Machine Learning with Python

Session 11: Measures of Proximity, Components of Machine Learning

Arghya Ray

Introduction:

The term proximity between two objects is a function of the proximity between the corresponding attributes of the two objects. Proximity measures refer to the **Measures of Similarity and Dissimilarity**.

Similarity and Dissimilarity are important because they are used by a number of data mining techniques, such as clustering, nearest neighbour classification, and anomaly detection.

What is Similarity?

→ It is a numerical measure of the degree to which the two objects are alike.

→ Higher for pair of objects that are more alike.

→ Usually non-negative and between 0 & 1.

0 ~ No Similarity, 1 ~ Complete Similarity

What is Dissimilarity?

→ It is a numerical measure of the degree to which the two objects are different.

→ Lower for pair of objects that are more similar.

→ Range 0 to infinity.

Transformation Function

It is a function used to convert similarity to dissimilarity and vice versa, or to transform a proximity measure to fall into a particular range. For instance:

s’ = (s-min(s)) / max(s)-min(s))

range

where,

s’ = new transformed proximity measure value,

s = current proximity measure value,

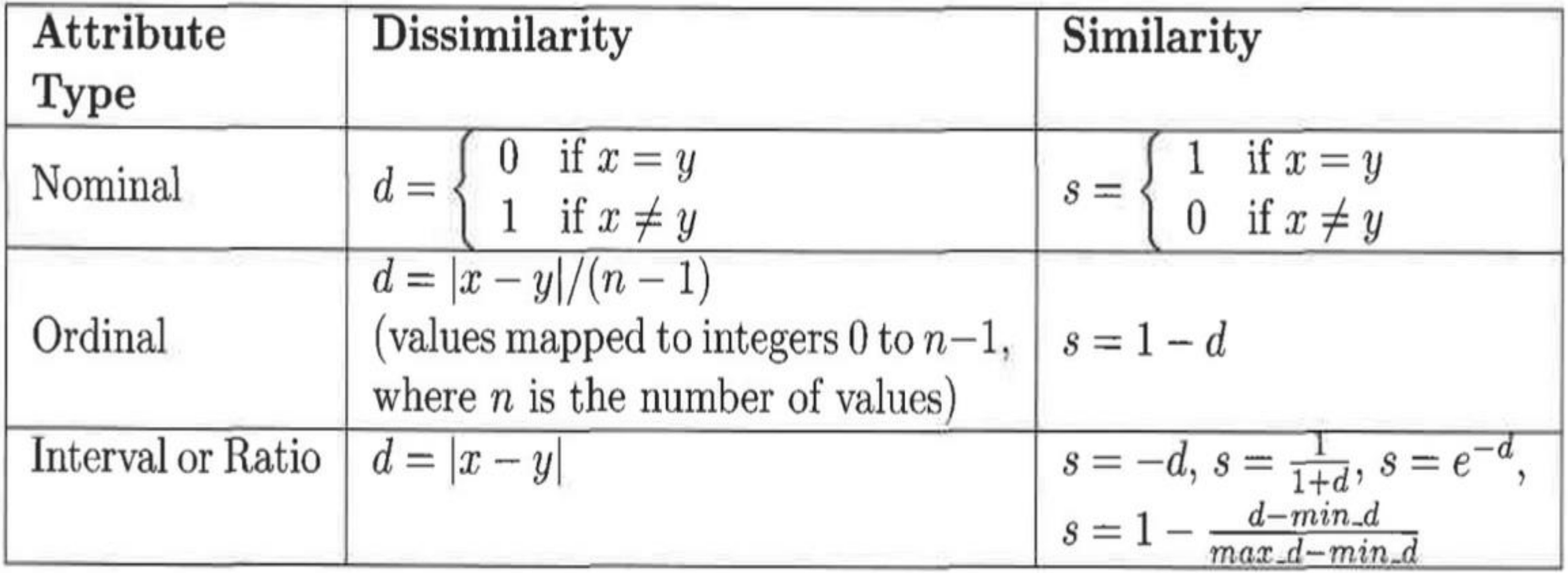
min(s) = minimum of proximity measure values,

max(s) = maximum of proximity measure values

This transformation function is just one example from all the available options out there.

Similarity and Dissimilarity between Simple Attributes

The proximity of objects with a number of attributes is usually defined by combining the proximities of individual attributes, so, we first discuss proximity between objects having a single attribute.



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

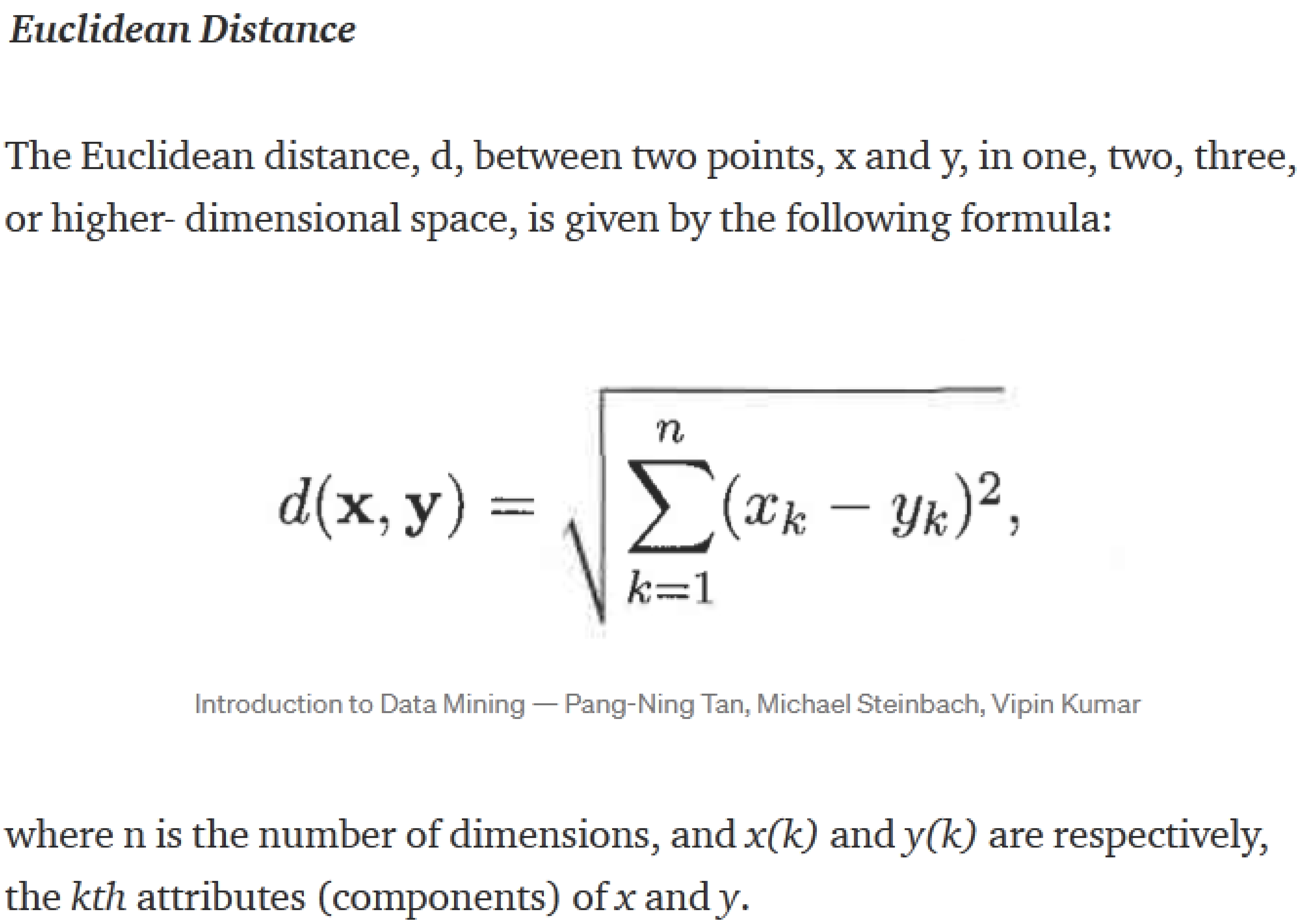
To understand it better, let us go through some examples.

Consider objects described by one **nominal** attribute. How to compare similarity of two objects like this? Nominal attributes only tell us about the distinctness of objects. Hence, in this case similarity is defined as 1 if attribute values match, and 0 otherwise and oppositely defined would be dissimilarity.

For objects with a single **ordinal** attribute, the situation is more complicated because information about order needs to be taken into account. Consider an attribute that measures the quality of a product, on the scale {poor, fair, OK, good, wonderful}. We have 3 products P1, P2, & P3 with quality as wonderful, good, & OK respectively. In order to compare **ordinal** quantities, they are mapped to successive integers. In this case, if the scale is mapped to {0, 1, 2, 3, 4} respectively. Then, dissimilarity(P1, P2) = 4−3 = 1.

