Tutorial on Chapter 1 Data: type, quality, dis/similarity

Reference:
Chapter 2 Data, i2DM
This tutorial doesn't cover Section 2.3 Data preprocessing

Acknowledgement:
i2DM (Tan, Steinbach, Kumar (2018) Introduction to Data Mining , 2nd Ed, Pearson) Pearson Press
For the course AMA546 Statistical Data Mining
Lecturer: Dr. Catherine Liu
AMA, PolyU, HKSAR

In [7]:

```
import pandas as pd
import numpy as np
```

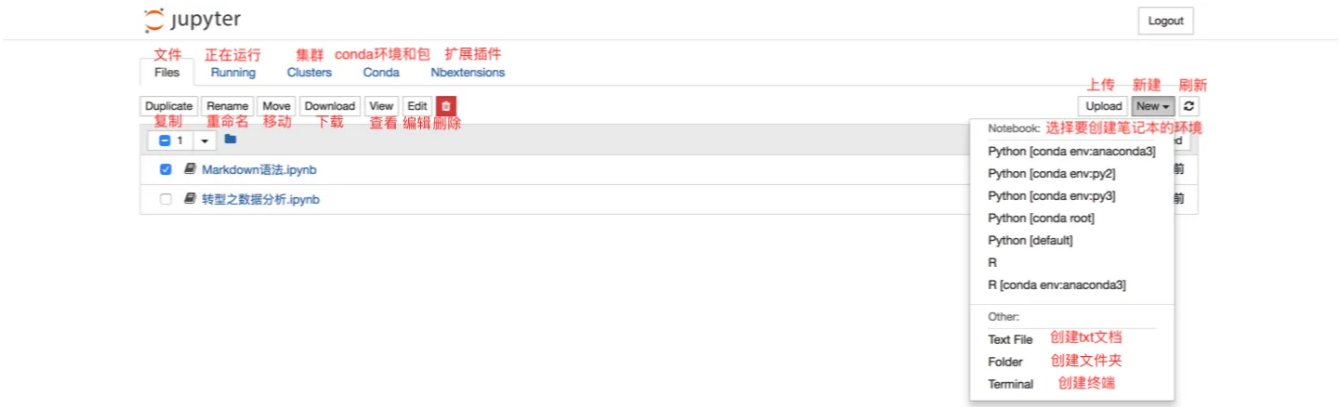
executed in 343ms, finished 02:25:38 2023-01-31

`pandas` is a Python package that offers data structures and operations for manipulating numerical tables.
`numpy` is a Python package that used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.
The `import pandas` portion of the code tells Python to bring the pandas library into your current environment.
The `as pd` portion of the code then tells Python to give pandas the alias of pd. This allows you to use pandas functions by simply typing pd.
Refer to the Python modules (https://docs.python.org/3/tutorial/modules.html) for more information.

1 How to use jupyter notebook?

Project Jupyter is a project with goals to develop open-source software, open standards, and services for interactive computing across multiple programming languages. It was spun off from IPython in 2014, named after the three core programming languages supported by Jupyter, which are Julia, Python and R. Project Jupyter has developed and supported the interactive computing products Jupyter Notebook, JupyterHub, and JupyterLab.

You can manage the files in the File page:



Once you click a `.ipynb` file, you enter into the Notebook page:

There are two modes in the jupyter notebook, which is the Command mode and the Edit mode:

- Command mode:
  - In the command mode, you can manage the cells but not the codes in the cells.
  - Press esc to enter the command mode.
  - In command mode, the cell border is gray, and the left border line is blue.
- Edit mode:
  - In the edit mode, you can edit code or documents within cells.
  - Press enter or return to enter editing mode.
  - In edit mode, the cell border and the left border line are green

The cells have two common modes, Code mode and Markdown mode:

- In the Code mode, you can type and run programming codes like python.
- In the Markdownn mode, you can edit the markdown codes to explain your codes.
- To switch between two modes, you may click m(for markdown) or y(for code) in the Command mode.

If you willing to prettify your jupyter notebook page, you may want to try:

- nbextensions (https://github.com/ipython-contrib/jupyter_contrib_nbextensions/blob/master/docs/source/install.md) or here (https://blog.csdn.net/weixin_44015669/article/details/104975271)

If you wish to type LaTeX in jupyter notebook, you may refer to:

- Help->LaTeX_envs help (http://127.0.0.1:8888/nbextensions/latex_envs/doc/latex_env_doc.html)

For more information, you may read:

- Help->Keyboard Shortcuts (https://mljar.com/blog/jupyter-notebook-shortcuts/)
- 最详尽使用指南：超快上手Jupyter Notebook (https://zhuanlan.zhihu.com/p/32320214)
- Jupyter Notebook Tutorial (https://github.com/aparrish/dmep-python-intro/blob/master/jupyter-notebook-tutorial.ipynb)

2 Exercises

2.1 Question 1

In Example 2.1 in Page 73, the author wrote:
... ...
The first few rows of the file are as follows:

field 1	field 2	field 3	field 4	field 5
12	232	33.5	0	10.7
20	121	16.9	2	210.1
27	165	24	0	427.6

... ...
Data Miner: Interesting. Were there any other problems?
Statistician: **Yes, fields 2 and 3 are basically the same**, but I assume that you probably noticed that.
Data Miner: Yes, but these fields were only weak predictors of field 5.
... ...
The statistician says, “Yes, fields 2 and 3 are basically the same.” Can you tell from the sample data that why she says that?

Answer: (See Example 2.1)

In [2]:

```
# initialize the DataFrame with data from the textbook and assign to variable df_q1
df_q1 = pd.DataFrame([
    {"field 1":"012","field 2":232,"field 3":33.5,"field 4":0,"field 5":10.7},
    {"field 1":"020","field 2":121,"field 3":16.9,"field 4":2,"field 5":210.1},
    {"field 1":"027","field 2":165,"field 3":24.0,"field 4":0,"field 5":427.6},
])
df_q1 # print df_q1
```

executed in 12ms, finished 00:10:18 2023-01-31

Out[2]:

	field 1	field 2	field 3	field 4	field 5
0	012	232	33.5	0	10.7
1	020	121	16.9	2	210.1
2	027	165	24.0	0	427.6

In Pandas, the function pd.DataFrame (https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html) constructs a 2 dimensional data structure, like a 2 dimensional matrix, or a table with rows and columns.
Here we use the data from a dictionary list to initialize the Pandas DataFrame.
List (https://developers.google.com/edu/python/lists) is used to store multiple items in a single variable.
Dictionary (https://python101.pythonlibrary.org/chapter3_lists_dicts.html) is used to store data values in key:value pairs.

Typesetting math: 100%

In [3]:

```
1 pd.DataFrame(df_q1['field 2'] / df_q1['field 3'], columns=['ratio']) # calculate the ratio of field 2 and field 3
```

executed in 4ms, finished 00:10:18 2023-01-31

Out[3]:

	ratio
0	6.925373
1	7.159763
2	6.875000

Note that $\frac{\text{field 2}}{\text{field 3}} \approx 7$.

While it can be dangerous to draw conclusions from such a small sample size, the two fields seem to contain essentially the same information.

2.2 Question 2

Classify the following attributes as:

- binary,
- discrete,
- continuous.

Also classify them as:

- qualitative:
  - nominal or,
  - ordinal,
- quantitative
  - interval,
  - ratio.

Some cases **may have more than one interpretation**, so briefly indicate your reasoning if you think there may be some ambiguity.

Example: Age in years.

Answer: Discrete, quantitative, ratio

(a) Time in terms of AM or PM.

(b) Brightness as measured by a light meter.

(c) Brightness as measured by people's judgments.

```
e.g. bright, medium and dark
```

(d) Angles as measured in degrees between 0◦ and 360◦.

(e) Bronze, Silver, and Gold medals as awarded at the Olympics.

(f) Height above sea level.

(g) Number of patients in a hospital.

(h) ISBN numbers for books. (Look up the format on the Web.)

```
e.g. ISBN-13: 9780137506286 for i2DM (2nd Ed, Pearson )
```

(i) Ability to pass light in terms of the following values: opaque, translucent, transparent.

(j) Military rank.

(k) Distance from the center of campus.

(l) Density of a substance in grams per cubic centimeter.

(m) Coat check number. (When you attend an event, you can often give your coat to someone who, in turn, gives you a number that you can use to claim your coat when you leave.)

Answer: (See Table 2.3.)

In [4]:

```
df = pd.DataFrame([['qualitative', 'nominal', '', '(h), (m)', ''],
                   ['qualitative', 'ordinal', '(a)', '(c), (e), (i), (j)',''],
                   ['quantitative', 'interval', '', '', '(f)*, (k)**'],
                   ['quantitative', 'ratio', '', '(g), (l)', '(b), (d), (f), (k)']],
                  columns=[' ', '', 'binary', 'discrete', 'continuous'])
df.set_index([' ', '', 'binary', 'discrete', 'continuous']) # merge the cells in the first column
```

executed in 8ms, finished 00:10:18 2023-01-31

Out[4]:

		binary	discrete	continuous
qualitative	nominal		(h), (m)	
	ordinal	(a)	(c), (e), (i), (j)	
quantitative	interval			(f)*, (k)**
	ratio		(g), (l)	(b), (d), (f), (k)

*depends on whether sea level is regarded as an arbitrary origin
**depends on the definition of the distance

2.3 Question 3

You are approached by the marketing director of a local company, who believes that he has devised a foolproof way to measure customer satisfaction.

