilarity between objects** depends on

- the type of the data considered
- what kind of similarity we are looking for

Similarity and Dissimilarity

- **Similarity**
  - Numerical measure of how alike two data objects are
  - Value is higher when objects are more alike
  - Often falls in the range [0,1]
- **Dissimilarity** (e.g., distance)
  - Numerical measure of how different two data objects are
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- **Proximity** refers to a similarity or dissimilarity

Common Properties of a Similarity

- Similarities, also have some well-known properties.

1. $s(\mathbf{x}, \mathbf{y}) = 1$ (or maximum similarity) only if $\mathbf{x} = \mathbf{y}$.

2. $s(\mathbf{x}, \mathbf{y}) = s(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}$ and $\mathbf{y}$. (Symmetry)

where $s(\mathbf{x}, \mathbf{y})$ is the similarity between points (data objects), $\mathbf{x}$ and $\mathbf{y}$.

Data Matrix and Dissimilarity Matrix

- Data matrix
  - n data points with p dimensions
  - Two modes

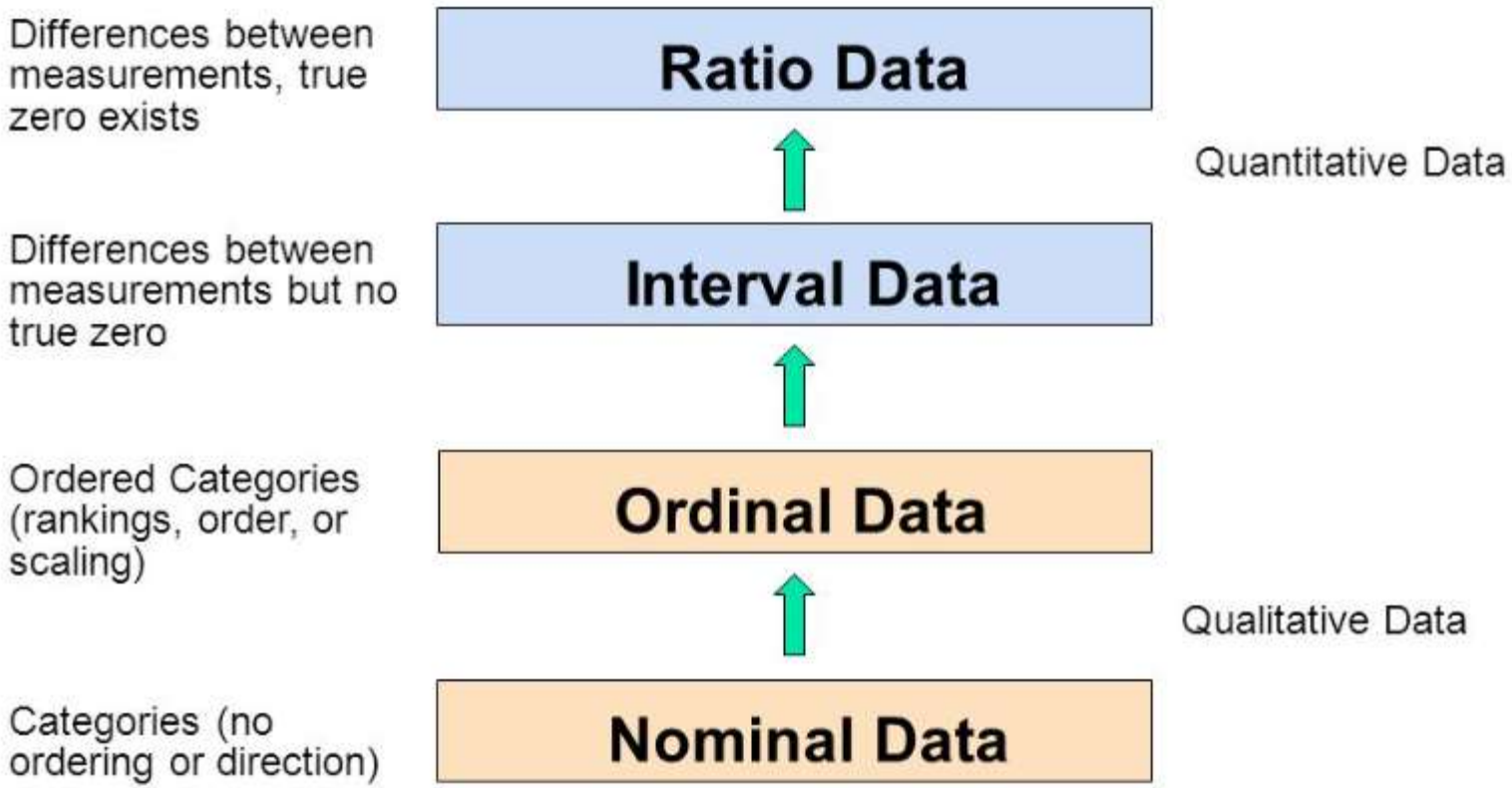
$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

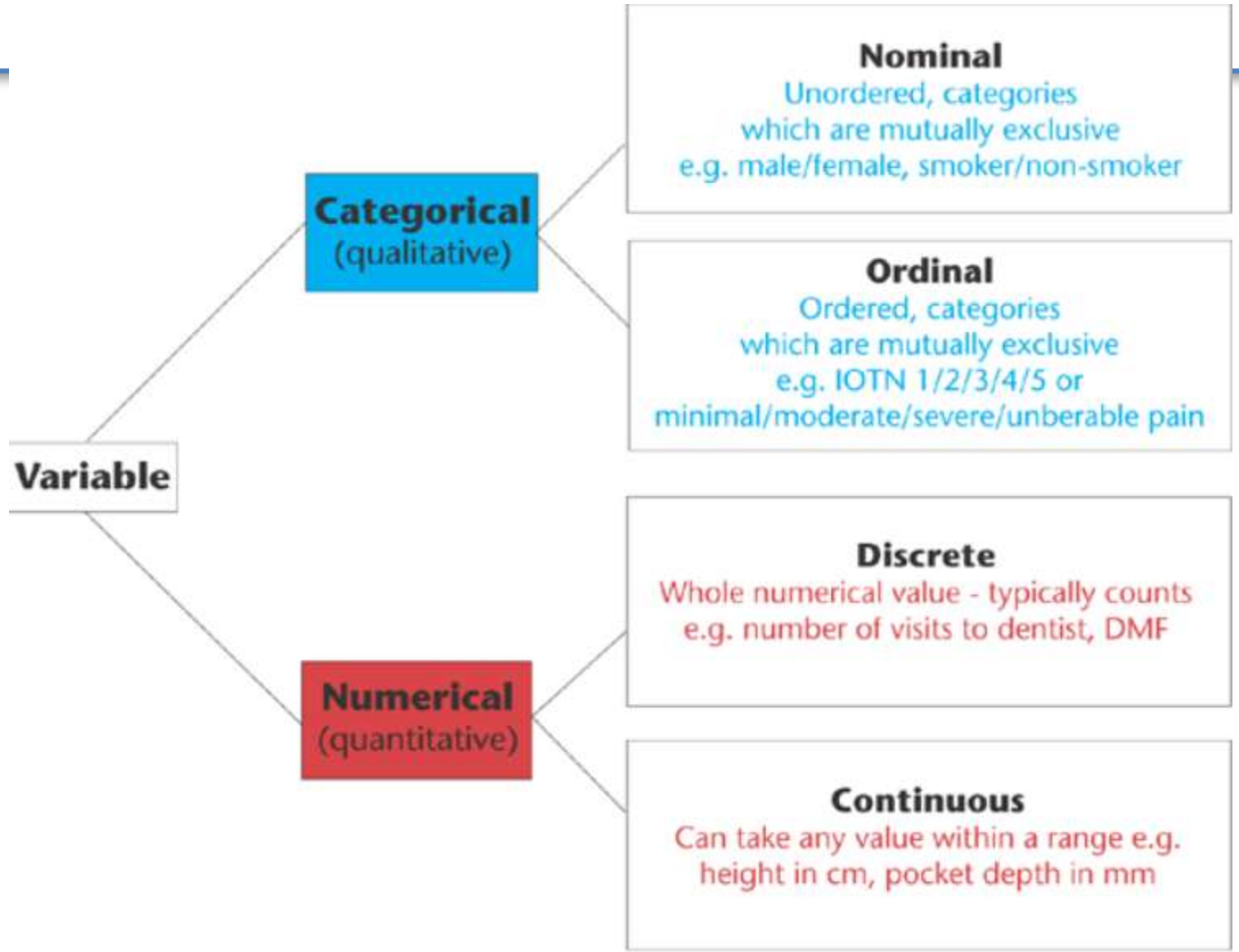
- Dissimilarity matrix
  - n data points, but registers only the distance
  - A triangular matrix
  - Single mode