For **interval or ratio** attributes, the natural measure of dissimilarity between two objects is the absolute difference of their values. For example, we might compare our current weight and our weight a year ago by saying “I am ten pounds heavier.”

Dissimilarities between Data Objects



Euclidean Distance

The Euclidean distance, d, between two points, x and y, in one, two, three, or higher- dimensional space, is given by the following formula:

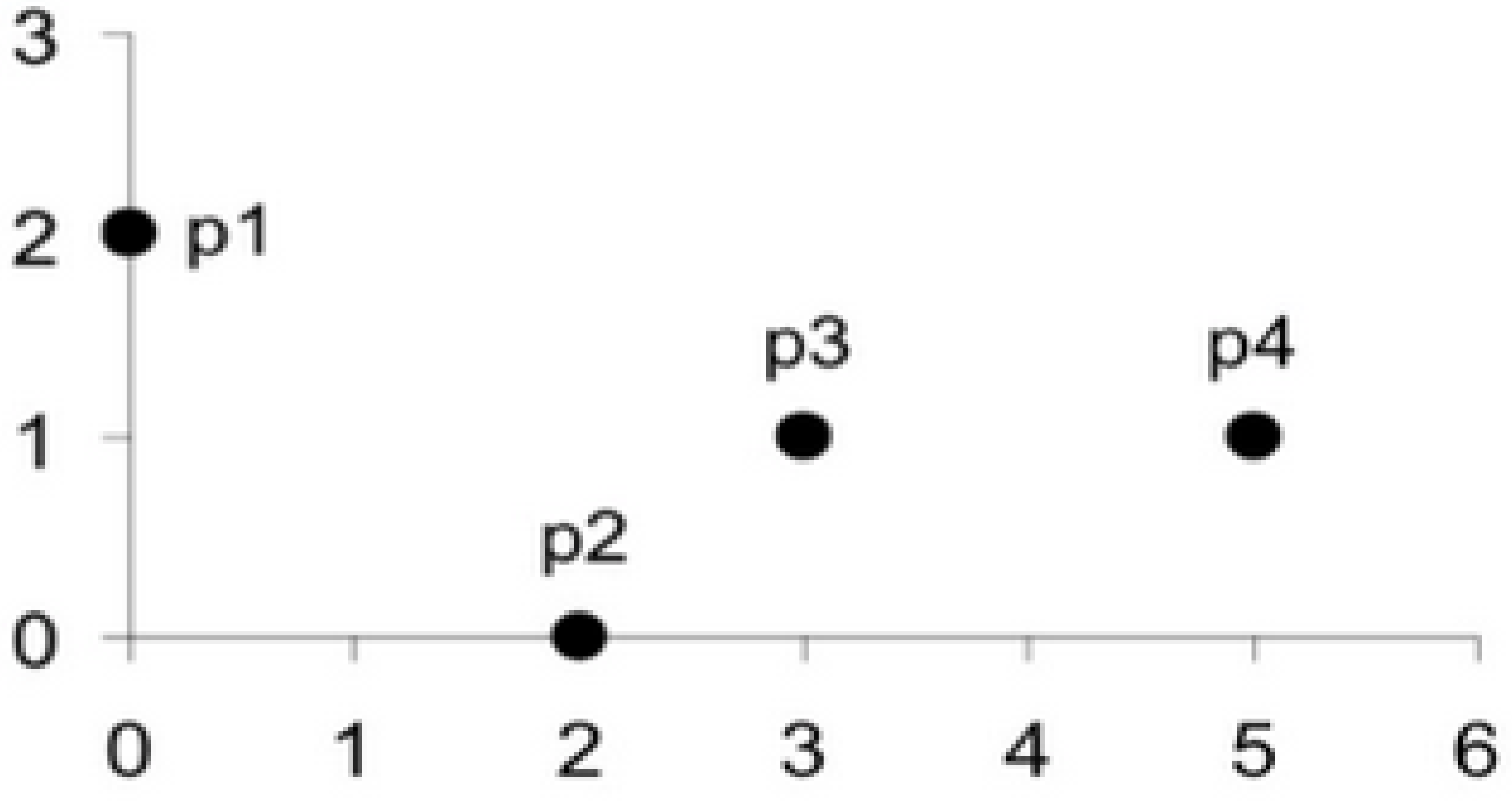
$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2},$$

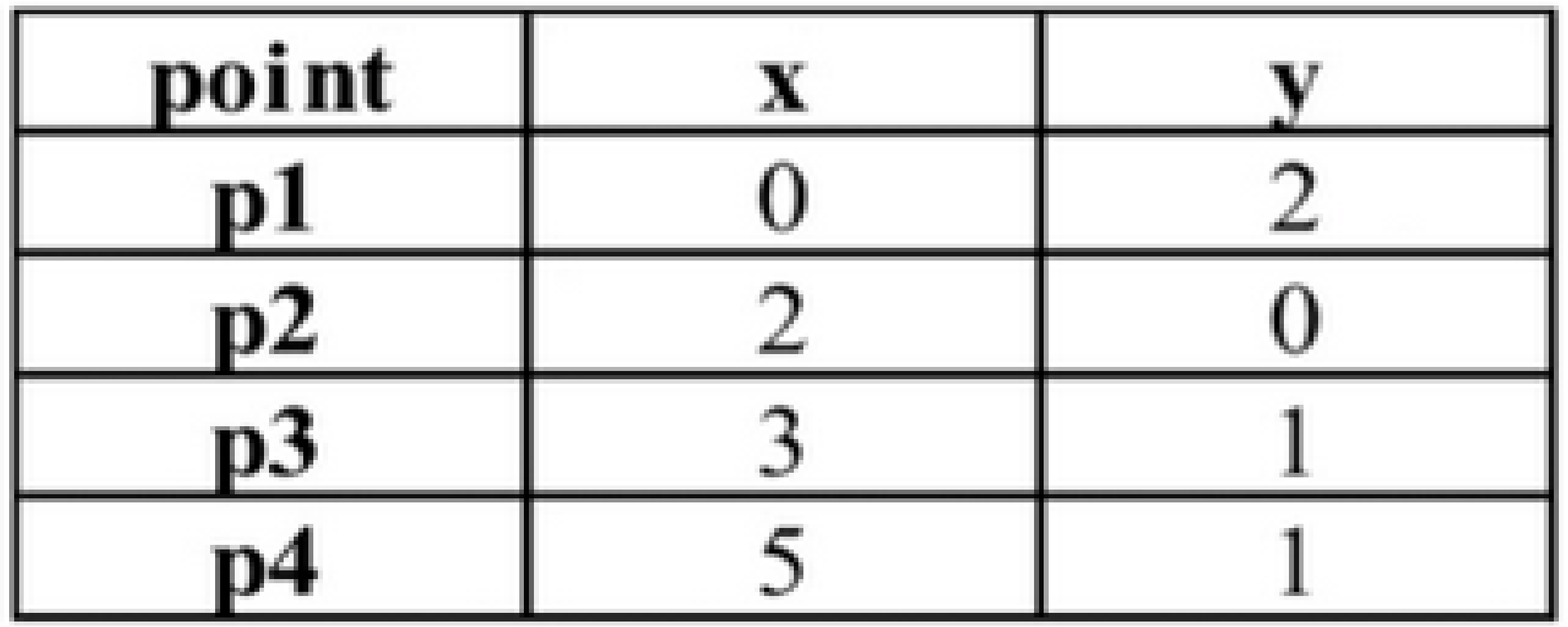
Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

where n is the number of dimensions, and *x(k)* and *y(k)* are respectively, the *kth* attributes (components) of *x* and *y*.