He explains his scheme as follows: “I just **keep track of the number of customer complaints for each product**. Because **counts are ratio attributes, my measure of product satisfaction must be a ratio attribute**. But **my boss he told me that I had overlooked the obvious**, and that my measure was worthless. And it turns out that **our best selling product had the worst satisfaction since it had the most complaints**. Could you help me set him straight?”

2.3.1 Q3(a)

What can you say about the attribute type of the original product satisfaction attribute?

Answer: (See Table 2.3.)

We can say nothing about the attribute type of the original measure. Because it is only a pure counts of customer complaints for each product rather than a satisfication rate.
For example, two products that have the same level of customer satisfaction may have different numbers of complaints. And two product that have the same number of complaints may have different levels of customer saticfaction.
Therefore we need a better measure.

2.3.2 Q3(b)

Who is right, the marketing director or his boss? If you answered, his boss, what would you do to fix the measure of satisfaction?

Answer:

The boss is right. The number of customer complaints for each product needs to be compared with the total number of products sold.
A better measure is given by

$$\text{Satisfaction rate of the product} = \frac{\text{number of complaints for the product}}{\text{total number of sales for the product}},$$

which describes the customer complaints rate of each product.

2.4 Question 4

A few months later, you are again approached by the same marketing director as in Exercise 3. This time, he has devised a better approach to measure the extent to which a customer prefers one product over other, similar products.

He explains, “When we develop new products, we typically create several variations and evaluate which one customers prefer. **Our standard procedure is to ask them to rank all of the product variations at one time**. However, our test subjects are very indecisive, especially when there are more than two products."

"**I suggested that we perform the comparisons in pairs and then use these comparisons to get the rankings**. Thus, if we have three product variations, we have the customers compare variations 1 and 2, then 2 and 3, and finally 3 and 1. But the **customers cannot come up with a consistent ranking from the results**. Can you help me?”

2.4.1 Q4(a)

Is the marketing director in trouble? Will his approach work for generating an ordinal ranking of the product variations in terms of customer preference? Explain.

Answer: (See Table 2.3.)

Yes. The marketing director's approach may not work for generating an ordinal ranking.
For example, a customer may give inconsistent rankings with `1>2`, `2>3` and `3>1`. (1, 2 and 3 represent the product variations)

2.4.2 Q4(b)

Is there a way to fix the marketing director's approach? More generally, what can you say about trying to create an ordinal measurement scale based on pairwise comparisons?

Answer:

One solution: For the case in Q4(a), only the first two comparisons are made: 1 and 2, then 2 and 3. Then the ranking from the customer is consistent: `1>2` and `2>3`.

A more general solution: The marketing director should take the customer choices into account when designing the pairs of the product variations to be compared. In other words, each of the respondents compares the relative importance of each pair of items using a specially designed questionnaire. Refer to the Analytic_hierarchy_process (https://en.wikipedia.org/wiki/Analytic_hierarchy_process) for more information.
In general, creating an ordinal measurement scale based on pairwise comparison is difficult because of possible inconsistencies.

2.4.3 Q4(c)

For the original product evaluation scheme, the overall rankings of each product variation are found by computing its average over all test subjects. Comment on whether you think that this is a reasonable approach. What other approaches might you take?

Answer:

First, there is the issue that the scale is likely not an interval or ratio scale. In other words, taking the average is meaningless for them.
Nonetheless, for practical purposes, an average may be good enough. A more important concern is that a few extreme ratings might result in an overall rating that is misleading. In this case, the median or a trimmed mean might be a better choice.

e.g. Suppose five customers are asked to rate ten subjects. The ranking is from 1 (best) to 10 (worst). For simplicity, only the rankings for the first three subjects are listed below.

In [17]:

```
df_q4c = pd.DataFrame({'subject 1' : [1, 2, 4, 3, 7],
                       'subject 2' : [4, 4, 5, 5, 4],
                       'subject 3' : [3, 3, 3, 4, 9]}) # initialize the DataFrame by a dictionary
df_q4c.index = ['customer 1', 'customer 2', 'customer 3', 'customer 4', 'customer 5']
df_q4c.loc['Mean'] = df_q4c[:5].mean() # calculate the mean with respect to rows
df_q4c.loc['Median'] = df_q4c[:5].median() # calculate the median with respect to rows
df_q4c
```

executed in 28ms, finished 03:17:19 2023-01-31

Out[17]:

	subject 1	subject 2	subject 3
customer 1	1.0	4.0	3.0
customer 2	2.0	4.0	3.0
customer 3	4.0	5.0	3.0
customer 4	3.0	5.0	4.0
customer 5	7.0	4.0	9.0
Mean	3.4	4.4	4.4
Median	3.0	4.0	3.0

From the ranking table, we have:

- *Mean: subject 3 = subject 2 > subject 1,*
- *Median: subject 3 = subject 1 > subject 2,*

The reason is that there is a extreme rating 9 in subject 3 from customer 5.

2.5 Question 5

Can you think of a situation in which identification numbers would be useful for prediction?

Answer:

The NetID of a master student in PolyU is a good predictor of the graduation year. (Assume the duration of the project is known and no deferred graduation)

```
e.g. If the NetID is 22057000G, and we know the project will last one year. Then the predicted graduation year will be
2023.
```

2.6 Question 6

Which of the following quantities is likely to show **more spatial autocorrelation**: daily rainfall or daily temperature? Why?

Answer: (See Section 2.1.1)

Autocorrelation (https://en.wikipedia.org/wiki/Autocorrelation) is the correlation between observations of a random variable as a function of time lag between them.

Spatial autocorrelation means closer places share more similar feature values. For two places close to each other, they are more common to be similar in temperature than in rainfall since rainfall can be very localized.
Hence daily temperature is likely to show more spatial autocorrelation.

2.7 Question 7

Discuss why a document-term matrix is an example of a data set that has asymmetric discrete or asymmetric continuous features.

Answer: (See Section 2.1.2)

The ij^{th} entry of a document-term matrix is the number of times that term j occurs in the document i.

- It represents the number of occurrences, so the data set is **discrete**.
- The asymmetric attribute means only non-zero attribute values are regarded as important. Even though documents have thousands or tens of thousands of attributes (terms), each document is sparse since it has relatively few nonzero attributes. Thus, zero entries are not very meaningful in describing and comparing documents. So the data set is **asymmetric**.

If we apply a TFIDF normalization to terms and normalize the documents to have an L2 norm of 1, then this creates a term-document matrix with **continuous** features. However, the features are still **asymmetric** because these transformations do not create non-zero entries for any entries that were previously 0, and thus, zero entries are still not very meaningful.

Refer to Question 15 for more details.

2.8 Question 8

Many sciences rely on observation instead of (or in addition to) designed experiments. Compare the **data quality issues** involved in **observational science** with those of **experimental science and data mining**.

Answer:

- An **observational science** is a science where researchers observe the effect of an intervention without trying to change who is or isn't exposed to it. Therefore, researchers do not have completely control of the quality of the data they obtain. According to Example 2.6 in the textbook, the SST data collected from ships or buoys are different from the data gathered from satellites.
- In **experimental science**, researchers introduce an intervention and study the effects. The subjects in experimental studies are usually grouped by chance. The results are much more reliable because almost all other irrelevant factors are excluded. Therefore, researchers can better control the quality of the data they obtain. For example, medical scientists in the laboratory can control irrelevant factors by randomly assigning patients to different groups.

Thus, it is necessary to work with the data available in observational science, rather than data from a carefully designed in experimental science. In that sense, **data analysis for observational science resembles data mining**.

Supplement material:

- **Observational studies** are ones where researchers observe the effect of a risk factor, diagnostic test, treatment or other intervention **without making the decision that who is or isn't exposed to it**.
- **Experimental studies** are ones where researchers introduce an intervention and study the effects. The **subjects in experimental studies are usually grouped by chance**.
  - **Randomized controlled trial** (RCT): **Eligible** people are **randomly assigned** to one of two or more groups. One group receives the intervention (such as a new drug) while the **control group** receives nothing or an inactive placebo. The researchers then study what happens to people in each group. **Any difference in outcomes can then be linked to the intervention**.

The strengths and weaknesses of a study design should be seen in light of the kind of question the study sets out to answer.

- **Sometimes, observational studies are the only way** researchers can explore certain questions. For example, it would be unethical to design a randomized controlled trial deliberately exposing workers to a potentially harmful situation. If a health problem is a rare condition, a observational study based on the existing cases may be the most efficient way to identify potential causes.
- However, **the results of observational studies are, by their nature, open to dispute**. They run the risk of containing confounding biases. Example: A cohort study might find that people who meditated regularly were less prone to heart disease than those who didn't. But the link may be explained by the fact that people who meditate also exercise more and follow healthier diets. In other words, although a cohort is defined by one common characteristic or exposure, they may also share other characteristics that affect the outcome.

The **RCT is still considered the "gold standard"** for producing reliable evidence because little is left to chance. But there's a growing evidence that such research is not perfect, and that many questions simply can't be studied using this approach.

- Such research is time-consuming and expensive. It may take years before results are available. Also, intervention research is often restricted by how many participants researchers can manage or how long participants can be expected to live in controlled conditions. As a result, an RCT would not be the right kind of study to pick up on outcomes that take a long time to appear or that are expected to affect a very minute number of people.

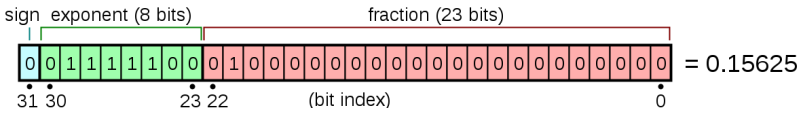
Refer to the Observational vs. experimental studies (https://www.iwh.on.ca/what-researchers-mean-by/observational-vs-experimental-studies) for more information.