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

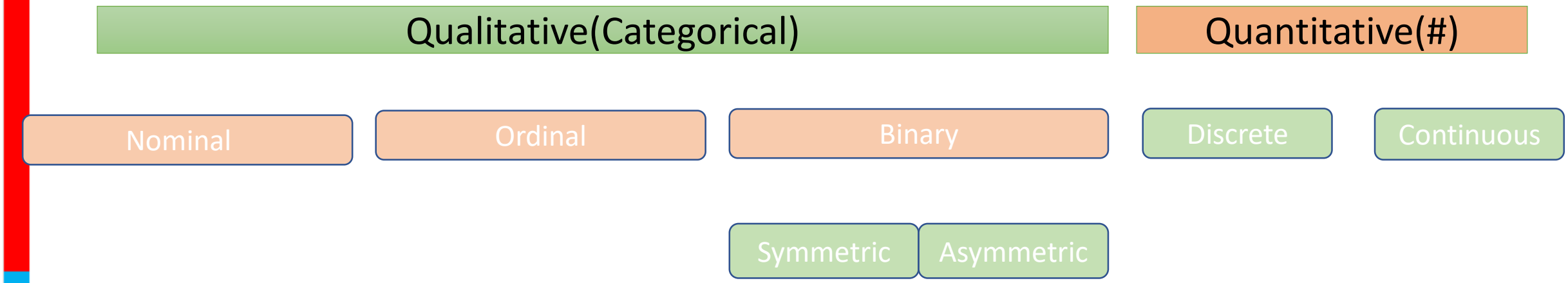
Measuring distance based onType of data

1. **Interval-scaled variables**
2. **Binary variables**
3. **Nominal, ordinal, and ratio variables**
4. **Variables of mixed types**


Differences between measurements, true zero exists
Ratio Data
Differences between measurements but no true zero
Interval Data
Ordered Categories (rankings, order, or scaling)
Ordinal Data
Categories (no ordering or direction)
Nominal Data
Quantitative Data
Qualitative Data


Variable
Categorical
(qualitative)
Nominal
Unordered, categories
which are mutually exclusive
e.g. male/female, smoker/non-smoker
Ordinal
Ordered, categories
which are mutually exclusive
e.g. IOTN 1/2/3/4/5 or
minimal/moderate/severe/unberable pain
Numerical
(quantitative)
Discrete
Whole numerical value - typically counts
e.g. number of visits to dentist, DMF
Continuous
Can take any value within a range e.g.
height in cm, pocket depth in mm

Attributes Types



1-Interval-valued variables (quantitative, numeric)

Measured on a scale of equal-sized units

◦ Values have order ◦ E.g., temperature in C˚or F˚, calendar dates ◦ No true zero-point.

Calculate the mean absolute deviation

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|)$$

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \ldots + x_{nf}).$$

Standardize data:

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

Calculate the standardized measurement (*z-score*)

Using mean absolute deviation is more robust to handle outliers than using standard deviation

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2 = \frac{1}{N}\sum_{i=1}^{n} x_i^2 - \mu^2$$

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n-1}[\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2]$$

Interval-valued variables

	Age	income
Employee1	18	50000
Employee2	17	10000
Employee3	16	100000
Employee4	20	100000

Calculate the **mean absolute deviation**

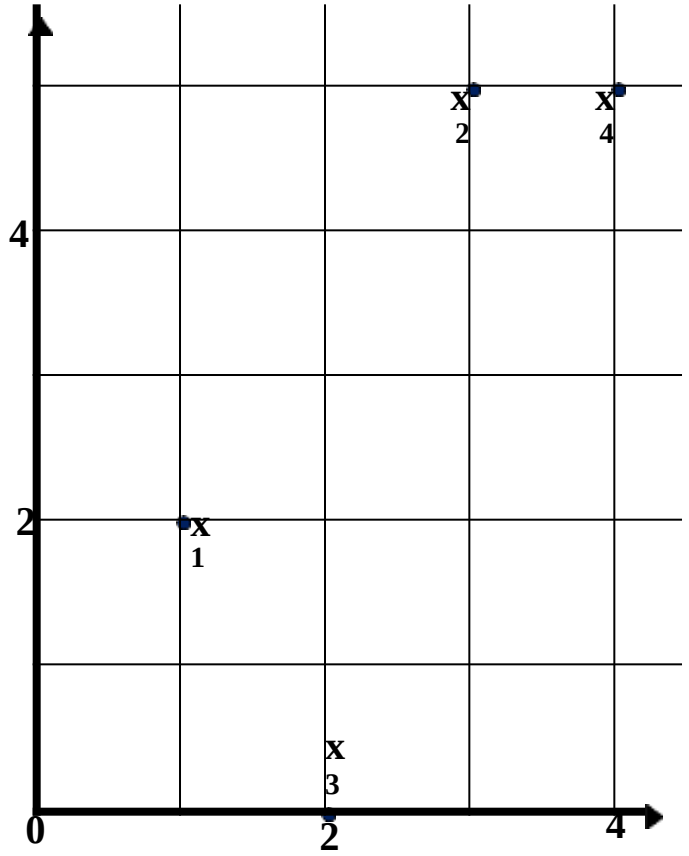
$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + ... + |x_{nf} - m_f|)$$

Calculate the **standardized measurement** (***z-score***

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

	Age	income	z-age	z-income
Employee1	18	50000	0.2	-0.42857143
Employee2	17	10000	-0.6	-1.57142857
Employee3	16	100000	-1.4	1
Employee4	20	100000	1.8	1
m	17.75	65000		
s(age)	1.25	35000		

Data Matrix and Dissimilarity Matrix



Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Standardization is necessary, if scales differ.

Data Matrix

point	attribute1	attribute2
x1	1	2
x2	3	5
x3	2	0
x4	4	5

Dissimilarity Matrix

(with Euclidean Distance)

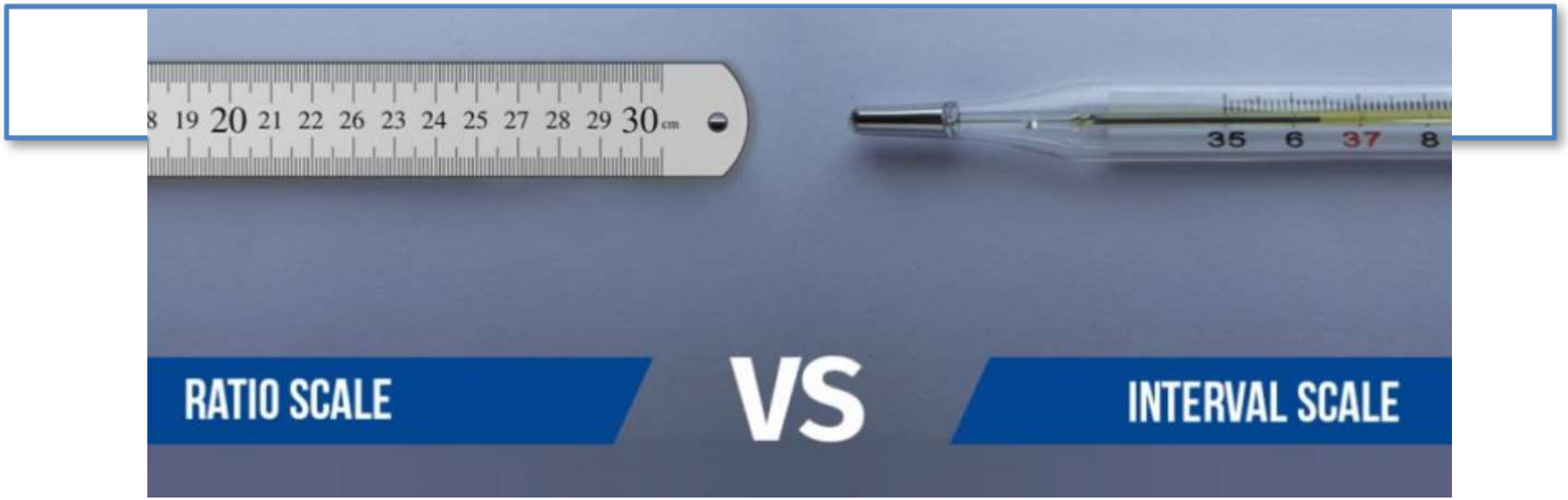
	x1	*x2*	*x3*	*x4*
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Interval-valued variables: Dissimilarity Matrix

	Age	income	z-age	z-income
Employee1	18	50000	0.2	-0.42857143
Employee2	17	10000	-0.6	-1.57142857
Employee3	16	100000	-1.4	1
Employee4	20	100000	1.8	1
m	17.75	65000		
s(age)	1.25	35000		

Dissimilaity	Employee1	Employee2	Employee3	Employee4
Employee1	0.00			
Employee2	1.40	0.00		
Employee3	2.14	2.69	0.00	
Employee4	2.14	3.52	3.20	0.00

Standardizing measurements attempts to give all variables an equal weight.