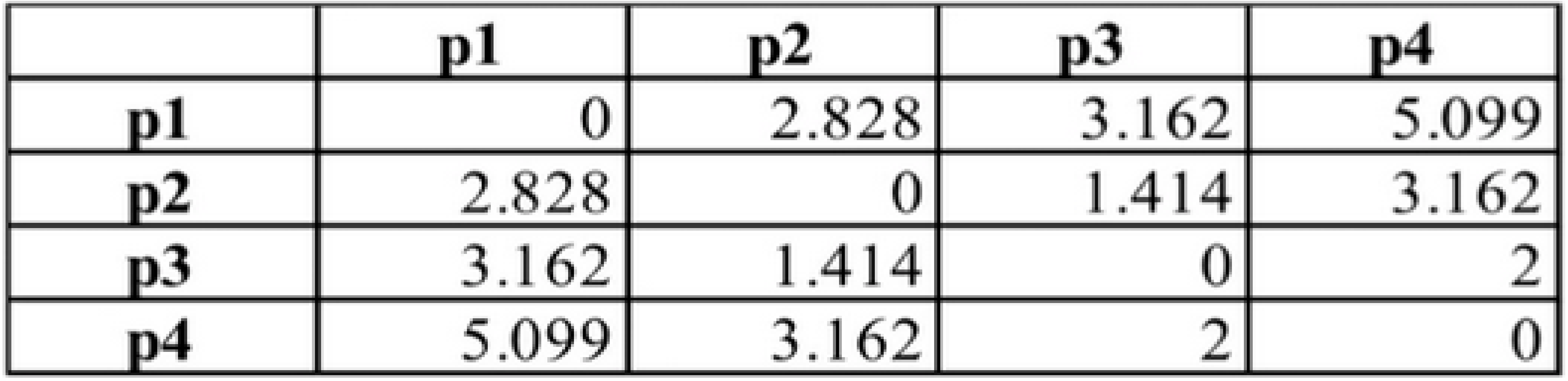
Dissimilarities between Data Objects

Example:





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

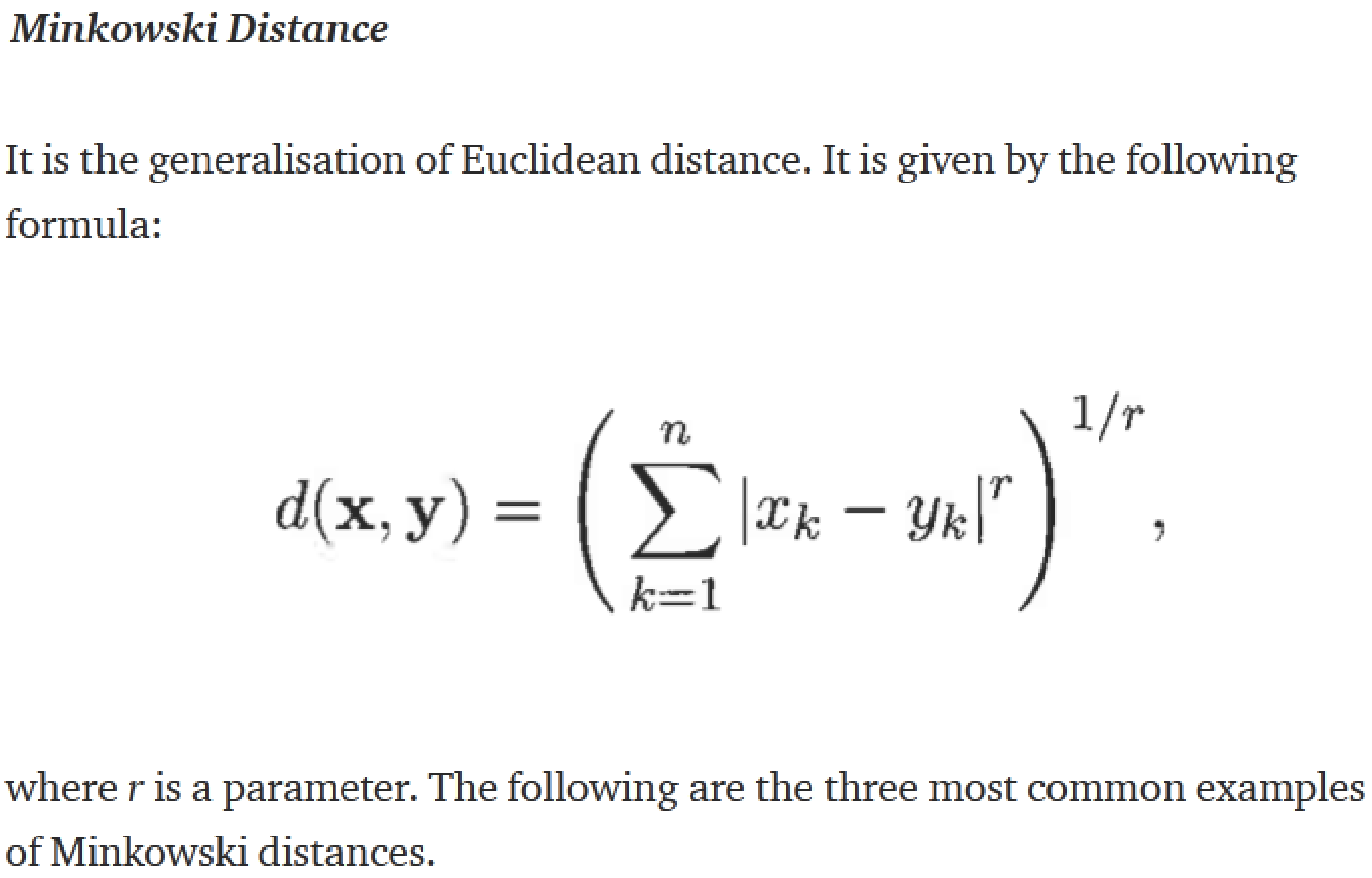


	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

Dissimilarities between Data Objects



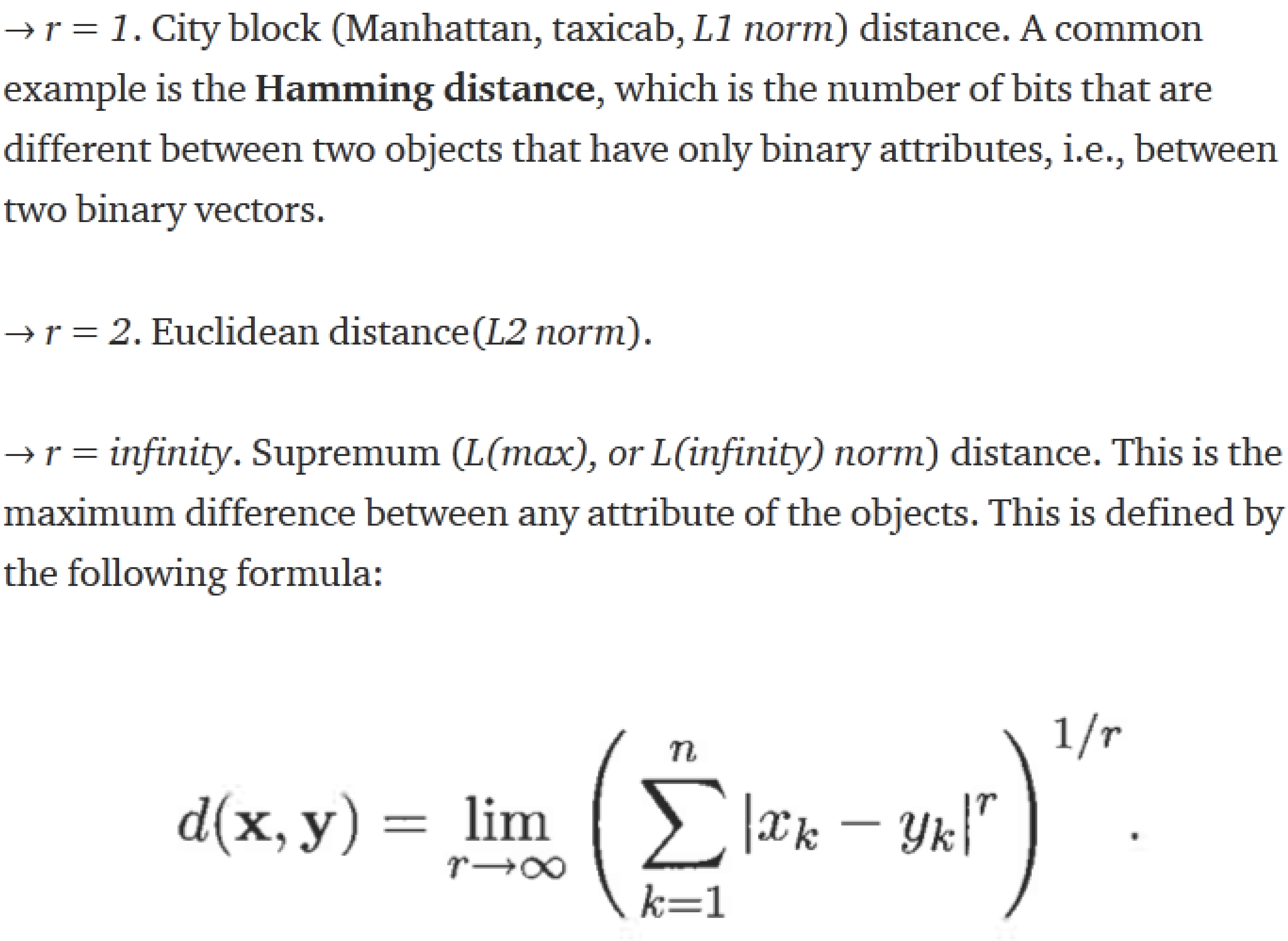
Minkowski Distance

It is the generalisation of Euclidean distance. It is given by the following formula:

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r},$$

where *r* is a parameter. The following are the three most common examples of Minkowski distances.

Dissimilarities between Data Objects



→ *r* = *1*. City block (Manhattan, taxicab, *L1 norm*) distance. A common example is the **Hamming distance**, which is the number of bits that are different between two objects that have only binary attributes, i.e., between two binary vectors.