2.9 Question 9

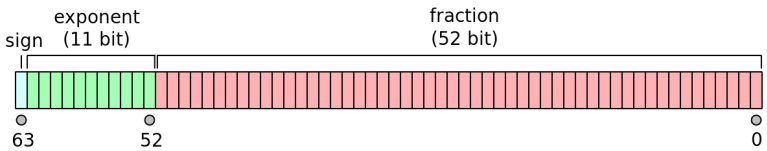
Discuss the **difference between the precision of a measurement and the terms single and double precision**, as they are used in computer science, typically to represent **floating-point numbers that require 32 and 64 bits**, respectively.

Answer: (See Section 2.2.1)

- The precision of a measurement (https://en.wikipedia.org/wiki/Accuracy_and_precision) is the closeness of repeated measurements (of the same quantity) to one another.
- The precision of floating point numbers (https://en.wikipedia.org/wiki/Floating-point_arithmetic) is the maximum precision of the number stored in the computer. More explicity, precision is often expressed in terms of the number of significant digits used to represent a value.
- A 32bits single-precision float (https://www.ibm.com/support/pages/single-precision-floating-point-accuracy) only has about 7 decimal digits of precision (actually the log base 10 of 2^{23}, or about 6.92 digits of precision).



- A 64bits single-precision float (https://stackoverflow.com/questions/9999221/double-precision-decimal-places) only has about 16 decimal digits of precision (actually the log base 10 of 2^{53}, or about 15.95 digits of precision).



In [18]:

```
print('The digits of precision for 32bits:', round(23*np.log10(2), 2)) # single precision
print('The digits of precision for 64bits:', round(53*np.log10(2), 2)) # double precision
```

executed in 6ms, finished 03:17:21 2023-01-31

```
The digits of precision for 32bits: 6.92
The digits of precision for 64bits: 15.95
```

2.10 Question 10

Give at least two advantages to working with data stored in text files instead of in a binary format.

Answer:

- **Text files can be inspected and modified by viewing it with a text editor or typing the file.** Because the data stored in text files uses the ASCII format, which is human-readable graphic characters. But the binary format stores data 0 and 1, which is abstract and difficult for humans to read.
- **Text files are portable both across systems and programs.** On the contrary, the binary files cannot easily be transferred from one computer system to another due to variations in the internal representation which varies from one computer to another.

Supplement material:

A text file (https://en.wikipedia.org/wiki/Text_file) is the one in which data is stored in the form of ASCII characters and is normally used for storing a stream of characters. Text files are organized around lines, each of which ends with a newline character ('\n'). The source code files are themselves text files.

A binary file (https://en.wikipedia.org/wiki/Binary_file) is the one in which data is stored in the file in the same way as it is stored in the main memory for processing. It is stored in binary format instead of ASCII characters. It is normally used for storing numeric information (int, float, double). Normally a binary file can be created only from within a program and its contents can be read only by a program.

Refer to the Difference Between Text File and Binary File (https://www.geeksforgeeks.org/difference-between-cpp-text-file-and-binary-file/) for more information.

ASCII (https://en.wikipedia.org/wiki/ASCII), abbreviated from 'American Standard Code for Information Interchange', is a standard data-encoding format for electronic communication between computers. Because of technical limitations of computer systems at the time it was invented, ASCII has just 128 code points, of which only 95 are printable characters, which severely limited its scope. Nowadays, all modern computer systems instead use Unicode, which has millions of code points, but the first 128 of these are the same as the ASCII set.

2.11 Question 11

Distinguish between noise and outliers. Be sure to consider the following questions.

2.11.1 Q11(a)

Is noise ever interesting or desirable? Outliers?

Answer: (See Section 2.2.1)

According to Chapter 2, noise is never interesting or desirable. Outliers can be interesting or desirable.

Supplement material:

- **Noise** is the random component of a measurement error.
- **Outliers**, also known as the **anomalous** objects or values, are data objects that have:
  - characteristics that are different from most of the other data objects in the data set, or
  - values of an attribute that are unusual with respect to the typical values for that attribute.
- Unlike noise, **outliers is a broader concept** that includes not only errors but also discordant data that may arise from the natural variation within the population or process.

2.11.2 Q11(b)

Can noise objects be outliers?

Answer: (See Section 2.1.2)

Yes. Noise in attribute values can make the data look more randomized or unusual.
Thus, it is possible that some instances in noisy data will appear as outliers.

2.11.3 Q11(c)

Are noise objects always outliers?

Answer:

No. Random distortion can result in an object or value much like a normal one.

2.11.4 Q11(d)

Are outliers always noise objects?

Answer:

No, outliers can be discordant data that may arise from the natural variation within the population or process.
Therefore, outliers are not always noise objects.

2.11.5 Q11(e)

Can noise make a typical value into an unusual one, or vice versa?

Answer:

The source of noise in data can randomly make some values appear as unusual. Or some outliers as typical data objects.

2.12 Question 12 (KNN algorithm)

Consider the problem of finding the K nearest neighbors of a data object. A programmer designs Algorithm 2.1 for this task.

Algorithm 2.1 Algorithm for finding K nearest neighbors.

1: **for** $i = 1$ to *number of data objects* **do**
2: Find the distances of the i^{th} object to all other objects.
3: Sort these distances in decreasing order.
(Keep track of which object is associated with each distance.)
4: **return** the objects associated with the first K distances of the sorted list
5: **end for**

2.12.1 Q12(a)

Describe the potential problems with this algorithm if there are duplicate objects in the data set. Assume the distance function will only return a distance of 0 for objects that are the same.

Answer:

There are several problems.

- First, if there are some objects with the same distance, the order of duplicate objects on the nearest neighbor list will depend on the order of objects in the data set.
- Second, according to the first point, the nearest neighbor of an object may not be itself but objects with the same distance.
- Third, if there are enough duplicates, the nearest neighbor list may only consist of duplicates.

2.12.2 Q12(b)

How would you fix this problem?

Answer:

There are various approaches depending on the situation.
One approach is to to keep only one object for each group of duplicate objects. In this case, each neighbor can represent either a single object or a group of duplicate objects.

Supplement material:

K-nearest neighbors algorithm:

In KNN algorithm above, the output is a list with K items.

An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

Python relization of Algorithm 2.1

In [19]:

```
### Algorithm for finding K nearest neighbors, though the KNN is for purpose of classification.
import math
import numpy as np
from matplotlib import pyplot
from collections import Counter

def k_nearest_neighbors(data, predict, k=5):
    # S2: Find the distances of the ith object to all other objects
    distances = []
    for group in data:
        for features in data[group]:
            euclidean_distance = np.sqrt(np.sum((np.array(features)-np.array(predict))**2))
            distances.append([euclidean_distance, group])
    # S3: Sort these distances in decreasing order
    sorted_distances = [i[1] for i in sorted(distances)]
    # S4: Return the objects associated with the first K distances of the sorted list
    print(pd.DataFrame(sorted(distances))[:k])
    top_nearest = sorted_distances[:k]
    group_res = Counter(top_nearest).most_common(1)[0][0]
    return group_res
```

executed in 11ms, finished 03:17:32 2023-01-31

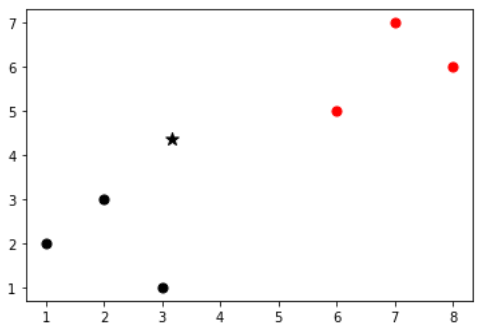
In [20]:

```
dataset = {'black': [[1, 2], [2, 3], [3, 1]], 'red': [[6, 5], [7, 7], [8, 6]]}
new_features = [3.5+np.random.normal(1), 4.2+np.random.normal(1)]  # new observation
for i in dataset:
    for ii in dataset[i]:
        pyplot.scatter(ii[0], ii[1], s=50, color=i)

which_group = k_nearest_neighbors(dataset, new_features, k=3)
print(which_group)
pyplot.scatter(new_features[0], new_features[1], s=100, color=which_group, marker='*')
pyplot.show()
```

executed in 113ms, finished 03:17:33 2023-01-31

```
          0      1
0  1.805543  black
1  2.897707    red
2  3.217218  black
black
```



The random new feature is marked as a star, you can run the code several times to see how the KNN algorithm behaves.

2.13 Question 13

The following attributes are measured for members of a herd of Asian elephants: *weight*, *height*, *tusk length*, *trunk length*, and *ear area*. Based on these measurements, what sort of similarity measure from Section 2.4 would you use to compare or group these elephants? Justify your answer and explain any special circumstances.

Answer:

- The attributes dataset are **numerical** and **dense**, and the **range of their values can vary widely** depending on the scale used to measure them. Because the attributes are **not asymmetric** and the **magnitude of an attribute matters**, the cosine and correlation measure are not suitable.

- The attributes associated with an Asian elephant are **correlated**, that means weight is dependent on height; tusk length, trunk length and ear area are correlated and can be dependent on an attribute such as age of the elephant.

A measure of proximity known as Mahalanobis distance is useful when attributes are correlated, and have different range of values. Also, this measure is scale-invariant. Therefore, in the presented problem, Mahalanobis distance measure is the most suitable to measure to compare the group of elephants.