Ratio Scale is defined as a variable measurement scale that not only produces the order of variables but also makes the difference between variables known along with information on the value of true zero.

Ratio scale variables

- What is your daughter's current height?
  - Less than 5 feet.
  - 5 feet 1 inch – 5 feet 5 inches
  - 5 feet 6 inches- 6 feet
  - More than 6 feet
- What is your weight in kilograms?
  - Less than 50 kilograms
  - 51- 70 kilograms
  - 71- 90 kilograms
  - 91-110 kilograms
  - More than 110 kilograms

- Ratio-scaled variable: a positive measurement on a nonlinear scale, approximately at exponential scale, such as Ae^{Bt} or Ae^{-Bt}

How do we transform the nonlinear quantitative variables to create a linear relationship

– Apply **logarithmic transformation**

$$y_{if} = log(x_{if})$$

– Treat them as continuous ordinal data

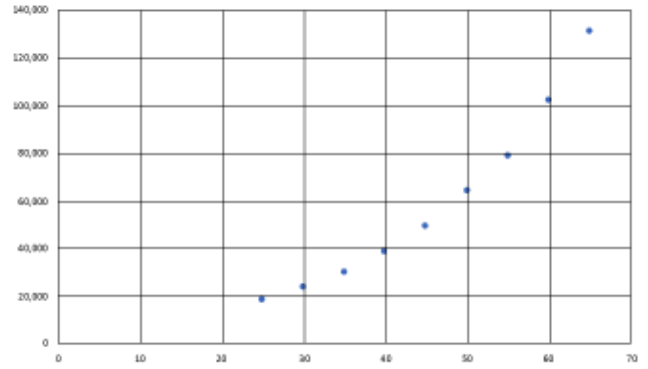
– treat their rank as interval-scaled.

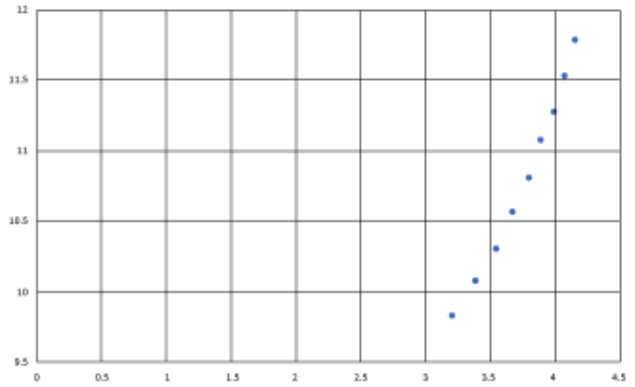
Dataset	Original Data	Transformed Data

Age X	Income Y
25	18,500
30	23,600
35	29,800
40	38,500
45	49,000
50	64,100
55	78,500
60	102,000
65	130,800

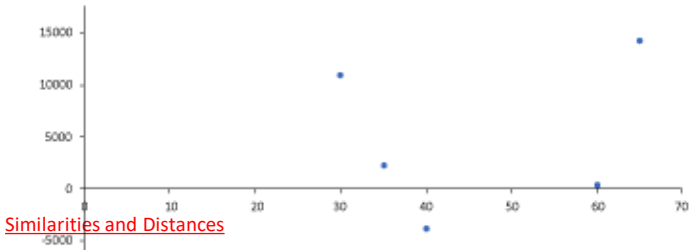
Age X	Income ln(Y)
25	ln(18,500)=9.83
30	ln(23,600)=10.07
35	ln(29,800)=10.30
40	ln(38,500)=10.56
45	ln(49,000)=10.80
50	ln(64,100)=11.07
55	ln(78,500)=11.27
60	ln(102,000)=11.53
65	ln(130,800)=11.78

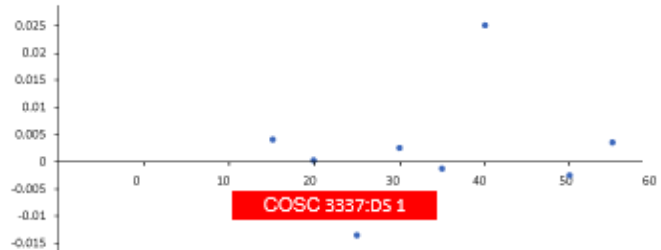
Scatterplot





Residual Plot





Similarity/Dissimilarity for Simple Attributes

p and *q* are the attribute values for two data objects.

Category/qualitative

Ranking/qualitative

Numeric/Quantitative
Discrete vs continuous

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p-q \rvert}{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{\lvert p-q \rvert}{n-1}$
Interval or Ratio	$d = \lvert p - q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Binary attributes are a special case of discrete attributes

Continuous attributes are typically represented as floating-point variables

Proximity Measure

Similarity

- Symmetric
- Asymmetric

Dissimilarity

- Symmetric
- Asymmetric

Binary

Simple Matching Coefficient(m11+m00/all)

Jaccard (m11/ all but not m00)

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

$$d(i,j) = \frac{b+c}{a+b+c}$$

Symmetric: Both values are equally important (Gender).

2-Proximity Measure for Binary Attributes

- **A contingency table for binary data**

Object *i* \ Object *j*	1	0	*sum*
1	a	b	$a+b$
0	c	d	$c+d$
sum	$a+c$	$b+d$	p

- **Distance measure for symmetric binary variables:**

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

- **Distance measure for asymmetric binary variables:**

$$d(i,j) = \frac{b+c}{a+b+c}$$

- **Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):**

$$sim_{Jaccard}(i,j) = \frac{a}{a+b+c}$$

https://www.sciencedirect.com/topics/computer-science/binary-attribute

Example of Binary Attributes =SMC & Jaccard

Object 1:	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Object 2:	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**

Calculate the similarity of symmetric and asymmetric

1- Contingency table always 1 0

2- SMC

3-J

J or Simple Matching Coefficient?

Three claims A, B & C with 20 binary attributes, (symmetric , asymmetric???)
Claim A = (R,R,R,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G)
Claim B = (R,R,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G)
Claim C = (R,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G,G)

Find the distances between A,B; A,C;B,C.

```
sklearn.metrics.jaccard_similarity_score(y_true, y_pred, normalize=True,
sample_weight=None)
```

If normalize == True, return the average Jaccard similarity coefficient, else it returns the sum of the Jaccard similarity coefficient over the sample set.

The best performance is 1 with normalize == True and the number of samples with normalize == False.

Dissimilarity between Binary Variables

- Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- Gender is a symmetric attribute
- The remaining attributes are asymmetric binary
- Let the values Y and P be 1, and the value N be 0

$$d(jack, mary) = \frac{0 + 1}{2 + 0 + 1} = 0.33$$

$$d(jack, jim) = \frac{1 + 1}{1 + 1 + 1} = 0.67$$

$$d(jim, mary) = \frac{1 + 2}{1 + 1 + 2} = 0.75$$

What if the variable is

continuous/numeric

?➔Euclidean

```
# calculating euclidean distance between vectors
from math import sqrt

# calculate euclidean distance
def euclidean_distance(a, b):
	return sqrt(sum((e1-e2)**2 for e1, e2 in
zip(a,b)))

# define data
row1 = [10, 20, 15, 10, 5]
row2 = [12, 24, 18, 8, 7]
# calculate distance
dist = euclidean_distance(row1, row2)
print(dist)
```

3-Proximity Measure for Nominal Attributes

(qualitative, categorical)

- Nominal=>Can take 2 or more states, e.g., red, yellow, blue, green (generalization of a binary attribute)
- Examples: ID numbers, eye color, zip codes
- Method 1: Simple matching
  - m: # of matches, p: total # of variables

$$d(i, j) = \frac{p - m}{p}$$

- Method 2: Use large number of binary attributes
  - creating a new binary attribute for each of the M nominal states

Dissimilarity between Nominal Attributes Here, we have one nominal attribute, test-1, so p=1. m=?The dissimilarity matrix is

Student	Grade
s1	A
s2	B
s3	C
s4	A

For p=1,
d(i, j) evaluates to 0 , if objects i and j match, and
1
if the objects differ.