→ *r* = *2*. Euclidean distance(*L2 norm*).

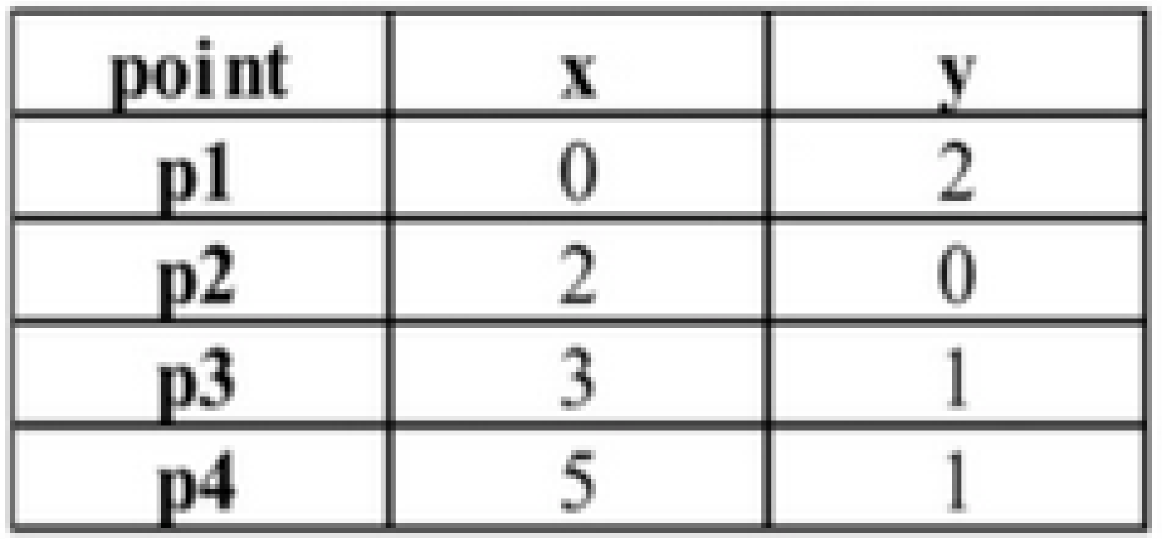
→ *r* = *infinity*. Supremum (*L(max), or L(infinity) norm*) distance. This is the maximum difference between any attribute of the objects. This is defined by the following formula:

$$d(\mathbf{x},\mathbf{y}) = \lim_{r\to\infty}\left(\sum_{k=1}^{n}|x_k-y_k|^r\right)^{1/r}.$$

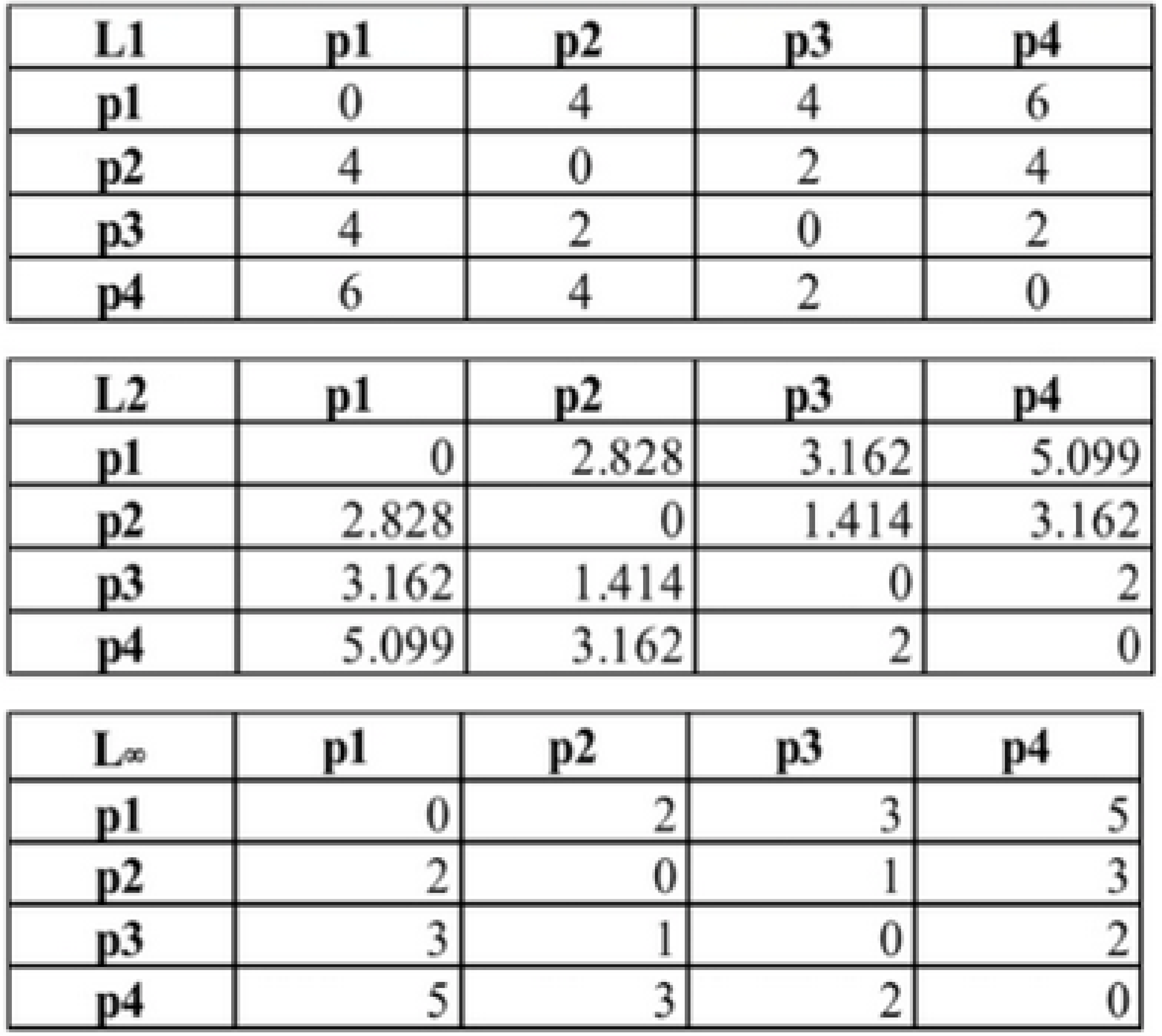
```
1 11 0001
1 1 01110
-----------------------
0011111  = 5

Barun
Beran
------------
0+1+0+1+0  = 2
```

Dissimilarities between Data Objects

Example:



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

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

Distances, such as the Euclidean distance, have some well-known properties. If $d(x, y)$ is the distance between two points, x and y, then the following properties hold.

Positivity

a) $d(x, y) > 0$ for all x and y,

b) $d(x, y) = 0$ only if $x = y$

2. **Symmetry**

$d(x, y) = d(y, x)$ for all x and y

3. **Triangle Inequality**

$d(x, z) \leq d(x, y) + d(y, z)$ for all points x, y and z

The measures that satisfy all three properties are called **metrics.**

Similarities between Data Objects

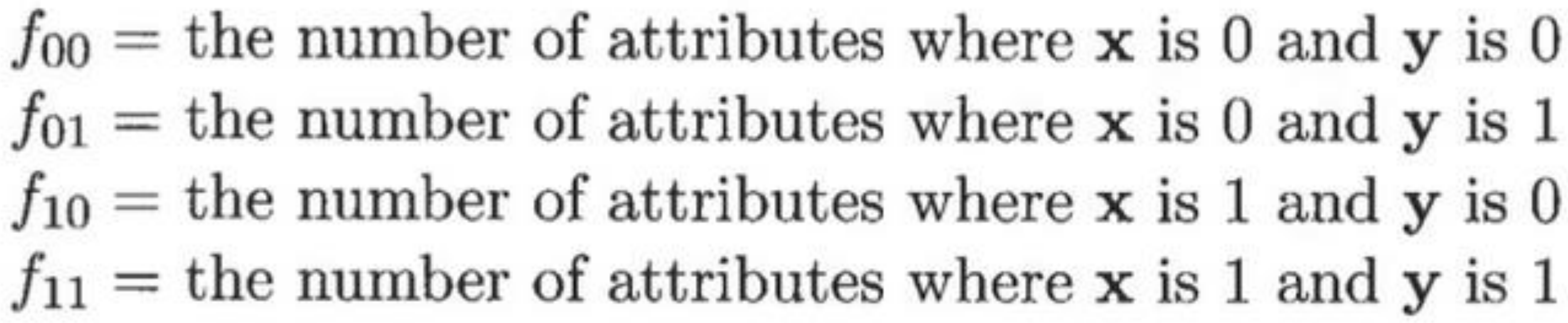
For similarities, the triangle inequality typically does not hold, but symmetry and positivity typically do. To be explicit, if *s(x, y)* is the similarity between points *x* and *y*, then the typical properties of similarities are the following:

s(x, y) = *1* only if *x* = *y*. (0 ≤ s ≤ 1)

s(x, y) = *s(y, x)* for all *x* and *y*. (Symmetry)

There is no general analog of the triangle inequality for similarity measure.