2.14 Question 14 (Data preprocessing)

You are given a set of m objects that is divided into K groups, where the ith group is of size mi. If the goal is to obtain a sample of size n < m, what is the difference between the following two sampling schemes? (Assume sampling with replacement.)
(a) We randomly select n ∗ mi/m elements from each group.
(b) We randomly select n elements from the data set, without regard for the group to which an object belongs.

Answer:

- The first scheme ensures the number of samples obtained from each group is proportional to the size of the group.
- However, the number of samples from each group will not proportional to the size of the group in the second scheme. More specifically, the second scheme only guarantees that, on average, the number of objects from each group will be n ∗ mi/m.

Supplement material:

Python relization:
Suppose m=8, K=3, m1=2, m2=2, m3=4 and n=4.

In [21]:

```
m = 8
m1 = 2
m2 = 2
m3 = 4
group = [m1, m2, m3]

n = 4
# (a)
n1 = n * m1 / m
n2 = n * m2 / m
n3 = n * m3 / m

#(b)
n_1 = n_2 = n_3 = 0
for i in np.arange(1,5):
    r = np.random.uniform(0,1,1)
    if r<m1/m:
        n_1+=1
    elif r<(m1+m2)/m:
        n_2+=1
    else:
        n_3+=1

df = pd.DataFrame({ 'group 1' : [m1, n1, n_1],
                    'group 2' : [m2, n2, n_2],
                    'group 3' : [m3, n3, n_3]})
df.index = ['group size', 'sample size under (a)', 'sample size under (b)']
df
```

executed in 20ms, finished 03:17:36 2023-01-31

Out[21]:

	group 1	group 2	group 3
group size	2.0	2.0	4.0
sample size under (a)	1.0	1.0	2.0
sample size under (b)	2.0	0.0	2.0

The scheme (a) ensures the number of samples obtained from each group is proportional to the size of the group.
While the number of samples from each group will not proportional to the size of the group in the second scheme.

2.15 Question 15

Consider a document-term matrix, where tf_{ij} is the frequency of the i^{th} word (term) in the j^{th} document and m is the number of documents. Consider the variable transformation that is defined by

$$tf'_{ij} = \underbrace{tf_{ij}}_{TF} * \underbrace{log\frac{m}{df_i}}_{IDF} \tag{1}$$

where df_i is the number of documents in which the i^{th} term appears and is known as the **document frequency** of the term. This transformation is known as the **inverse document frequency** transformation.

2.15.1 Q15(a)

What is the effect of this transformation if a term **occurs in one document**? **In every document**?

Answer:

- The term occurs in one document has the maximum weight: $df_i = 1, log\frac{m}{df_i} = logm$ and $tf'_{ij} = tf_{ij} * logm$.
- The term occurs in every document has 0 weight: $df_i = m, log\frac{m}{df_i} = 0$ and $tf'_{ij} = 0$.

2.15.2 Q15(b)

What might be the **purpose of the inverse document frequency transformation**?

Answer:

The purpose of the **inverse document frequency** transformation is to **offset** the **document frequency** of a word by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

Supplement material:

TFIDF (https://en.wikipedia.org/wiki/Tf%E2%80%93idf), short for "term frequency–inverse document frequency", is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. TFIDF is one of the most popular term-weighting schemes today.

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

The TFIDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

Python realization of TFIDF:

In [22]:

```
### Corpus preprocessing, to acquire the word frequency.
# Reference: https://medium.com/@imamun/creating-a-tf-idf-in-python-e43f05e4d424
import sklearn as sk
import math

# load up our sample sentences
first= 'The car is driven on the road'
second= 'The truck is driven on the highway'
# split so each word has their own string
first = first.split(" ")
second= second.split(" ")
# join them to remove common duplicate words
total= set(first).union(set(second))
# print(total)
# Now let's add a way to count the words using a dictionary key-value pairing for both sentences
wordDictA = dict.fromkeys(total, 0)
wordDictB = dict.fromkeys(total, 0)
for word in first:
    wordDictA[word]+=1
for word in second:
    wordDictB[word]+=1 # use dictionary to store the word frequency; list doesn't work
```

executed in 8ms, finished 03:17:40 2023-01-31

In [23]:

```
### Define the components of the TF-IDF algorithm.
# define the TF function:
def computeTF(wordDict, bow): # def keyword is used to define a function
    tfDict = {}
    bowCount = len(bow)
    for word, count in wordDict.items():
        tfDict[word] = count/float(bowCount)
    return tfDict # return keyword is used to return a function value

# the IDF function:
def computeIDF(docList):
    idfDict = {}
    N = len(docList)
    idfDict = dict.fromkeys(docList[0].keys(), 0)
    for doc in docList:
        for word, val in doc.items():
            if val > 0:
                idfDict[word] += 1
    for word, val in idfDict.items():
        idfDict[word] = math.log10(N / float(val))
    return idfDict

# the TF*IDF function:
def computeTFIDF(tfBow, idfs):
    tfidf = {}
    for word, val in tfBow.items():
        tfidf[word] = val*idfs[word]
    return tfidf
```

executed in 12ms, finished 03:17:41 2023-01-31

In [24]:

```
### Apply the algorithm to the corpus
# tf table
tfFirst = computeTF(wordDictA, first)
tfSecond = computeTF(wordDictB, second)
# Converting to dataframe for visualization
tf_df= pd.DataFrame([tfFirst, tfSecond])

# inputing our sentences in the log file
idfs = computeIDF([wordDictA, wordDictB])
# running our two sentences through the IDF:
idfFirst = computeTFIDF(tfFirst, idfs)
idfSecond = computeTFIDF(tfSecond, idfs)
# Converting to dataframe for visualization
idf= pd.DataFrame([idfFirst, idfSecond])
```

executed in 11ms, finished 03:17:41 2023-01-31

In [25]:

```
pd.DataFrame([wordDictA, wordDictB]) # Frequency table
```

executed in 16ms, finished 03:17:42 2023-01-31

Out[25]:

	truck	road	is	car	driven	The	highway	the	on
0	0	1	1	1	1	1	0	1	1
1	1	0	1	0	1	1	1	1	1

In [26]:

```
tf_df # TF table
```

executed in 14ms, finished 03:17:42 2023-01-31

Out[26]:

	truck	road	is	car	driven	The	highway	the	on
0	0.000000	0.142857	0.142857	0.142857	0.142857	0.142857	0.000000	0.142857	0.142857
1	0.142857	0.000000	0.142857	0.000000	0.142857	0.142857	0.142857	0.142857	0.142857

In [27]:

```
idf # TF-IDF table
```

executed in 17ms, finished 03:17:42 2023-01-31

Out[27]:

	truck	road	is	car	driven	The	highway	the	on
0	0.000000	0.043004	0.0	0.043004	0.0	0.0	0.000000	0.0	0.0
1	0.043004	0.000000	0.0	0.000000	0.0	0.0	0.043004	0.0	0.0

Typesetting math: 100%

Compare the Count table, TF table and TF-IDF table:

- *`The, the, driven, on and is` are no longer important in the TF-IDF table. This is because they are observed in every documents.*
- *The significance of `truck, highway, road and car` are standing out.*

2.16 Question 16

This exercise compares and contrasts some similarity and distance measures.

2.16.1 Q16(a)

For binary data, the L1 distance corresponds to the Hamming distance; that is, the number of bits that are different between two binary vectors. The Jaccard similarity is a measure of the similarity between two binary vectors. Compute the **Hamming distance** and the **Jaccard similarity** between the following two binary vectors:

```
x = 0101010001

y = 0100011000
```

Answer:

Hamming distance = 3

Jaccard Similarity = $\frac{2}{5}$ = 0.4

Python realization of Hamming distance, Jaccard coefficient, Simple Matching Coefficient, cosine measure and correlation:

In [28]:

```
def hammingDistance(x, y):
    if len(x) != len(y):
        raise ValueError("Undefined for sequences of unequal length")
    return sum(el1 != el2 for el1, el2 in zip(x, y))

def jaccradDistance(x, y):
    if len(x) != len(y):
        raise ValueError("Undefined for sequences of unequal length")
    f11 = f00 = 0
    f = len(x)
    for i in range(0,f):
        if x[i] == y[i]:
            if x[i] == 0:
                f00 += 1
            elif x[i] == 1:
                f11 += 1
    return float(f11 / (f - f00))

def SMCDistance(x, y):
    if len(x) != len(y):
        raise ValueError("Undefined for sequences of unequal length")
    f_match = 0
    f = len(x)
    for i in range(0,f):
        if x[i] == y[i]:
                f_match += 1
    return float(f_match / f)

def COSDistance(x,y):
    if len(x) != len(y):
        raise ValueError("Undefined for sequences of unequal length")
    x = np.array(x)
    y = np.array(y)
    return (x @ np.transpose(y)) / (np.linalg.norm(x)*np.linalg.norm(y))

def pearsonCorr(x, y):
    if len(x) != len(y):
        raise ValueError("Undefined for sequences of unequal length")
    x = np.array(x); y = np.array(y)
    x_1 = x - np.mean(x)
    y_1 = y - np.mean(y)
    return round(np.sum(x_1*y_1)/np.sqrt(np.sum(x_1**2)*np.sum(y_1**2)),3)

def euclideanDistance(x, y):
    if len(x) != len(y):
        raise ValueError("Undefined for sequences of unequal length")
    x = np.array(x); y = np.array(y)
    d = np.sqrt(np.sum((x-y)**2))
    return d
```

executed in 20ms, finished 03:17:45 2023-01-31

In [29]:

```
x = [0, 1, 0, 1, 0, 1, 0, 0, 0, 1]
y = [0, 1, 0, 0, 0, 1, 1, 0, 0, 0]
```

executed in 4ms, finished 03:17:45 2023-01-31

Typesetting math: 100%

In [30]:

```
# Hamming distance
hammingDistance(x, y)
```

executed in 7ms, finished 03:17:46 2023-01-31

Out[30]:

```
3
```

In [31]:

```
# Jaccard coefficient
jaccradDistance(x, y)
```

executed in 6ms, finished 03:17:46 2023-01-31

Out[31]:

```
0.4
```

In [32]:

```
# Simple Matching Coefficient
SMCDistance(x,y)
```

executed in 5ms, finished 03:17:46 2023-01-31

Out[32]:

```
0.7
```

In [33]:

```
# cosine measure
COSDistance(x,y)
```

executed in 8ms, finished 03:17:47 2023-01-31

Out[33]:

```
0.5773502691896258
```

In [34]:

```
# Pearson's correlation
pearsonCorr(x,y)
```

executed in 6ms, finished 03:17:47 2023-01-31

Out[34]:

```
0.356
```

In [35]:

```
# Euclidean distance
euclideanDistance(x,y)
```

executed in 5ms, finished 03:17:47 2023-01-31

Out[35]:

```
1.7320508075688772
```

You can also import the python standard library `scipy` to calculate the similarity and distance measures:

- Hamming distance: scipy.spatial.distance.hamming (https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.hamming.html#scipy-spatial-distance-hamming)
- Simple Matching Coefficient: There is no direct way to calculate it in scipy. Do you know why?
- Jaccard measure: scipy.spatial.distance.jaccard (https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.jaccard.html)
- cosine measure: scipy.spatial.distance.cosine (https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cosine.html)
- Pearson's correlation: scipy.spatial.distance.correlation (https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.correlation.html)
- Euclidean distance: scipy.spatial.distance.euclidean (https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.euclidean.html)

The outputs may be a little different. You may read the documents of the functions for more details.

2.16.2 Q16(b)

Which approach, **Jaccard coefficient** or **Hamming distance**, is more similar to the **Simple Matching Coefficient**, and which approach is more similar to the **cosine measure**? Explain.
(Note: The Hamming measure is a distance, while the other three measures are similarities, but don't let this confuse you.)

Answer:

- The Hamming distance is more similar to the Simple Matching Coefficient. Because the Simple Matching Coefficient can be written as a funciton as the Hamming distance.

$$\text{Simple Matching Coefficient} = 1 - \frac{\text{Hamming distance}}{\text{number of bits}} \tag{2}$$

- The Jaccard measure is more similar to the cosine measure. Because both the Jaccard measure and the cosine measure ignore 0-0 matches.

Supplement material:

- Hamming distance = number of different bits
- Simple Matching Coefficient = $\frac{\text{number of matched bits}}{\text{number of bits}}$
- Jaccard measure = $\frac{f_{11}}{f_{01}+f_{10}+f_{11}}$
- cosine measure = $\frac{x \cdot y}{||x|| \cdot ||y||}$

2.16.3 Q16(c)

Suppose that you are **comparing how similar two organisms of different species** are in terms of the number of genes they share. Describe which measure, Hamming or Jaccard, you think would be more appropriate for comparing the genetic makeup of two organisms. Explain.
(Assume that each animal is represented as a binary vector, where each attribute is 1 if a particular gene is present in the organism and 0 otherwise.)

Answer:

(Section 2.4.10) Since we are comparing the **similarity** of two gene series from two different species, the attributes are asymmetric and the **0–0 matches should be ignored**. The similarity only depends on the number of characteristics they both share. The **Jaccard measure** is more appropriate for such data.

2.16.4 Q16(d)

If you wanted to **compare the genetic makeup of two organisms of the same species**, e.g., two human beings, would you use the Hamming distance, the Jaccard coefficient, or a different measure of similarity or distance? Explain.
(Note that two human beings share > 99.9% of the same genes.)

Answer:

Because >99.9% of the genes of human beings are the same, we should focus on their differences. The **dissimilarity** only depends on the number of characteristics they differ from. Thus, the **Hamming distance** is more appropriate in this situation.

2.17 Question 17

For the following vectors, x and y, calculate the indicated similarity or distance measures.

2.17.1 Q17(a)

Calculate cosine, correlation, Euclidean for x = (1, 1, 1, 1), y = (2, 2, 2, 2).

Answer:

In [36]:

```
x = [1,1,1,1]
y = [2,2,2,2]
```

executed in 4ms, finished 03:17:54 2023-01-31

In [37]:

```
# The Pearson correlation is undefined in this case, because the denominator is 0.
print('The cosine similarity is:', COSDistance(x,y),
      '\nThe Pearson correlation is undefined in this case:', pearsonCorr(x,y),
      '\nThe Euclidean distance is:', euclideanDistance(x,y))
```

executed in 9ms, finished 03:17:55 2023-01-31

```
The cosine similarity is: 1.0
The Pearson correlation is undefined in this case: nan
The Euclidean distance is: 2.0

/var/folders/j1/yv9cp32d6pq0cyk3_l8cfh8w0000gn/T/ipykernel_27390/3967830448.py:42: RuntimeWarning: invalid value
encountered in double_scalars
  return round(np.sum(x_1*y_1)/np.sqrt(np.sum(x_1**2)*np.sum(y_1**2)),3)
```

2.17.2 Q17(b)

Calculate the cosine, correlation, Euclidean, Jaccard for x = (0, 1, 0, 1), y = (1, 0, 1, 0).

Answer:

In [38]:

```
x = [0,1,0,1]
y = [1,0,1,0]
```

executed in 5ms, finished 03:17:57 2023-01-31

In [39]:

```
print('The cosine similarity is:', COSDistance(x,y),
      '\nThe Pearson correlation is:', pearsonCorr(x,y),
      '\nThe Euclidean distance is:', euclideanDistance(x,y),
      '\nThe Jaccard measure is:', jaccradDistance(x,y))
```

executed in 7ms, finished 03:17:57 2023-01-31

```
The cosine similarity is: 0.0
The Pearson correlation is: -1.0
The Euclidean distance is: 2.0
The Jaccard measure is: 0.0
```

2.17.3 Q17(c)

Calculate the cosine, correlation, Euclidean for x = (0, −1, 0, 1), y = (1, 0, −1, 0).

Answer:

In [40]:

```
x = [0,-1,0,1]
y = [1,0,-1,0]
```

executed in 5ms, finished 03:17:58 2023-01-31

In [41]:

```
print('The cosine similarity is:', COSDistance(x,y),
      '\nThe Pearson correlation is:', pearsonCorr(x,y),
      '\nThe Euclidean distance is:', euclideanDistance(x,y))
```

executed in 7ms, finished 03:17:58 2023-01-31

```
The cosine similarity is: 0.0
The Pearson correlation is: 0.0
The Euclidean distance is: 2.0
```

2.17.4 Q17(d)

Calculate the cosine, correlation, Jaccard for x = (1, 1, 0, 1, 0, 1), y = (1, 1, 1, 0, 0, 1) .

Answer:

In [42]:

```
x = [1,1,0,1,0,1]
y = [1,1,1,0,0,1]
```

executed in 5ms, finished 03:17:58 2023-01-31

In [43]:

```
print('The cosine similarity is:', COSDistance(x,y),
      '\nThe Pearson correlation is:', pearsonCorr(x,y),
      '\nThe Jaccard measure is:', jaccradDistance(x,y))
```

executed in 7ms, finished 03:17:59 2023-01-31

```
The cosine similarity is: 0.75
The Pearson correlation is: 0.25
The Jaccard measure is: 0.6
```

2.17.5 Q17(e)

Calculate the cosine, correlation for x = (2, −1, 0, 2, 0, −3), y = (−1, 1, −1, 0, 0, −1).

Answer:

In [44]:

```
x = [2,-1,0,2,0,-3]
y = [-1,1,-1,0,0,-1]
```

executed in 3ms, finished 03:17:59 2023-01-31

In [45]:

```
print('The cosine similarity is:', COSDistance(x,y),
      '\nThe Pearson correlation is:', pearsonCorr(x,y))
```

executed in 7ms, finished 03:18:00 2023-01-31

```
The cosine similarity is: 0.0
The Pearson correlation is: -0.0
```

2.18 Question 18

Here, we further explore the cosine and correlation measures.

2.18.1 Q18(a)

What is the range of values that are possible for the cosine measure?

Answer:

The range of cosine measure is [-1,1], because cosine measure = $\frac{x \cdot y}{||x|| \cdot ||y||} \in [-1, 1]$

If data has only positive entries, then $x \cdot y \geq 0$ and cosine measure = $\frac{x \cdot y}{||x|| \cdot ||y||} \in [0, 1]$

2.18.2 Q18(b)

If two objects have a cosine measure of 1, are they identical? Explain.

Answer:

They may be identical but not necessary. The two objects **differ by a constant factor**.

Counter example:

In [46]:

```
x = [1,1,1,1]
y = [2,2,2,2]
COSDistance(x,y)
```

executed in 9ms, finished 03:18:01 2023-01-31

Out[46]:

```
1.0
```

2.18.3 Q18(c)

What is the **relationship of the cosine measure to Pearson's correlation**, if any?
(Hint: Look at statistical measures such as mean and standard deviation in cases where cosine measure to Pearson's correlation are the same and different.)

Answer:

If the two vectors have **zero means**, then their cosine measure will be the same as their Pearson's correlation.