What is the dissimilarity matrix?

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Calculate the dissimilarity matrix of the Nominal Attributes

RollNo	Marks	Grade
1	90	A
2	80	B
3	82	B
4	90	A

$$d(i, j) = \frac{p - m}{p}$$

d(1,1) = P – M / P = 2 – 2 / 2 = 0	d(RollNo1,RollNo2)	d(RollNo1,RollNo3)	d(RollNo1,RollNo4)
(2,1) = P – M / P = (2 – 0) / 2 = 1	(2,2) = P – M / P = (2 – 2) / 2 = 0	d(RollNo2,RollNo3)	d(RollNo2,RollNo4)
(3,1) = P – M / P = (2 – 0) / 2 = 1	(3,2) = P – M / P = (2 – 1)/ 2 = 0.5	(3,3) = P – M / P = (2 – 2)/ 2 = 0	d(RollNo3,RollNo4)
(4,1) = P – M / P = (2 – 2) / 2 = 0	(4,2) = P – M / P = (2 – 0) / 2 = 1	(4,3) = P – M / P =(2 – 0)/ 2 = 1	(4,4) = P – M / P =(2 – 2) / 2 = 0

3- Ordinal Variables (qualitative, Ranking)

- A discrete ordinal variable resembles a categorical variable, except that the M states of the ordinal value are ordered in a meaningful sequence.
  Example: professional ranks are often enumerated in a sequential order, such as assistant, associate, and full for professors.

- Ordinal variables may also be obtained from the discretization of interval-scaled quantities by splitting the value range into a finite number of classes.
- The values of an ordinal variable can be mapped to ranks.
  - Example: suppose that an ordinal variable f has M_f states.
  - These ordered states define the ranking $1, \ldots, M_f$.

Ordinal Variable With Numeric Value

How satisfied are you with our service tonight?
1.Very satisfied
2.Satisfied
3.Indifferent
4.Dissatisfied
5.Very dissatisfied

Ordinal Variable Without Numeric value

How satisfied are you with our service tonight?
•Very satisfied
•Satisfied
•Indifferent
•Dissatisfied
•Very dissatisfied

Characteristics of Ordinal Variable

•It is an extension of nominal data.
•It has no standardized interval scale.
•It establishes a relative rank.
•It measures qualitative traits.
•The median and mode can be analyzed.
•It has a rank or order.

Examples of Ordinal Variable

Likert Scale: A Likert scale is a psychometric scale used by researchers to prepare questionnaires and get people's opinions.

How satisfied are you with our service?
1.Very satisfied
2.Satisfied
3.Indifferent
4.Dissatisfied
5.Very dissatisfied

Interval Scale: each response in an interval scale is an interval on its own.

How old are you?
•13-19 years
•20-30 years
•31-50 years

Step 1

- Suppose that f is a variable from a set of ordinal variables describing n objects.
- The dissimilarity computation with respect to f involves the following steps:
- Step 1:
  - The value of f for the ith object is x_{if}, and f has M_f ordered states, representing the ranking $1, \ldots, M_f$.
  - Replace each x_{if} by its corresponding rank:

$$r_{if} \in \{1, \ldots, M_f\}$$

Step 2 & 3

- Step 2:
  - Since each ordinal variable can have a different number of states, it is often necessary to map the range of each variable onto [0.0, 1.0] so that each variable has equal weight.
  - This can be achieved by replacing the rank r_{if} of the i th object in the f th variable by:

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- Step 3:
  - Dissimilarity can then be computed using any of the distance measures described for interval-scaled variables.

Example

- Suppose that we have the sample data:

Student	Exam
s1	Excellent
s2	Fair
s3	Good
s4	Excellent

- There are three states for the exam, namely fair, good, and excellent, that is $M_f = 3$.

Example: Dissimilarity between ordinal variables

- Step 1: if we replace each value for the exam by its rank, the four objects are assigned the ranks 3, 1, 2, and 3, respectively.
- Step 2: normalizes the ranking by mapping rank 1 to 0.0, rank 2 to 0.5, and rank 3 to 1.0.
- Step 3: we can use, say, the Euclidean distance, which results in the following dissimilarity matrix:

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}$$

Example 2

A1	A2	A3
La	1	K
La	2	L
Med	1	L
Sm	2	K

Ranks(A1) = {1,2,3}
Ranks(A2)= {1,2}
Ranks(A3) = {1,2}

The next step is to calculate the Z value for each attribute(normalize) using the formula:

The z-scores for A1 are

Zsm = (1–1)/(3–1) = 0
Zmed = (2–1)/(3–1) = ½
Zlg = (3–1)/(3–1) = 1

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

The z-scores for A2 are

Z1 = (1–1)/(2–1) = 0
Z2 = (2–1)/(2–1) = 1

The z-scores for A3 are

ZL = (1–1)/(2–1) = 0
ZK = (2–1)/(2–1) = 1

Then we replace each value with the new z-scores.

A1	A2	A3
1	0	1
1	1	0
0.5	0	0
0	1	1

Step 3 :calculate the dissimilarity matrix using Euclidean

	A1	A2	A3
s1	1	0	1
s2	1	1	0
s3	0.5	0	0
s4	0	1	1

Euclidean

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

	s1	s2	s3	s4
s1	0.000			
s2	1.414	0.000		
s3	1.118	1.118	0.000	
s4	1.414	1.414	1.500	0.000

4- Attributes of Mixed Type

- A database may contain all attribute types
  - Nominal, symmetric binary, asymmetric binary, numeric, ordinal
- One may use a weighted formula to combine their effects

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum^{p} \delta^{(f)}}$$

- $\delta_{ij}^{(f)} = 0$
  - if either (1) x_{if} or x_{jf} is missing (i.e., there is no measurement of variable f for object i or object j),
  - or (2) $x_{if} = x_{jf} = 0$ and variable f is asymmetric binary;
- otherwise

object identifier	*test-1 (nominal)*	*test-2 (ordinal)*	*test-3 (numeric)*
1	code-A	excellent	45
2	code-B	fair	22
3	code-C	good	64
4	code-A	excellent	28

1- Dissimilarity of Nominal
2- Dissimilarity of Ordinal
3- Dissimilarity of Numeric

object identifier	*test-1 (nominal)*	*test-2 (ordinal)*	*test-3 (numeric)*
1	code-A	excellent	45
2	code-B	fair	22
3	code-C	good	64
4	code-A	excellent	28

Min-max normalization is one of the most common ways to normalize data. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1.

Nominal	1	2	3	4
1	0			
2	1	0		
3	1	1	0	
4	0	1	1	0

Ordinal	1	2	3	4
1	0.0			
2	1.0	0.0		
3	0.5	0.5	0.0	
4	0.0	1.0	0.5	0.0

Min-max normalization	Z-score normalization
Not very well efficient in handling the outliers	Handles the outliers in a good way.
Min-max Guarantees that all the features will have the exact same scale.	Helpful in the normalization of the data but not with the *exact* same scale.