Similarity Measures for Binary Data are called **similarity coefficients** and typically have values between 0 and 1. The comparison between two binary objects is done using the following four quantities:



f_{00} = the number of attributes where $\mathbf{x}$ is 0 and $\mathbf{y}$ is 0

f_{01} = the number of attributes where $\mathbf{x}$ is 0 and $\mathbf{y}$ is 1

f_{10} = the number of attributes where $\mathbf{x}$ is 1 and $\mathbf{y}$ is 0

f_{11} = the number of attributes where $\mathbf{x}$ is 1 and $\mathbf{y}$ is 1

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

Simple Matching Coefficient

It is defined as follows:

$$SMC = \frac{\text{number of matching attribute values}}{\text{number of attributes}} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}.$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

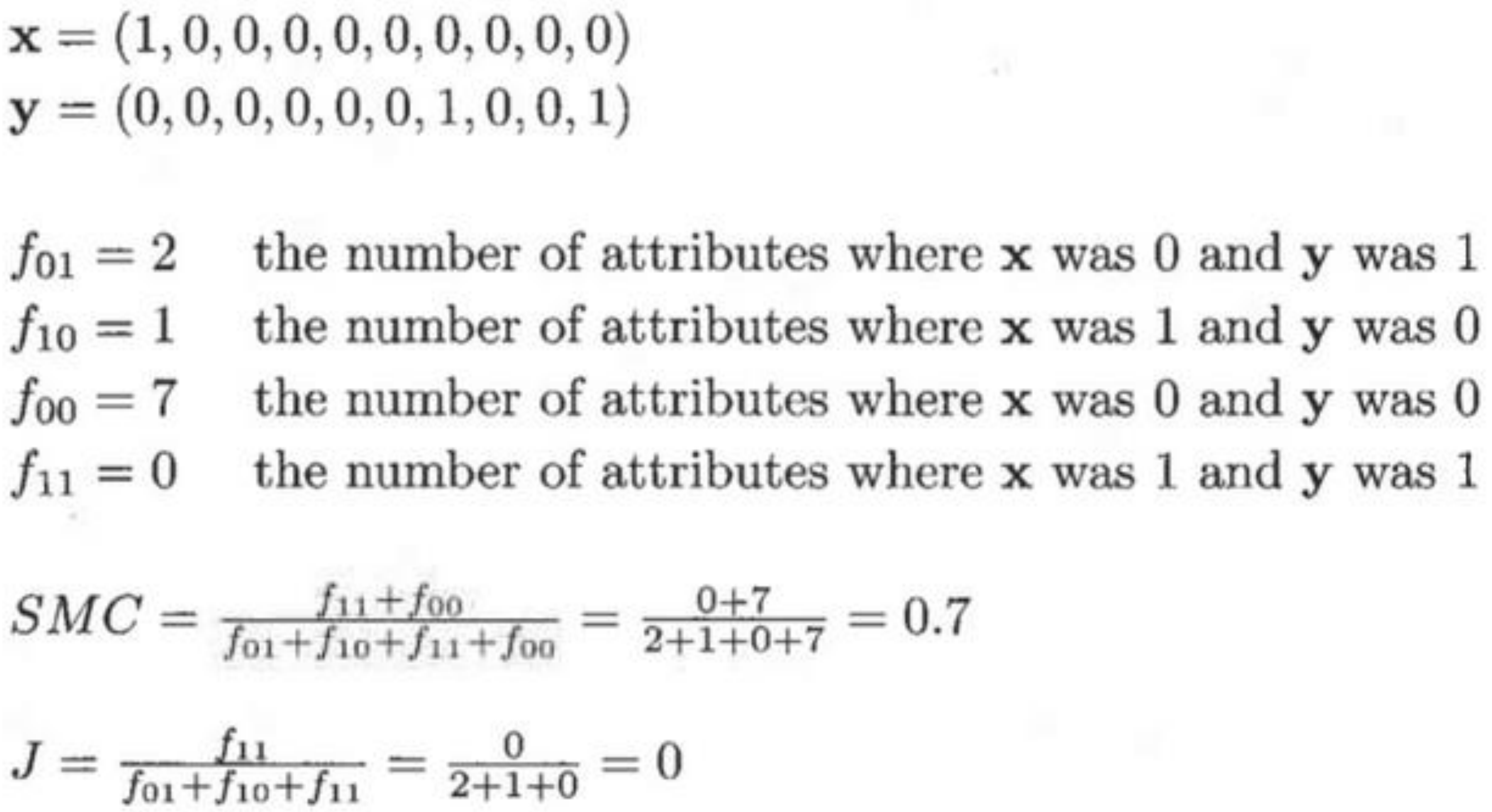
Jaccard Coefficient

It is defined as follows:

$$J = \frac{\text{number of matching presences}}{\text{number of attributes not involved in 00 matches}} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}.$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

An example comparing these two similarity methods:



$$\mathbf{x} = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$
$$\mathbf{y} = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)$$

$f_{01} = 2$ the number of attributes where $\mathbf{x}$ was 0 and $\mathbf{y}$ was 1

$f_{10} = 1$ the number of attributes where $\mathbf{x}$ was 1 and $\mathbf{y}$ was 0

$f_{00} = 7$ the number of attributes where $\mathbf{x}$ was 0 and $\mathbf{y}$ was 0

$f_{11} = 0$ the number of attributes where $\mathbf{x}$ was 1 and $\mathbf{y}$ was 1

$$SMC = \frac{f_{11}+f_{00}}{f_{01}+f_{10}+f_{11}+f_{00}} = \frac{0+7}{2+1+0+7} = 0.7$$

$$J = \frac{f_{11}}{f_{01}+f_{10}+f_{11}} = \frac{0}{2+1+0} = 0$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

Q. Using the given snapshot of a movie recommender system, find the Jaccard's coefficient and Matching coefficient between (a) User 1 and User 3 (b) User 2 and User 3.

	User 1	User 2	User 3
There	1	0	0
Gravity	0	1	1
X-Men	0	0	1
Inception	0	1	1
Jurassic Park	1	0	0
Avengers: End Game	1	1	1

Between User 1 & User 3:

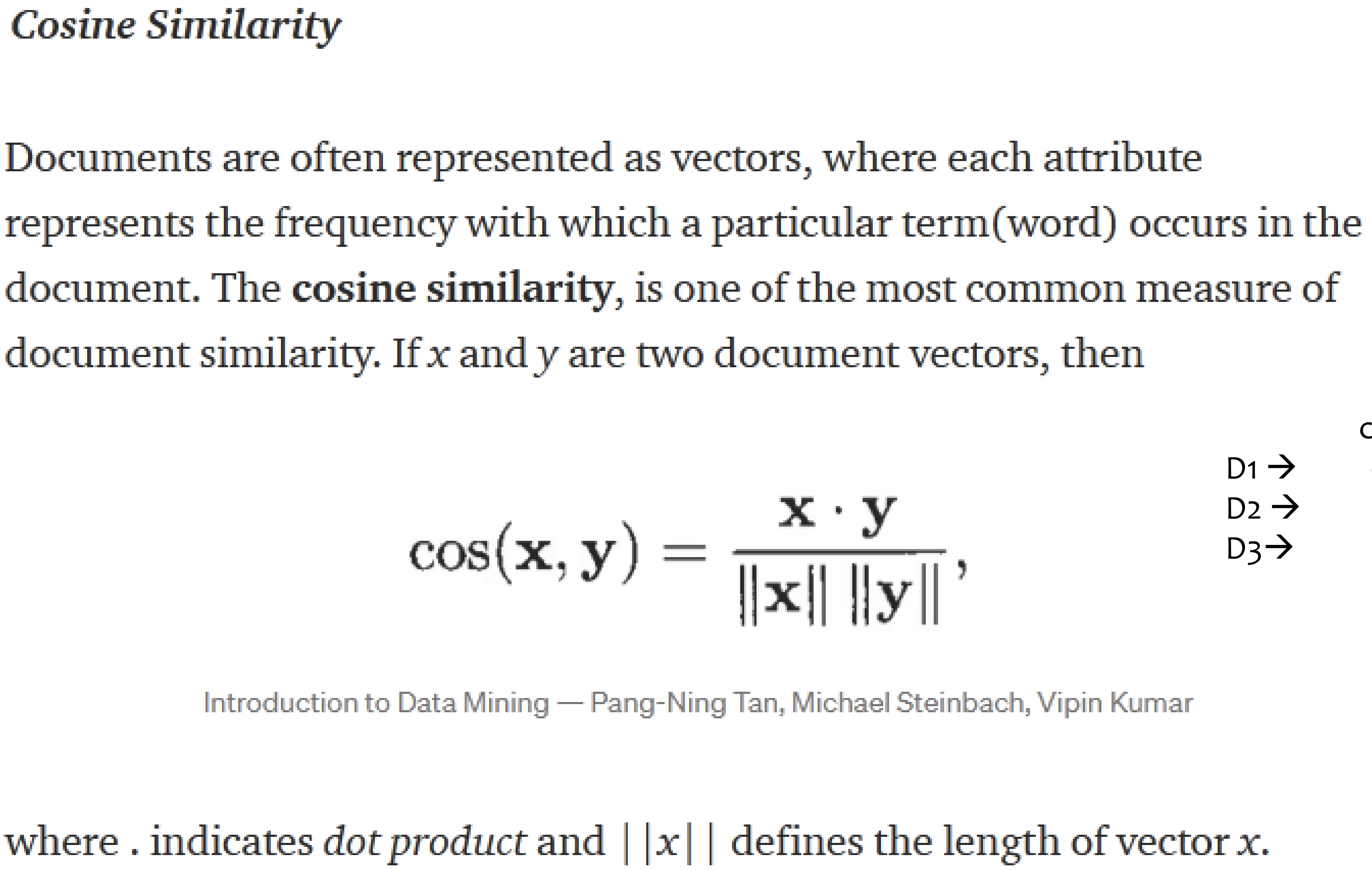
Jaccard's Coefficient= 1/6

Matching coefficient = 1/6

Between User 2 and User 3:

Jaccard's Coefficient= 3/4

Matching coefficient = 5/6



Cosine Similarity

Documents are often represented as vectors, where each attribute represents the frequency with which a particular term(word) occurs in the document. The **cosine similarity**, is one of the most common measure of document similarity. If *x* and *y* are two document vectors, then

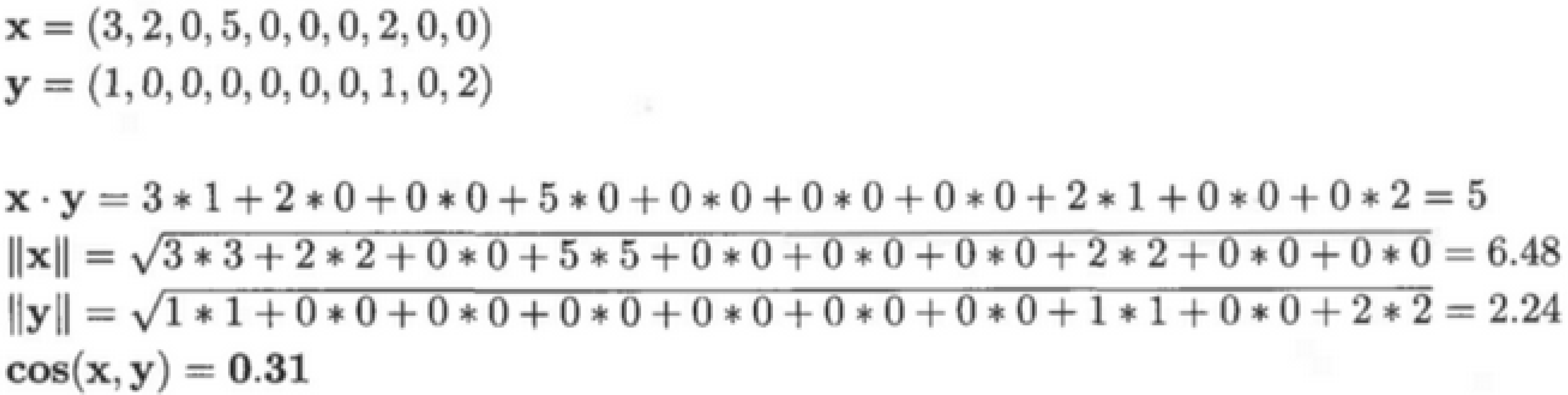
$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|},$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

where . indicates *dot product* and $||x||$ defines the length of vector *x*.

	cos	sin	tan
D1 →	0	1	2
D2 →	1	2	3
D3 →	2	1	1

An example of **cosine similarity** measure is as follows:

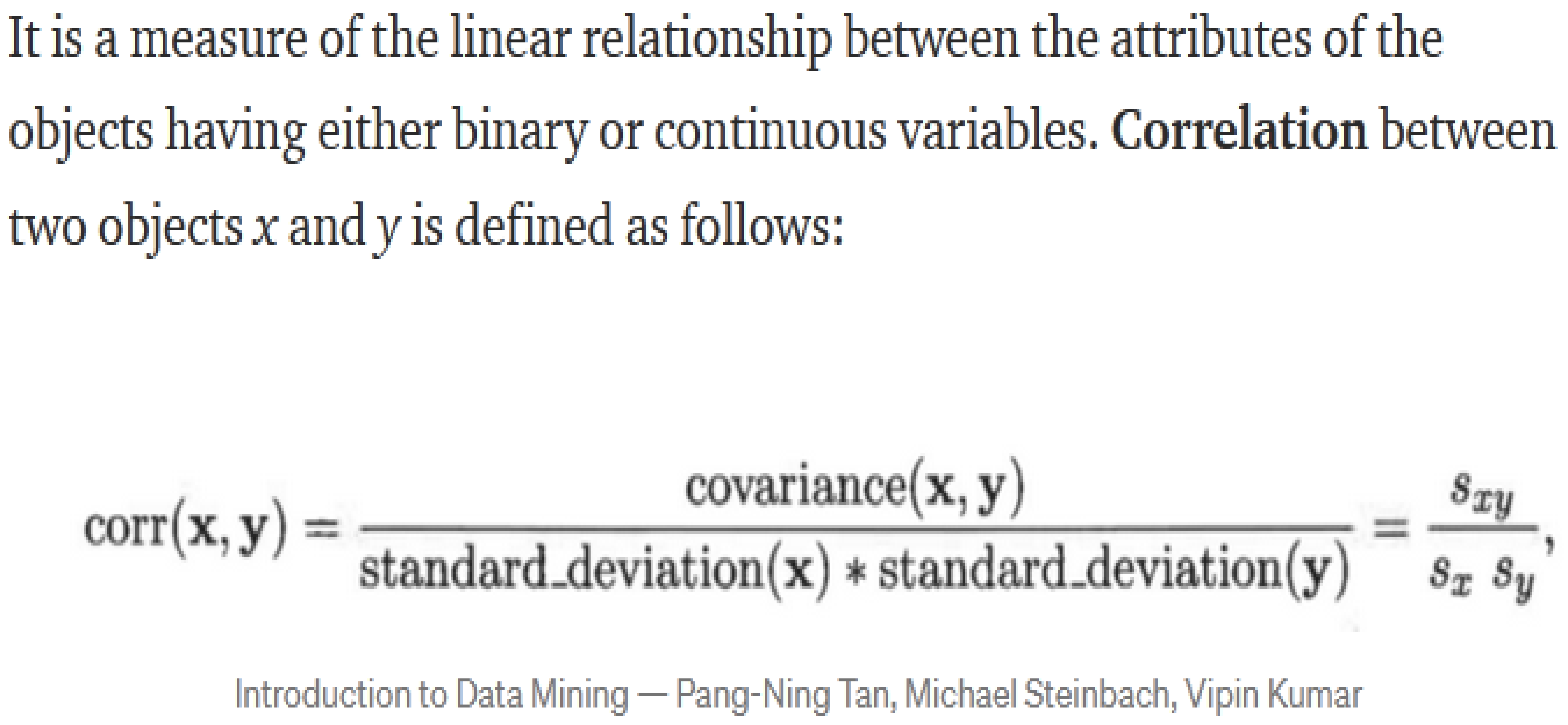


$$\mathbf{x} = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$$
$$\mathbf{y} = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$$

$$\mathbf{x} \cdot \mathbf{y} = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$$
$$\|\mathbf{x}\| = \sqrt{3*3 + 2*2 + 0*0 + 5*5 + 0*0 + 0*0 + 0*0 + 2*2 + 0*0 + 0*0} = 6.48$$
$$\|\mathbf{y}\| = \sqrt{1*1 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 1*1 + 0*0 + 2*2} = 2.24$$
$$\mathbf{cos(x, y) = 0.31}$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