Proof:

The Pearson's correlation is

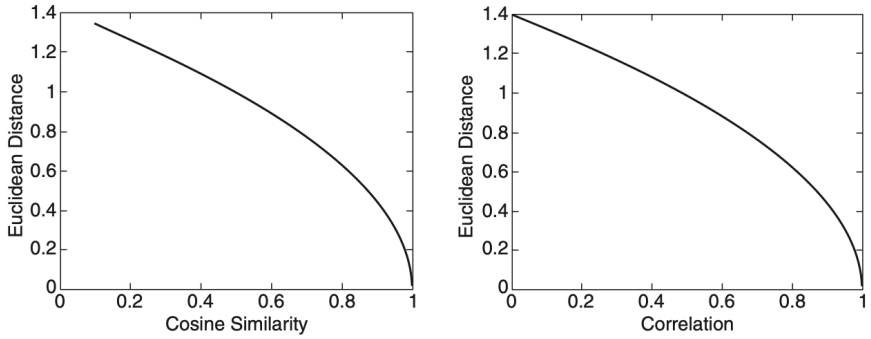
$$\rho(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}}) \left(\mathbf{y}_i - \bar{\mathbf{y}}\right)}{\sqrt{\sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})^2} \sqrt{\sum_{i=1}^{n} \left(\mathbf{y}_i - \bar{\mathbf{y}}\right)^2}}, \tag{3}$$

if $E(\mathbf{x}) = E(\mathbf{y}) = 0$, then

$$\begin{aligned} \rho(\mathbf{x}, \mathbf{y}) &= \frac{\sum_{i=1}^{n} \mathbf{x}_i \mathbf{y}_i}{\sqrt{\sum_{i=1}^{n} \mathbf{x}_i^2} \sqrt{\sum_{i=1}^{n} \mathbf{y}_i^2}} \\ &= \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}||_2 ||\mathbf{y}||_2} \\ &= cos(\mathbf{x}, \mathbf{y}) \end{aligned}$$

2.18.4 Q18(d)

Figure 2.1(a) shows the relationship of the cosine measure to Euclidean distance for 100,000 randomly generated points that have been normalized to have an ℓ_2 length of 1. What **general observation** can you make about the relationship between Euclidean distance and cosine similarity when vectors have an L2 norm of 1?



(a) Relationship between Euclidean distance and the cosine measure.

(b) Relationship between Euclidean distance and correlation.

Figure 2.1. Figures for exercise 20.

Answer:

- Firstly, since all the points fall on the same curve, we can safely claim that there is a functional relationship between Euclidean distance and cosine similarity for ℓ_2 normalized data.
- More specifically, there is an inverse relationship between cosine similarity and Euclidean distance. For example,
  - if two data points have a cosine similarity is one (two data points are identical), their Euclidean distance is zero,
  - if two data points have a high Euclidean distance, their cosine value is close to zero.

Proof: For ℓ_2-normalized vectors $\mathbf{x}$ and $\mathbf{y}$,

$$||\mathbf{x}||_2 = ||\mathbf{y}||_2 = 1,$$

we have the squared Euclidean distance is a function of the cosine distance,

$$\begin{aligned} \|\mathbf{x}-\mathbf{y}\|_2^2 &= (\mathbf{x}-\mathbf{y})^\top(\mathbf{x}-\mathbf{y}) \\ &= \mathbf{x}^\top\mathbf{x} - 2\mathbf{x}^\top\mathbf{y} + \mathbf{y}^\top\mathbf{y} \\ &= 2 - 2\mathbf{x}^\top\mathbf{y} \\ &= 2 - 2\cos(\mathbf{x},\mathbf{y}) \end{aligned}$$

2.18.5 Q18(e)

Figure2.1(b) shows the relationship of correlation to Euclidean distance for 100,000 randomly generated points that have been standardized to have a mean of 0 and a standard deviation of 1. What general observation can you make about the relationship between **Euclidean distance** and **Pearson's correlation** when the vectors have been standardized to have a mean of 0 and a standard deviation of 1?

Answer:

According to the Q18(c) and Q18(d), if the sample points are standardized to have a mean of 0 and a standard deviation of 1, then

- the vectors have zero means,
- the vectors have ℓ_2 norm = 1,

thus the cosine measure will be the same as their Pearson's correlation.
Therefore, the general observation we can make is the same as the answer in Q18(d).

2.18.6 Q18(f)

Derive the mathematical relationship between **cosine similarity** and **Euclidean distance** when each data object has an ℓ_2 length of 1.

Answer:

Based on the proof in Q18(d), we have

$$\|\mathbf{x}-\mathbf{y}\|_2 = \sqrt{2-2\cos(\mathbf{x},\mathbf{y})}.$$

2.18.7 Q18(g)

Derive the mathematical relationship between **correlation** and **Euclidean distance** when each data point has been been standardized by subtracting its mean and dividing by its standard deviation.

Answer:

The correlation between vectors $\mathbf{x}$ and $\mathbf{y}$ are defined as follows:

$$r(\mathbf{x},\mathbf{y}) = \frac{\frac{1}{n-1}\sum_i x_i y_i - \mu_x\mu_y}{s_x s_y} \tag{4}$$

where μ_x and μ_y are the means of $\mathbf{x}$ and $\mathbf{y}$ respectively, and s_x and s_y are the standard deviations of $\mathbf{x}$ and $\mathbf{y}$. Note that if $\mathbf{x}$ and $\mathbf{y}$ are standardized, they will each have a **mean of 0** and a **standard deviation of 1**, so the formula reduces to:

$$r(\mathbf{x}, \mathbf{y}) = \frac{1}{n-1} \sum_i x_i y_i \tag{5}$$

The Euclidean distance between vectors $\mathbf{x}$ and $\mathbf{y}$ are defined as follows:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i^n (x_i - y_i)^2} \tag{6}$$

$$= \sqrt{\sum_i x_i^2 + \sum_i y_i^2 - 2\sum_i x_i y_i} \tag{7}$$

Since $\mathbf{x}$ and $\mathbf{y}$ are standardized, the sums $\sum_i x_i^2$ and $\sum_i y_i^2$ are both equal to n-1. Thus, for standardized data, we can write the Euclidean distance as a function of the Pearson's correlation:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(n-1) + (n-1) - 2(n-1)corr(\mathbf{x}, \mathbf{y})} \tag{8}$$

$$= \sqrt{2(n-1)(1 - corr(\mathbf{x}, \mathbf{y}))} \tag{9}$$

2.19 Question 19

Show that the set difference metric given by

$$d(A, B) = size(A - B) + size(B - A)$$

satisfies the metric axioms given on page 77. A and B are sets and A – B is the set difference.

Supplement material:

The metric axioms:

- Positivity:
  - a. $d(\mathbf{x}, \mathbf{y}) \geq 0$ for all $\mathbf{x}$ and $\mathbf{y}$,
  - b. $d(\mathbf{x}, \mathbf{y}) = 0$ only if $\mathbf{x} = \mathbf{y}$.
- Symmetry: $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}$ and $\mathbf{y}$.
- Triangle inequality: $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ for all $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$.

Proof

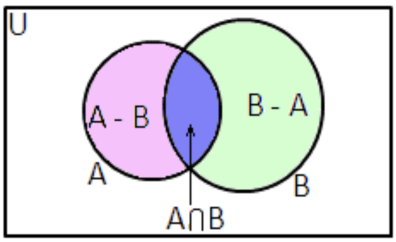
Given two sets A and B, the set difference is defined as:

$$A - B = \{x : x \in A \text{ and } x \notin B\}.$$

The size of a set (also called its cardinality) is the **number of elements** in the set.

- Positivity:
  - a. According to the definition, the size of a set is greater than or equal to 0, thus $d(A, B) = size(A - B) + size(B - A) \geq 0$
  - b. If $d(A, B) = 0$, then $size(A - B) = size(B - A) = 0$, then $A - B = B - A = \emptyset$, then $A \subset B$ and $B \subset A$, and thus, $A = B$.
- Symmetry: $d(A, B) = size(A - B) + size(B - A) = size(B - A) + size(A - B) = d(B, A)$.
- Triangle inequality: Since:

$$d(A, B) = size(A) + size(B) - 2size(A \cap B),$$



then

$$\begin{aligned} d(A, B) + d(B, C) &= size(A) + size(C) + 2size(B) - 2size(A \cap B) - 2size(B \cap C) \\ &= size(A) + size(C) - 2size(A \cap C) + 2size(B) - 2size(A \cap B) - 2size(B \cap C) + 2size(A \cap C) \\ &= d(A, C) + 2size(B) - 2size(A \cap B) - 2size(B \cap C) + 2size(A \cap C). \end{aligned}$$

Therefore, we only need to prove

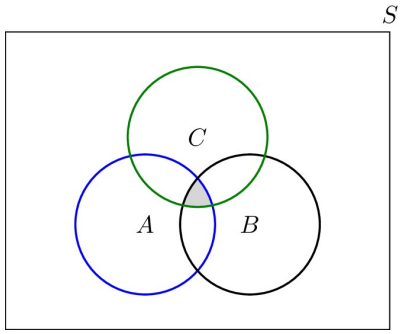
$$2size(B) - 2size(A \cap B) - 2size(B \cap C) + 2size(A \cap C) \geq 0$$

or equivalently,

$$size(B) \geq size(A \cap B) + size(B \cap C) - size(A \cap C).$$

Since

$$\begin{aligned} size(A \cap B) + size(B \cap C) - size(A \cap C) &= size((A \cap B) - C) + size(A \cap B \cap C) + size((B \cap C) - A) + size(B \cap C \cap A) - size((A \cap C) - B) - size(A \\ &= size((A \cap B) - C) + size(A \cap B \cap C) + size((B \cap C) - A) - size((A \cap C) - B) \\ &= size(B \cap (A \cup C)) - size((A \cap C) - B) \\ &\leq size(B \cap (A \cup C)) \\ &\leq size(B), \end{aligned}$$



then,

$$2size(B) - 2size(A \cap B) - 2size(B \cap C) + 2size(A \cap C) \geq 0,$$

thus,

$$d(A, C) \leq d(A, B) + d(B, C).$$

2.20 Question 20

Proximity is typically defined between a **pair** of objects.