object identifier	*test-1 (nominal)*	*test-2 (ordinal)*	*test-3 (numeric)*
1	code-A	excellent	45
2	code-B	fair	22
3	code-C	good	64
4	code-A	excellent	28

d=i-j/max-min	Numeric	1	2	3	4
45	1	0.00			
22	2	0.55	0.00		
64	3	0.45	1.00	0.00	
28	4	0.40	0.14	0.86	0.00

$$d(2,1)=\frac{(1x1)+(1x1)+(1x0.55}{1+1+1}$$

All	1	2	3	4
1				
2	0.85			
3				
4				

Variables of Mixed Types

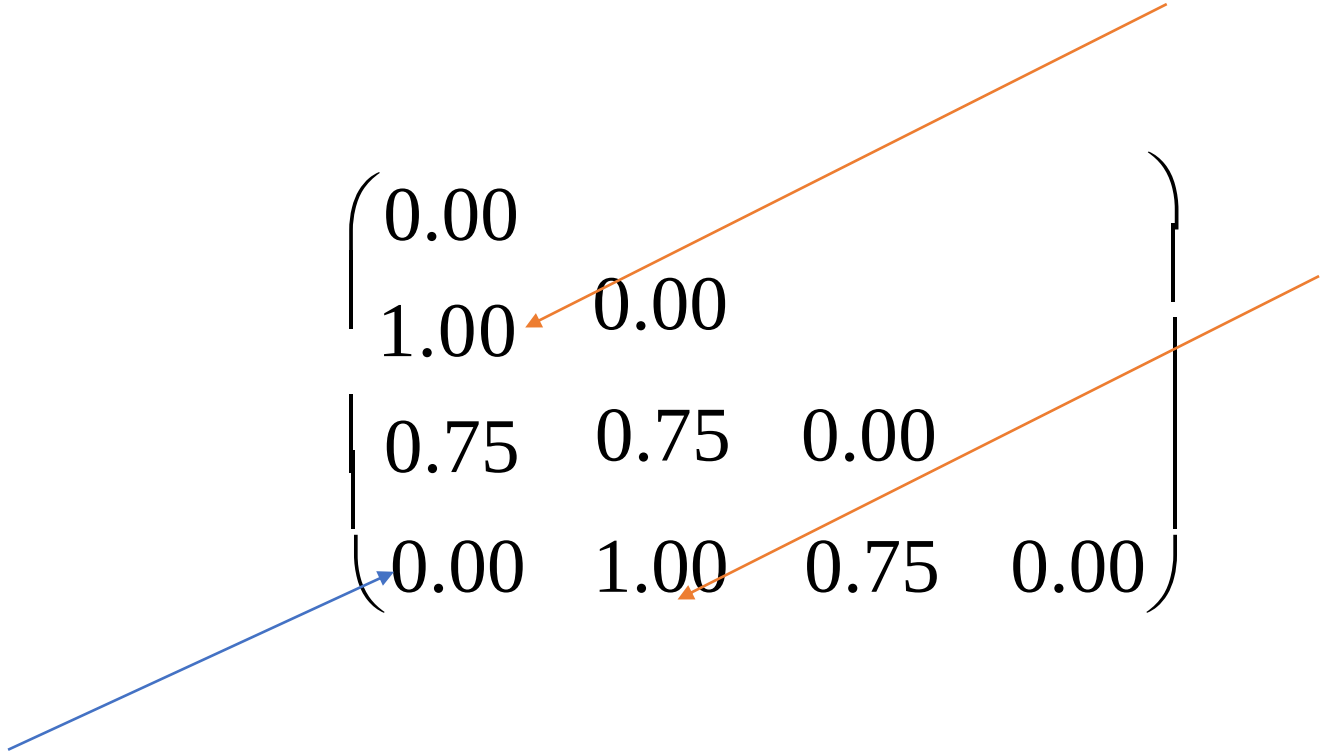
- The contribution of variable f to the dissimilarity between i and j, that is, $d_{ij}^{(f)}$
- If f is interval-based:
  - use the normalized distance so that the values map to the interval [0.0,1.0].
- If f is binary or categorical:
  - $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ otherwise
- If f is ordinal:
  - compute ranks r_{if} and

Example: Dissimilarity between variables of mixed type

Student	Grade	Exam
s1	A	Excellent
s2	B	Fair
s3	C	Good
s4	A	Excellent

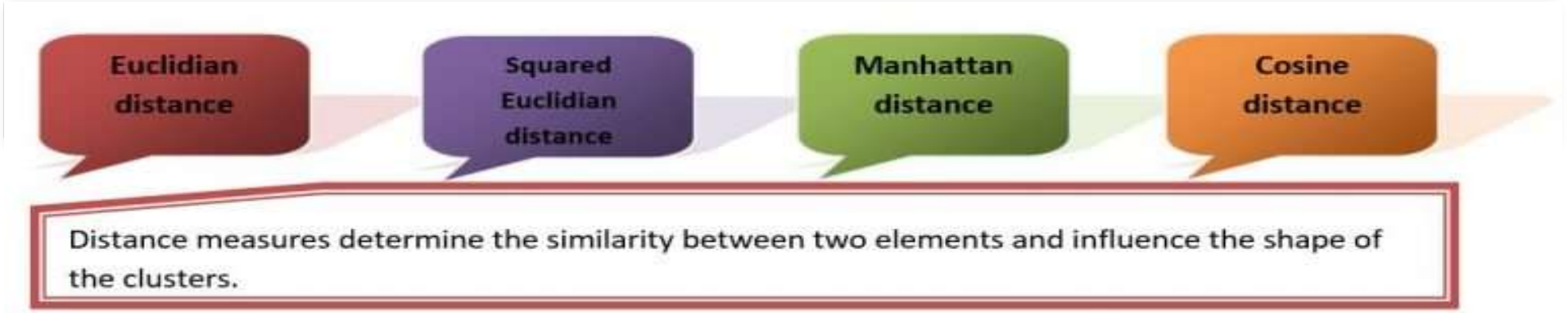
- For Grade (which is categorical) is the same as outlined before
- For the exam (which is ordinal) is the same as outlined before
- We can now calculate the dissimilarity matrices for the two variables.

Result …Verify the dissimilarity matrix



Attributes of Mixed Type - Extra

Name	Age	Income	Bac. Mark	Married	Bac. Type	School	City	English	Gender	Result
said	18	50000	180	Y	S	A	Damas	Bad	M	Y
hassan	17	10000	100	N	S	B	Alep	Good	F	N
fadi	16	100000	190	N	S	A	Daraa	Excellent	F	Y
rana	20	100000	140	Y	L	C	Damas	Excellent	F	N
	Interval			Binary (only 2 values)		Nominal		Ordinal	Binary (only 2 values)	




```
def distance(x,y):
    p = len(x)
    m = sum(map(lambda (a,b): 1 if a == b else 0, zip(x,y)))
    return float(p-m)/p
```

Example:

```
x1 = ("forestry", "plantation", "high", "low", "high", "medium", 3, "low", 297, 1, True)
x2 = ("plantation", "plantation", "high", "medium", "low", "low", 1, "low", 298, 2, True)

print distance(x1,x2) # result: 0.636363636364 = (11-4)/7
```

Common Properties of a Distance

- Distances, such as the Euclidean distance, have some well known properties.

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$ for all x and y and $d(\mathbf{x}, \mathbf{y}) = 0$ only if $\mathbf{x} = \mathbf{y}$. (Positive definiteness)
2. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}$ and $\mathbf{y}$. (Symmetry)
3. $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ for all points $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$. (Triangle Inequality)

where $d(\mathbf{x}, \mathbf{y})$ is the distance (dissimilarity) between points (data objects), $\mathbf{x}$ and $\mathbf{y}$.

- A distance that satisfies these properties is a metric

```
from math import*

def euclidean_distance(x,y):

  return sqrt(sum(pow(a-b,2) for a, b in zip(x, y)))

print euclidean_distance([0,3,4,5],[7,6,3,-1])
```

```
from math import*

def manhattan_distance(x,y):

  return sum(abs(a-b) for a,b in zip(x,y))

print manhattan_distance([10,20,10],[10,20,20])
```

Euclidean Distance

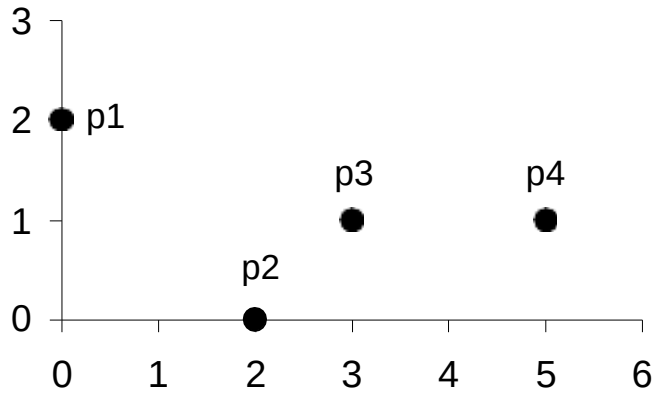
- Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

Where n is the number of dimensions (attributes) and p_k and q_k are, respectively, the k[th] attributes (components) or data objects p and q.

- Standardization is necessary, if scales differ.