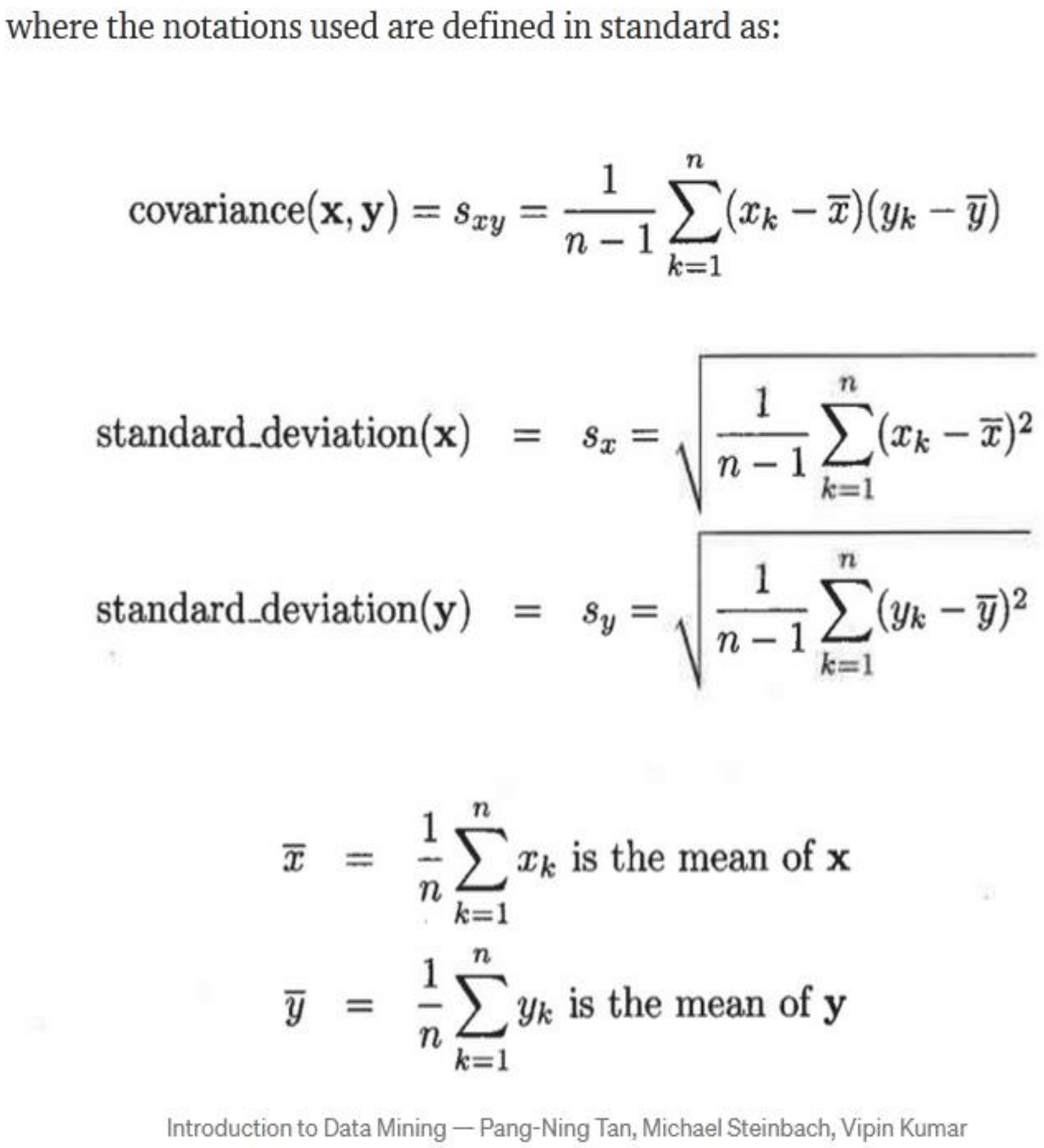
Correlation



It is a measure of the linear relationship between the attributes of the objects having either binary or continuous variables. **Correlation** between two objects *x* and *y* is defined as follows:

$$\mathrm{corr}(\mathbf{x},\mathbf{y}) = \frac{\mathrm{covariance}(\mathbf{x},\mathbf{y})}{\mathrm{standard\_deviation}(\mathbf{x}) * \mathrm{standard\_deviation}(\mathbf{y})} = \frac{s_{xy}}{s_x\, s_y},$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar



where the notations used are defined in standard as:

$$\mathrm{covariance}(\mathbf{x},\mathbf{y}) = s_{xy} = \frac{1}{n-1}\sum_{k=1}^{n}(x_k-\overline{x})(y_k-\overline{y})$$

$$\mathrm{standard\_deviation}(\mathbf{x}) = s_x = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(x_k-\overline{x})^2}$$

$$\mathrm{standard\_deviation}(\mathbf{y}) = s_y = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(y_k-\overline{y})^2}$$

$$\overline{x} = \frac{1}{n}\sum_{k=1}^{n}x_k \text{ is the mean of } \mathbf{x}$$

$$\overline{y} = \frac{1}{n}\sum_{k=1}^{n}y_k \text{ is the mean of } \mathbf{y}$$

Introduction to Data Mining — Pang-Ning Tan, Michael Steinbach, Vipin Kumar

Correlation Example: https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/

The Chi-square goodness of fit test:

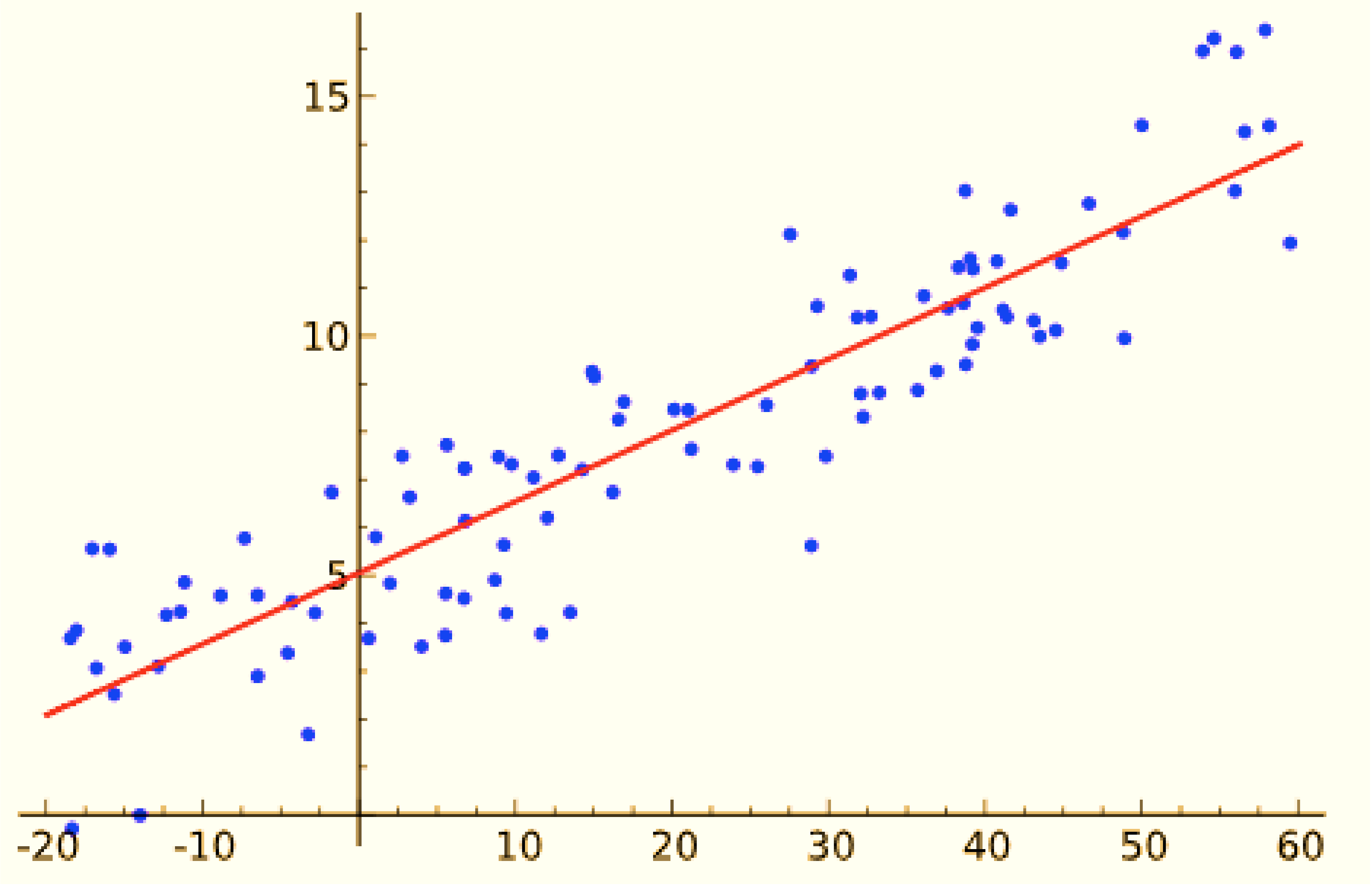
What is the goodness of fit?

A goodness-of-fit is a statistical technique. It is applied to measure "***how well the actual(observed) data points fit into a Machine Learning model***". It summarizes the divergence between actual observed data points and expected data points in context to a statistical or Machine Learning model.