2.20.1 Q20(a)

Define **two** ways in which you might define the **proximity among a group of objects**.

Answer:

The proximity is used to refer to either **similarity** or **dissimilarity (distance)**.

Two examples are the following:

- Based on pairwise proximity like cosine similarity, we can define the proximity among a group of objects by the minimum pairwise similarity or maximum pairwise dissimilarity among all the pairs. However, the drawback of this approach is that it is computationally expensive to compute all the pairwise proximities.
- If we compute a centroid (the mean of all the points) of the group in Euclidean space, then we can define the proximity among a group of objects by the sum or average of the distances of the points to the centroid. You may refer to the Section 7.2 for more details.

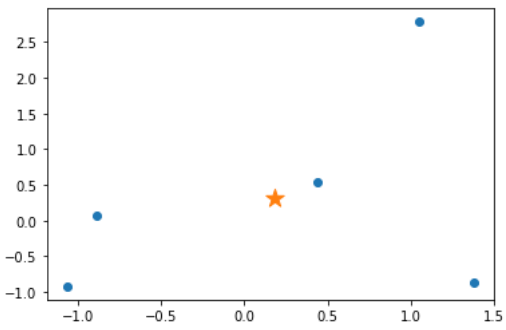
Python realization of finding the centroid:

In [47]:

```
import matplotlib.pyplot as plt  # standard way of importing matplotlib
%matplotlib inline

data = np.random.normal(0,1,size=(5,2))
centroid = np.mean(data,axis=0)
plt.scatter(data[:,0], data[:,1])
plt.scatter(centroid[0],centroid[1],marker='*',s=200)
plt.show()
```

executed in 78ms, finished 03:18:19 2023-01-31



*Here we randomly generate five points from the Multivariate Normal Distribution with $\mu = 0$ and $\Sigma = I$. The **centroid** is highlighted as a star.*

2.20.2 Q20(b)

How might you define the **distance** between two sets of points in Euclidean space?

Answer:

There are several approaches to define the distance between two sets of points in Euclidean space:

- the **distance between the centroids** of the two sets of points,
- the **maximum\minimum\median\average** pairwise distance between the two sets of points.

2.20.3 Q20(c)

How might you define the **proximity** between two sets of data objects? (Make no assumption about the data objects, except that a proximity measure is defined between any pair of objects.)

Answer:

You can define the proximity between two sets of data objects as:

- the **maximum\minimum\median\average** pairwise distance between the two sets of points.

*The definitions above are closely related to the concepts in the **agglomerative hierarchical clustering**, you may refer to the **section 8.2** for more details.*

2.21 Question 21

You are given **a set of points** S in Euclidean space, as well as the **distance of each point in** S **to a point** $\mathbf{x}$. (It does not matter if $\mathbf{x} \in S$.)

2.21.1 Q21(a)

If the goal is to **find all points within a specified distance** ε of point $\mathbf{y}$, $\mathbf{y} \neq \mathbf{x}$, explain how you could use the **triangle inequality** and the already calculated distances to $\mathbf{x}$ to potentially reduce the number of distance calculations necessary?
(Hint: If $\mathbf{z}$ is an arbitrary point of S, then the triangle inequality, $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{y}, \mathbf{z})$, can be rewritten as $d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{y}) - d(\mathbf{x}, \mathbf{z})$.)

Answer:

According to the triangle inequality

$$d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{y}, \mathbf{z}),$$

we have

$$d(\mathbf{x}, \mathbf{y}) - d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{y}, \mathbf{z}),$$

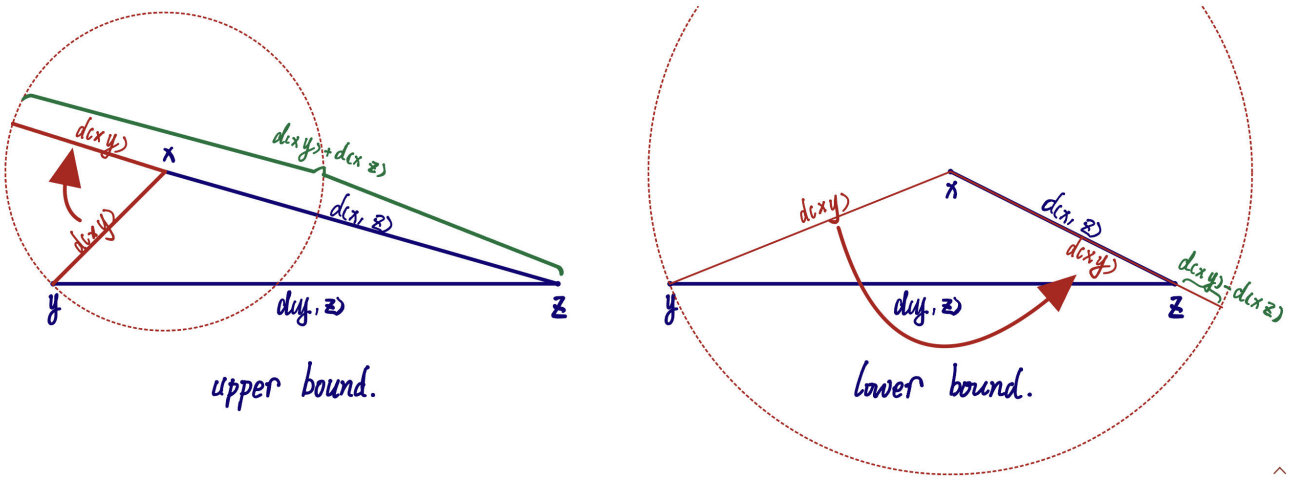
moreover, directly from the triangle inequality, we have

$$d(\mathbf{y}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{x}, \mathbf{z}).$$

Therefore, $d(\mathbf{y}, \mathbf{z})$ is bounded on both sides

$$d(\mathbf{x}, \mathbf{y}) - d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{y}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{x}, \mathbf{z}).$$

- If the upper bound of $d(\mathbf{y}, \mathbf{z})$ obtained from $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{x}, \mathbf{z})$ is less than or equal to ϵ, then $d(\mathbf{y}, \mathbf{z})$ does not need to be calculated. And the distance between point $\mathbf{z}$ and point $\mathbf{y}$ is less than or equal to ε.
- If the lower bound of $d(\mathbf{y}, \mathbf{z})$ obtained from $d(\mathbf{x}, \mathbf{y}) - d(\mathbf{x}, \mathbf{z})$ is larger than or equal to ϵ, then $d(\mathbf{y}, \mathbf{z})$ does not need to be calculated. And the distance between point $\mathbf{z}$ and point $\mathbf{y}$ is larger than or equal to ε.



2.21.2 Q21(b)

Based on Q21(a), how would the distance between x and y affect the number of distance calculations?

As the figure above shows,

- If x and y are very close (compare to other points), then the upper bound $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{x}, \mathbf{z})$ will close to $d(\mathbf{y}, \mathbf{z})$. Meanwhile, the lower bound will close to zero, since $d(\mathbf{x}, \mathbf{y}) \approx 0$, then less calculations are needed. In particular, no calculations are necessary when x = y.
- As x becomes farther away from y (compare to other points), both $d(\mathbf{x}, \mathbf{y})$ and $d(\mathbf{x}, \mathbf{z})$ will be very large. As a result, the lower bound $d(\mathbf{x}, \mathbf{y}) - d(\mathbf{x}, \mathbf{z}) \approx 0$ and the upper bound $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{x}, \mathbf{z})$ will be very large. More distance calculations are needed typically.

2.21.3 Q21(c)

Suppose that you can find **a small subset of points** S', from the original data set, such that **every point in the data set is within a specified distance** ε **of at least one of the points in** S', and that you also have the pairwise distance matrix for S'. Describe a technique that uses this information to compute, with a minimum of distance calculations, the set of all points within a distance of β of a specified point from the data set.

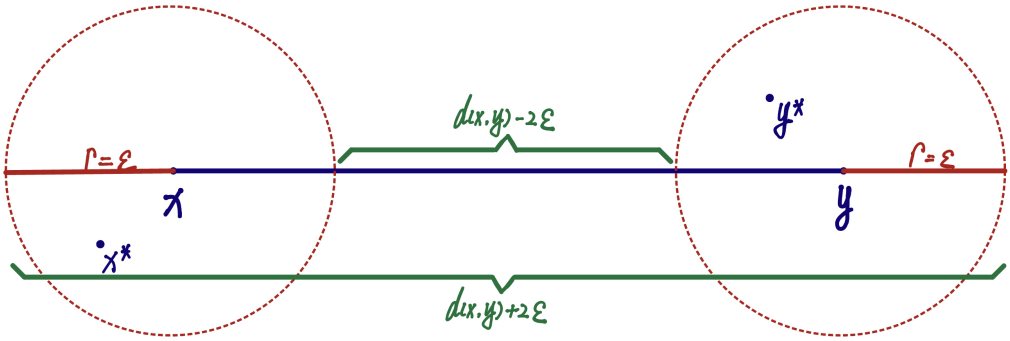
Answer:

The strategy with a minimum of distance calculations is as follows:

Let $\mathbf{x}$ and $\mathbf{y}$ be the two points and let $\mathbf{x}^*$ and $\mathbf{y}^*$ be the points in S' that are closest to the two points, respectively.