Euclidean Distance



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

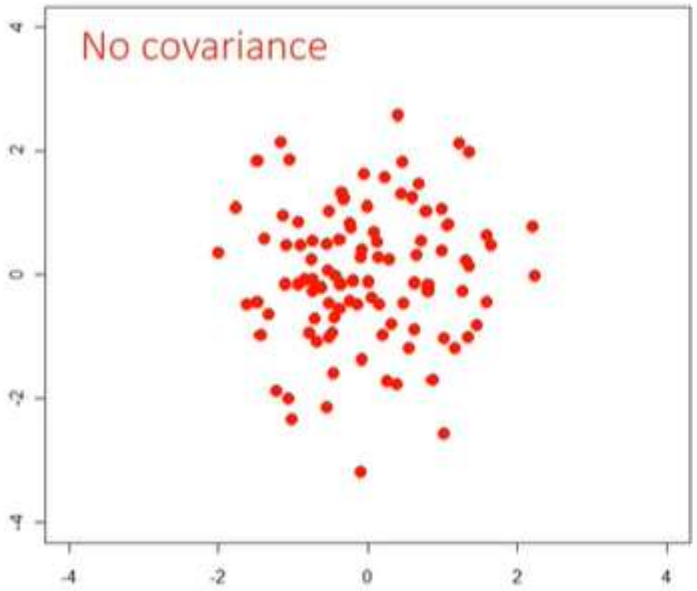
Distance Matrix

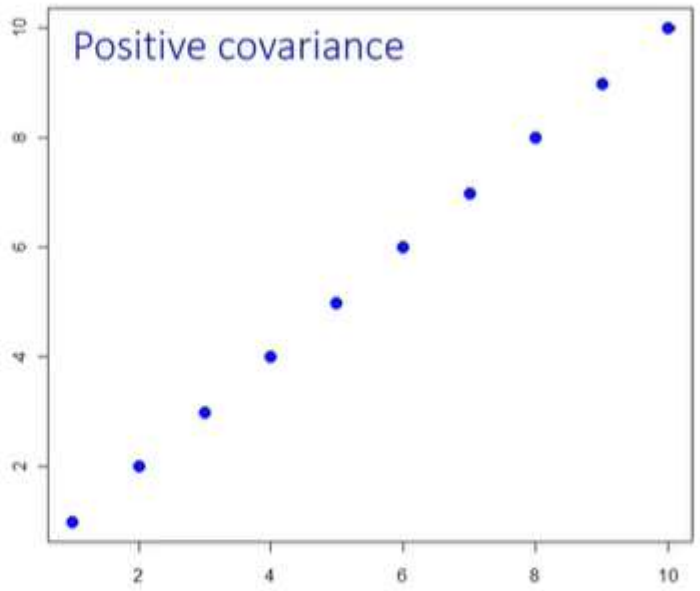
```
from math import*
def euclidean_distance(x,y):
    return sqrt(sum(pow(a-b,2)
for a, b in zip(x, y)))

Print(euclidean_distance([0,3,4,
5],[7,6,3,-1])
```

9.74679434481
[Finished in 0.0s]

Limitations of Euclidean distance (because of covariance)

A measure of how much two variables change together:





For multivariate data, this is better expressed as the *variance-covariance matrix*

Need to rescale variables to eliminate any variance !!!

Variance-Covariance Matrix

Square matrix of covariances between all pairs of variables

$$\begin{array}{c} \\ \boldsymbol{x1} \\ \boldsymbol{x2} \\ \boldsymbol{x3} \end{array} \begin{array}{c} \begin{array}{ccc} \boldsymbol{x1} & \boldsymbol{x2} & \boldsymbol{x3} \end{array} \\ \begin{bmatrix} cov(x1,x1) & cov(x1,x2) & cov(x1,x3) \\ cov(x2,x1) & cov(x2,x2) & cov(x2,x3) \\ cov(x3,x1) & cov(x3,x2) & cov(x3,x3) \end{bmatrix} \end{array}$$

The covariance of a variable with itself is just its variance, so the matrix diagonals contain the variances. Matrix is also symmetrical because cov(*x1*, *x2*) = cov(*x2*, *x1*).

$$\begin{array}{c} \\ \boldsymbol{x1} \\ \boldsymbol{x2} \\ \boldsymbol{x3} \end{array} \begin{array}{c} \begin{array}{ccc} \boldsymbol{x1} & \boldsymbol{x2} & \boldsymbol{x3} \end{array} \\ \begin{bmatrix} var(x1) & cov(x1,x2) & cov(x1,x3) \\ cov(x2,x1) & var(x2) & cov(x2,x3) \\ cov(x3,x1) & cov(x3,x2) & var(x3) \end{bmatrix} \end{array}$$

Minkowski Distance

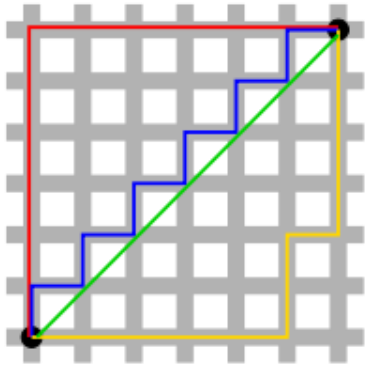
- Minkowski Distance is a generalization of Euclidean Distance

$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

Where r is a parameter, n is the number of dimensions (attributes) and p_k and q_k are, respectively, the kth attributes (components) or data objects p and q.

Minkowski Distance: Examples

- $r = 1$. City block (Manhattan, taxicab, L_1 norm) distance.
  - A common example of this is the Hamming distance, which is just the number of bits that are different between two binary vectors



- $r = 2$. Euclidean distance

- $r \rightarrow \infty$. "supremum" (L_{max} norm, L_∞ norm) distance.
  - This is the maximum difference between any component of the vectors

- Do not confuse r with n, i.e., all these distances are defined for all numbers of dimensions.

Minkowski Distance

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

Distance on Numeric Data: Minkowski Distance

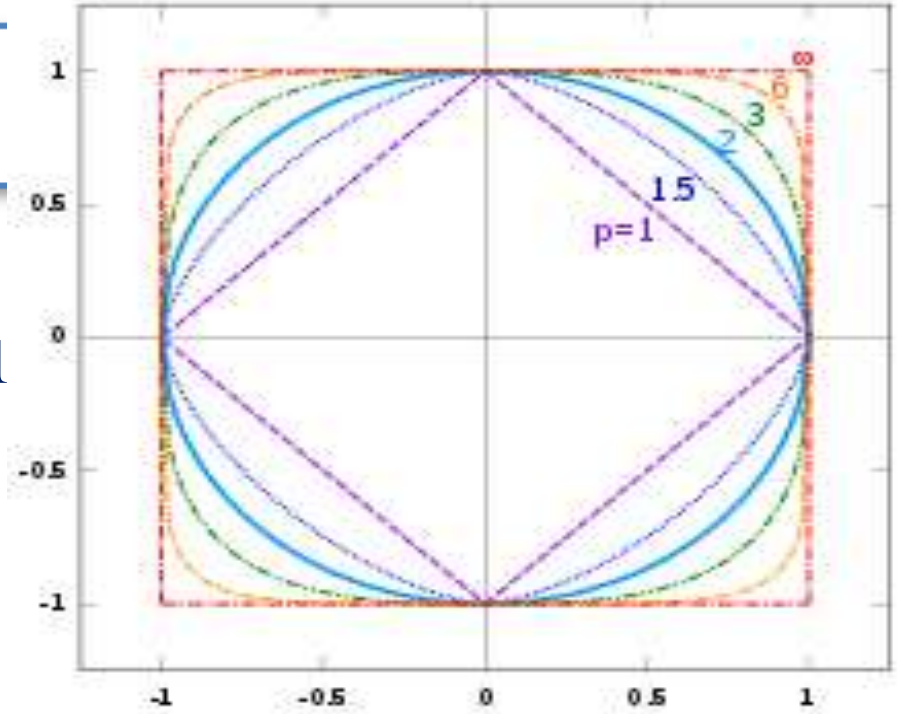
- *Minkowski distance*: A popular distance measure

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects, and h is the order (the distance so defined is also called L-h norm)

- Properties
- $d(i, j) > 0$ if $i \neq j$, and $d(i, i) = 0$ (Positive definiteness)
- $d(i, j) = d(j, i)$ (Symmetry)
- $d(i, j) \leq d(i, k) + d(k, j)$ (Triangle Inequality)
- A distance that satisfies these properties is a metric

What is a norm?



$||x||_1=1$ =>One norm of x equals to 1

$||x||_2=1$ =>two norm of x equals to 1

$||x||_\infty=1$ =>Infinity norm of x equals

$||x||_1$= sum (abs(x1)+abs(x2)=1 =>One norm of x equals to 1

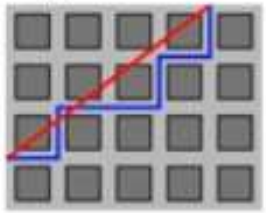
$||x||_2$= (sum (abs(x1)$^{2+}$abs(x2)$^{2})^{1/2}$=1 =>two norm of x equals to 1

$||x||_\infty$ = Max value of X

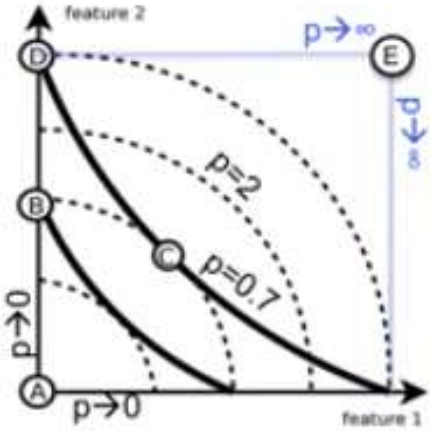
$||x||_1 >= |x||_2 >= ||x||_\infty$

Minkowski distance (*p*-norm): $D(x,x') = \sqrt[p]{\sum_d |x_d - x'_d|^p}$

- $p = 2$: Euclidian
- $p = 1$: Manhattan



- $p \to \infty$: $\max_d |x_d - x'_d|$ ... logical OR
- $p \to 0$: ... logical AND



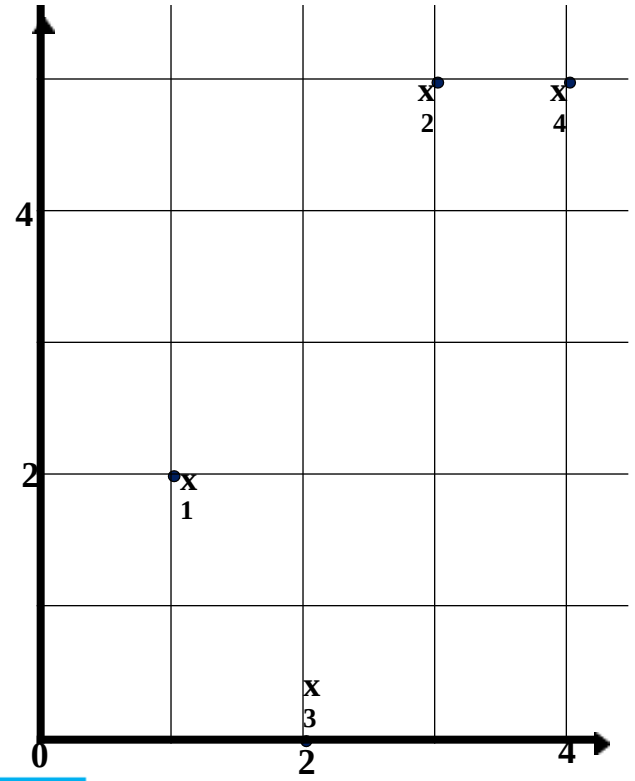
A is the test set
B,C, D, E are training sets
How far away feature1 and feature2 in terms of distance functions ?

For p=2 B and C are the same distance away , while D is more further, E way further
Euclidean find that B and C are at the same distance !
For p=0.7 C and D are the same distance away , while B is closer, E way further
B is different form A only on the second attribute
C is different from both attributes

Example: Minkowski Distance

Dissimilarity Matrices

point	attribute 1	attribute 2
x1	1	2
x2	3	5
x3	2	0
x4	4	5



Manhattan (L_1)

L	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Euclidean (L_2)

L2	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Supremum

L_∞	x1	x2	x3	x4
x1	0			
x2	3	0		
x3	2	5	0	
x4	3	1	5	0

Minkowski Distance in Python

```
from math import*
from decimal import Decimal

def nth_root(value, n_root):
     root_value = 1/float(n_root)
    return round (Decimal(value) ** Decimal(root_value),3)

def minkowski_distance(x,y,p_value):

    return nth_root(sum(pow(abs(a-b),p_value) for a,b in zip(x, y)),p_value)

print (minkowski_distance([0,3,4,5],[7,6,3,-1],3))
```

Mahalanobis distance gets rid of scaling and collinearity issues

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$$

Euclidean distance issue

A	
Income ($ 000's)	Lot Size (000's sq ft)
75.0	19.6
52.8	20.8
64.8	17.2
43.2	20.4
84.0	17.6
49.2	17.6

B	
Income ($)	Lot Size (000's sq ft)
75,000	19.6
52,800	20.8
64,800	17.2
43,200	20.4
84,000	17.6
49,200	17.6

C	
Income ($ 000's)	Lot Size (sq ft)
75.0	19,600
52.8	20,800
64.8	17,200
43.2	20,400
84.0	17,600
49.2	17,600

1. same distance➔ Standardization z= x-avg/variance will fix the issue

2. If variables are correlated ➔count the impact twice and the distance is not accurate

Income ($ 000's)	Lot Size (000's sq ft)	Saving ($ 000's)
75.0	20	27.9
52.8	21	23.3
64.8	17	28.6
43.2	20	19.3
84.0	18	34.8
49.2	18	23.0

Mahalanobis Distance

T indicates a transposed matrix

$$D^2 = (x - \overline{x})^T S^{-1} (x - \overline{x})$$

Matrix of distances from mean

Inverse of covariance matrix

Matrix of:

$(x_1, x_2, \ldots, x_n)$ - $(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$

or

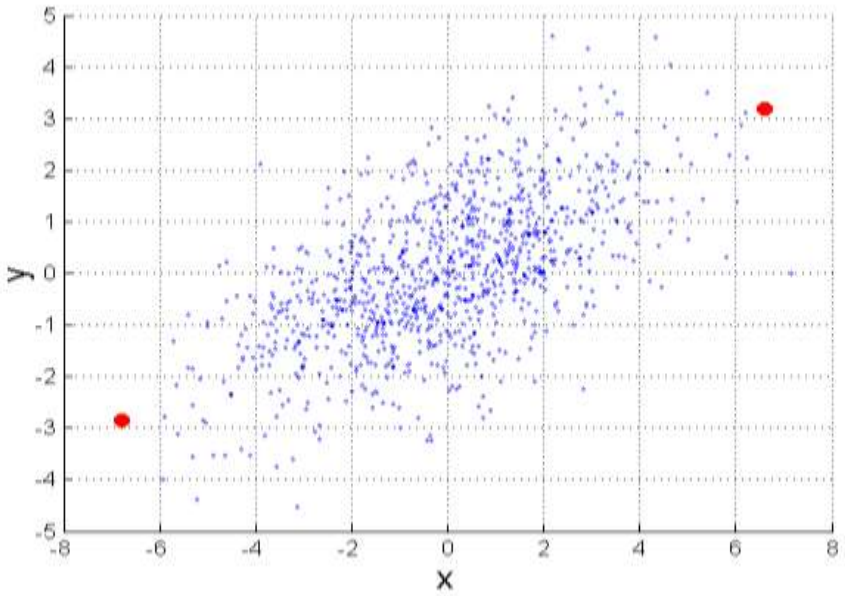
$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} - \begin{bmatrix} \overline{x}_1 \\ \overline{x}_2 \\ \vdots \\ \overline{x}_n \end{bmatrix}$$

$$\begin{bmatrix} s_1^2 & \cdots & \mathrm{Cov}(s_n, s_1) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}(s_1, s_n) & \cdots & s_n^2 \end{bmatrix}$$

Matrix with diagonals = variance of samples 1 ... *n* and cells = covariance of samples (1,2) ... (1,*n*)