- If $d(\mathbf{x}^*, \mathbf{y}^*) + 2\epsilon \leq \beta$, then we can safely conclude $d(\mathbf{x}, \mathbf{y}) \leq \beta$.
- If $d(\mathbf{x}^*, \mathbf{y}^*) - 2\epsilon \geq \beta$, then we can safely conclude $d(\mathbf{x}, \mathbf{y}) \geq \beta$.

These formulas are derived by considering the cases where $\mathbf{x}$ and $\mathbf{y}$ are as far from $\mathbf{x}^*$ and $\mathbf{x}^*$ as possible and as far or close to each other as possible.



2.22 Question 22

Show that 1 minus the Jaccard similarity is a distance measure between two data objects, x and y, that satisfies the metric axioms given on page 77. Specifically:

$$d(x, y) = 1 - J(x, y). \tag{10}$$

Answer:

The Jaccard measure is defined as $J = \frac{f_{11}}{f_{01}+f_{10}+f_{11}}$

- Positivity:
  - a. Because $J(\mathbf{x}, \mathbf{y}) \leq 1$, $d(\mathbf{x}, \mathbf{y}) \geq 0$.
  - b. If $d(\mathbf{x}, \mathbf{y}) = 0$, then $J(\mathbf{x}, \mathbf{y}) = 1$, then $\frac{f_{11}}{f_{01}+f_{10}+f_{11}} = 1$, then $f_{01} = f_{10} = 0$, then $\mathbf{x} = \mathbf{y}$.
- Symmetry: Because $J(\mathbf{x}, \mathbf{y}) = J(\mathbf{y}, \mathbf{x})$, then $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$.
- Triangle inequality:

(Proof due to Jeffrey Ullman)
$minhash(\mathbf{x})$ is the index of first nonzero entry of $\mathbf{x}$
$prob(minhash(\mathbf{x}) = k)$ is the probability that $minhash(\mathbf{x})$ = k when x is randomly permuted.
Note that $prob(minhash(\mathbf{x}) = minhash(\mathbf{y})) = J(\mathbf{x}, \mathbf{y})$ (minhash lemma) Therefore,
$d(\mathbf{x}, \mathbf{y}) = 1 - prob(minhash(\mathbf{x}) = minhash(\mathbf{y})) = prob(minhash(x) \neq minhash(y))$
We have to show that,
$prob(minhash(\mathbf{x}) \neq minhash(\mathbf{z})) \leq prob(minhash(\mathbf{x}) \neq minhash(\mathbf{y})) + prob(minhash(\mathbf{y}) \neq minhash(\mathbf{z})$
However, note that whenever $minhash(\mathbf{x}) \neq minhash(\mathbf{z})$, then at least one of $minhash(\mathbf{x}) \neq minhash(\mathbf{y}) and minhash(\mathbf{y}) \neq minhash(\mathbf{z})$ must be true.

2.23 Question 23

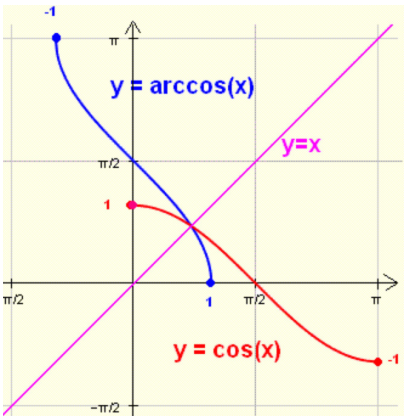
Show that the distance measure defined as the angle between two data vectors with ℓ_2 norm =1, $\mathbf{x}$ and $\mathbf{y}$, satisfies the metric axioms given on page 77. Specifically:

$$d(\mathbf{x}, \mathbf{y}) = arccos(cos(\mathbf{x}, \mathbf{y})), \tag{11}$$

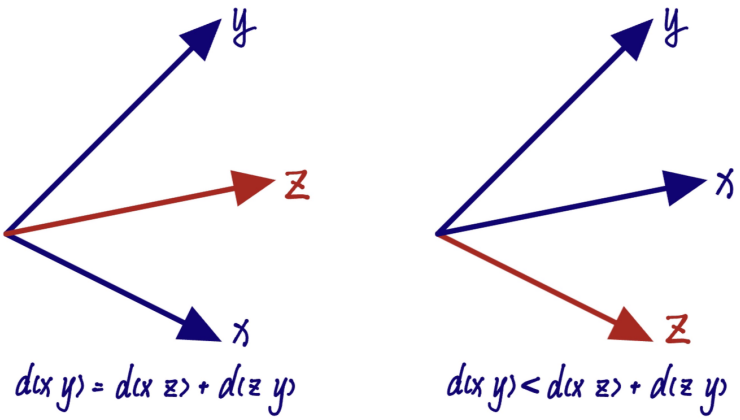
where $||\mathbf{x}||_2 = ||\mathbf{y}||_2 = 1$.

Answer:

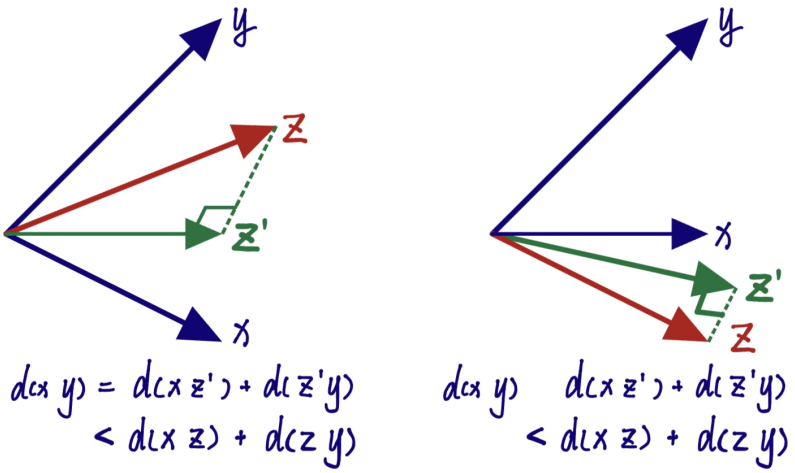
The angles between two vectors $\mathbf{x}$ and $\mathbf{y}$ are in the range $0°$ to $180°$.



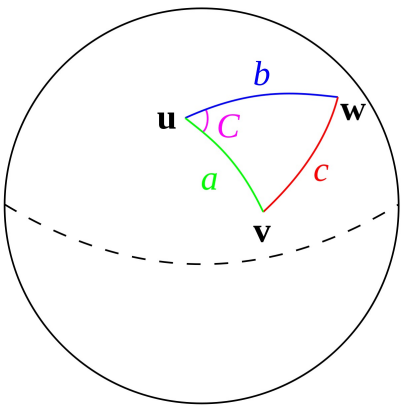
- Positivity:
  - Because $-1 \leq cos(\mathbf{x}, \mathbf{y}) \leq 1$, then $0 \leq arccos(\mathbf{x}, \mathbf{y}) \leq \pi$.
  - If $d(\mathbf{x}, \mathbf{y}) = 0$, then $cos(\mathbf{x}, \mathbf{y}) = 1$, then $\mathbf{x} = \mathbf{y}$ because $||\mathbf{x}||_2 = ||\mathbf{y}||_2 = 1$.
- Symmetry: Because $cos(\mathbf{x}, \mathbf{y}) = cos(\mathbf{y}, \mathbf{x})$, $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$.
- Triangle inequality:
  - If the three vectors lie in the same plane then it is obvious that the angle between $\mathbf{x}$ and $\mathbf{y}$ must be less than or equal to the sum of the angles between $\mathbf{x}$ and $\mathbf{z}$ and $\mathbf{z}$ and $\mathbf{y}$.



  - If $\mathbf{z}'$ is the projection of $\mathbf{z}$ into the plane defined by $\mathbf{x}$ and $\mathbf{y}$, then note that the angles between $\mathbf{x}$ and $\mathbf{z}$ and $\mathbf{z}$ and $\mathbf{y}$ are greater than those between between $\mathbf{x}$ and $\mathbf{z}'$ and $\mathbf{z}'$ and $\mathbf{y}$.



  - An insightful proof: The angle is to the sphere what the distance is to the plane, so we can prove the triangle inequality in the sphere. Without loss of generality, let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{R}^3$. Then we can find points $\mathbf{x}, \mathbf{y}, \mathbf{z}$ on the unit sphere. The triangle inequality holds for minor arcs on a sphere, and the arc length is equal to the angle, so the required result holds.



Here (https://math.stackexchange.com/questions/1924742/prove-the-triangle-inequality-on-the-sphere-s2-in-mathbbr3/1925049) is a proof of the triangle inequality on spherical surfaces.

2.24 Question 24

Explain why computing the **proximity between two attributes** is often simpler than computing the **similarity between two objects**.

Answer:

In general, an object is corresponding to a record with different types of attributes.

- Since the values of an attribute are all of the same type, and thus, if another attribute is of the same type, then the **proximity between two attributes** is conceptually and computationally straightforward.

- However, to compute the **similarity between two objects**, we need to decide how to compute and combine the similarities for heterogeneous attributes. This can be done by using Equations 2.15 or 2.16 in the textbook, but is still somewhat ad hoc, at least compared to proximity measures such as the Euclidean distance or correlation, which are mathematically wellfounded.