Mahalanobis Distance

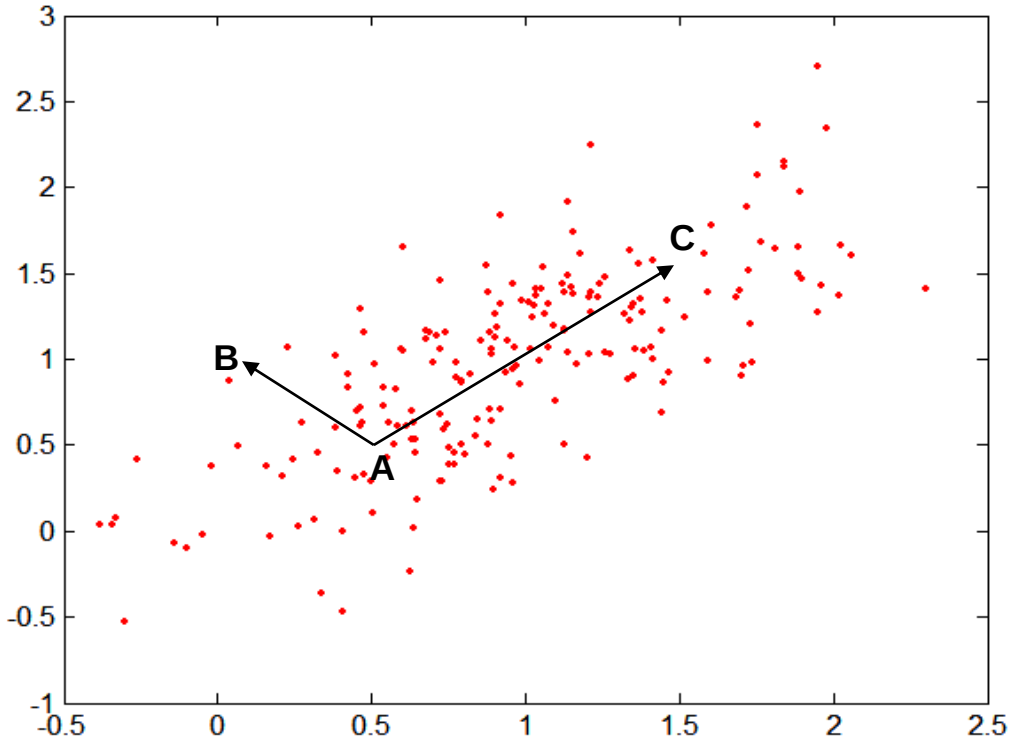
$$\mathbf{mahalanobis}(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^T \Sigma^{-1} (\mathbf{x}-\mathbf{y})$$



Σ is the covariance matrix

For red points, the Euclidean distance is 14.7, Mahalanobis distance is 6.

Mahalanobis Distance



Covariance Matrix:

$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

A: (0.5, 0.5)

B: (0, 1)

C: (1.5, 1.5)

Mahal(A,B) = 5

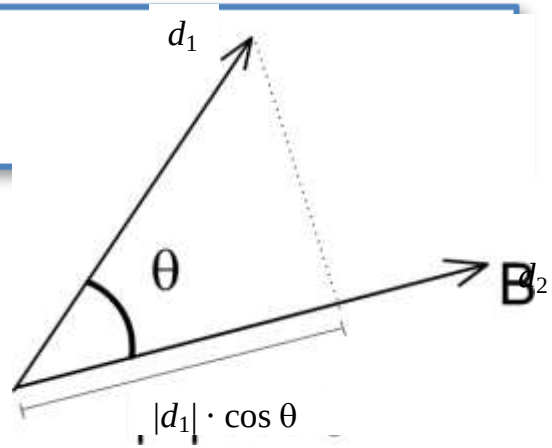
Mahal(A,C) = 4

```
from math import*
from decimal import Decim
 class Similarity():
    """ Five similarity measu

    def euclidean_distance(s
        """ return euclidean di
        return sqrt(sum(pow(a

    def manhattan_distance(
        """ return manhattan c
        return sum(abs(a-b) f

    def minkowski_distance(
        """ return minkowski c
        return self.nth_root(su
          p_value)

    def nth_root(self,value, n
        """ returns the n_root
        root_value = 1/float(n
       return round (Decimal(
```

```
def cosine_similarity(self,x,y):

    """ return cosine similarity between two lists """

    numerator = sum(a*b for a,b in zip(x,y))
    denominator = self.square_rooted(x)*self.square_rooted(y)
    return round(numerator/float(denominator),3)

def square_rooted(self,x):

    """ return 3 rounded square rooted value """

    return round(sqrt(sum([a*a for a in x])),3)

def jaccard_similarity(self,x,y):

""" returns the jaccard similarity between two lists """

    intersection_cardinality = len(set.intersection(*[set(x), set(y)]))
    union_cardinality = len(set.union(*[set(x), set(y)]))
    return intersection_cardinality/float(union_cardinality)
```

```
from similaritymeasures import Similarity

def main():

    """ the main function to create Similarity class instance and get used to it """

    measures = Similarity()

    print measures.euclidean_distance([0,3,4,5],[7,6,3,-1])
    print measures.jaccard_similarity([0,1,2,5,6],[0,2,3,5,7,9])

if __name__ == "__main__":
    main()
```

Cosine Similarity



- If d_1 and d_2 are two document vectors, then

$$\cos(d_1, d_2) = \frac{d_1 \times d_2}{\|d_1\| \times \|d_2\|}$$

where • indicates vector dot product and || d || is the length of vector d.

- Example:

d_1 = **3 2 0 5 0 0 0 2 0 0**

d_2 = **1 0 0 0 0 0 0 1 0 2**

$d_1 \bullet d_2$= 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 =5

$||d_1||$ = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)$^{0.5}$ = (42) $^{0.5}$ = 6.481

$||d_2||$ = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2) $^{0.5}$ = (6) $^{0.5}$ =2.245

cos(d_1, d_2) = .3150

Example: Cosine Similarity

- $\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| \, ||d_2||$,
  where $\bullet$ indicates vector dot product, $||d|$: the length of vector d

- Ex: Find the **similarity** between documents 1 and 2.

$d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$
$d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$

$d_1 \bullet d_2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25$
$||d_1|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$
$||d_2|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)^{0.5} = (17)^{0.5} = 4.12$
$\cos(d_1, d_2) = 0.94$

Cosine Similarity

- A **document** can be represented by thousands of attributes, each recording the *frequency* of a particular word (such as keywords) or phrase in the document.

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

  - Other vector objects: gene features in micro-arrays, …
  - Applications: information retrieval, biologic taxonomy, gene feature mapping, ...
  - Cosine measure: If d_1 and d_2 are two vectors (e.g., term-frequency vectors), then $\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| \, ||d_2||$,

where $\bullet$ indicates vector dot product, $||d||$: the length of vector d

Clustering algorithm: Spherical k-means

```
from math import*

def square_rooted(x):

   return round(sqrt(sum([a*a for a in x])),3)

def cosine_similarity(x,y):

 numerator = sum(a*b for a,b in zip(x,y))
 denominator = square_rooted(x)*square_rooted(y)
 return round(numerator/float(denominator),3)

print cosine_similarity([3, 45, 7, 2], [2, 54, 13, 15])
```

General Approach for Combining Similarities

- Sometimes attributes are of many different types, but an overall similarity is needed.

1. For the k^{th} attribute, compute a similarity, s_k, in the range $[0, 1]$.
2. Define an indicator variable, δ_k, for the k_{th} attribute as follows:

$$\delta_k = \begin{cases} 0 & \text{if the } k^{th} \text{ attribute is a binary asymmetric attribute and both objects have a value of 0, or if one of the objects has a missing values for the } k^{th} \text{ attribute} \\ 1 & \text{otherwise} \end{cases}$$

3. Compute the overall similarity between the two objects using the following formula:

$$similarity(p, q) = \frac{\sum_{k=1}^{n} \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

Using Weights to Combine Similarities

- May not want to treat all attributes the same.
  - Use weights w_k which are between 0 and 1 and sum to 1.

$$similarity(p,q) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

$$distance(p,q) = \left(\sum_{k=1}^{n} w_k |p_k - q_k|^r \right)^{1/r}$$

Extended Jaccard Coefficient (Tanimoto)

- Variation of Jaccard for continuous or count attributes
  - Reduces to Jaccard for binary attributes

$$EJ(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \mathbf{x} \cdot \mathbf{y}}$$

Comparison of Proximity Measures

- Domain of application
  - Similarity measures tend to be specific to the type of attribute and data
  - Record data, images, graphs, sequences, 3D-protein structure, etc. tend to have different measures
- However, one can talk about various properties that you would like a proximity measure to have
  - Symmetry is a common one
  - Tolerance to noise and outliers is another
  - Ability to find more types of patterns?
  - Many others possible
- The measure must be applicable to the data and produce results that agree with domain knowledge