Assessment of divergence between the observed data points and model-predicted data points is critical to understand, a decision made on poorly fitting models might be badly misleading. A seasoned practitioner must examine the fitment of actual and model-predicted data points.

Why do we test Goodness of fit?

Goodness-of-fit tests are statistical tests to determine whether a set of actual observed values match those predicted by the model. Goodness-of-fit tests are frequently applied in business decision making. For example, if we check linear regression function. The goodness-of-fit test here will compare the actual observed values to the predicted values.



What are the most common goodness of fit tests?

Broadly, the goodness of fit test categorization can be done based on the distribution of the predict and variable of the dataset.

- The chi-square
- Kolmogorov-Smirnov
- Anderson-Darling

The Chi-square test for a goodness-of-fit test is

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where:

O_i= an observed count for bin*i*

E_i= an expected count for bin*i*, asserted by thenull hypothesis.

The expected frequency is calculated by:

$$E_i = \Big(F(Y_u) - F(Y_l)\Big) N$$

where:

F= thecumulative distribution functionfor theprobability distributionbeing tested.

Y_u= the upper limit for class*i*,

Y_l= the lower limit for class*i*, and

N= the sample size

The Chi-Square Goodness of Fit Test

Chi-square goodness of fit test is conducted when the predictand variable in the dataset is categorical. It is applied to determine whether sample data are consistent with a hypothesized distribution.

Chi-Square test can be applied when the distribution has the following characteristics:

- The sampling method is random.
- Predicted variables are categorical.
- The expected value of the number of sample observations at each level of the variable is at least 5. It requires a sufficient sample size for the chi-square approximation to be valid.

Merits of the Chi-square Test

- A distribution-free test. It can be used in any type of population distribution.
- It is widely applicable not only in social sciences but in business research as well.
- It can be easy to calculate and to conclude.
- The Chi-Square test provides an additive property. This allows the researcher to add the result of independence to related samples.
- This test is based on the observed frequency and not on parameters like mean, and standard deviation.

Until now we have defined and understood both similarity and dissimilarity measures amongst data objects. Now, let’s discuss the issues faced in proximity calculations.

Issues in Proximity Calculation

- how to handle the case in which attributes have different scales and/or are correlated,
- how to calculate proximity between objects that are composed of different types of attributes, e.g., quantitative and qualitative, and
- how to handle proximity calculation when attributes have different weights i.e., when not all attributes contribute equally to the proximity of objects.

Selecting the Right Proximity Measure

The following are a few general observations that may be helpful. First, ***the type of proximity measure should fit the type of data***. For many types of dense, continuous data, metric distance measures such as Euclidean distance are often used. Proximity between continuous attributes is most often expressed in terms of differences, and distance measures provide a well-defined way of combining these differences into an overall proximity measure.

For sparse data, which often consists of asymmetric attributes, we typically employ similarity measures that ignore 0–0 matches. Conceptually, this reflects the fact that, for a pair of complex objects, similarity depends on the number of characteristics they both share, rather than the number of characteristics they both lack. For such type of data, Cosine Similarity or Jaccard Coefficient can be used.

Main Components of Machine Learning Algorithm:

1) Feature Extraction + Domain knowledge

First and foremost we really need to understand what type of data we are dealing with and what eventually we want to get out of it. Essentially we need to understand how and what features need to be extracted from the data. For instance assume we want to build a software that distinguishes between male and female names. All the names in text can be thought of as our raw data while our features could be number of vowels in the name, length, first & last character, etc of the name.

2) Feature Selection

In many scenarios we end up with a lot of features at our disposal. We might want to select a subset of those based on the resources and computation power we have. In this step we select a few of those influential features and separate them from the not-so-influential features. There are many ways to do this, information gain, gain ratio, correlation etc.

3) Choice of Algorithm

There are wide range of algorithms from which we can choose based on whether we are trying to do prediction, classification or clustering. We can also choose between linear and non-linear algorithms. Naive Bayes, Support Vector Machines, Decision Trees, k-Means Clustering are some common algorithms used.

4) Training

In this step we tune our algorithm based on the data we already have. This data is called training set as it is used to train our algorithm. This is the part where our machine or software learn and improve with experience.

5) Choice of Metrics/Evaluation Criteria

Here we decide our evaluation criteria for our algorithm. Essentially we come up with metrics to evaluate our results. Commonly used measures of performance are precision, recall, f1-measure, robustness, specificity-sensitivity, error rate etc.

6) Testing

Lastly, we test how our machine learning algorithm performs on an unseen set of test cases. One way to do this, is to partition the data into training and testing set. The training set is used in step 4 while the test set is then used in this step. Techniques such as cross-validation and leave-one-out can be used to deal with scenarios where we do not have enough data.

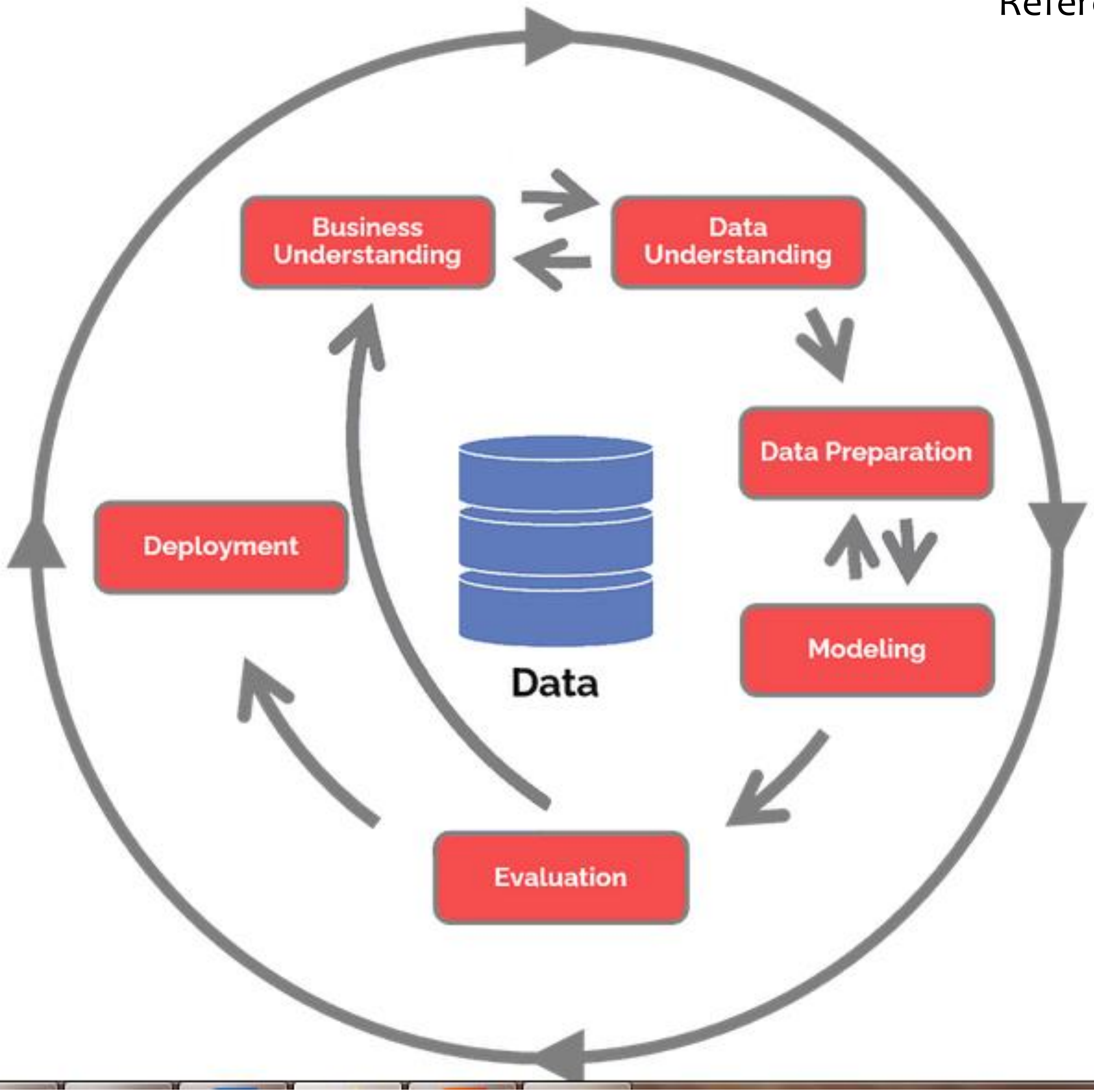
(Reference: https://www.linkedin.com/pulse/20140822073217-180198720-6-components-of-a-machine-learning-algorithm)

Main steps involved for End-to-End Machine Learning Project: (Chapter 2 of textbook)

1. Look at the big picture
2. Get the data
3. Discover and visualize the data to gain insights
4. Prepare the data for Machine Learning algorithms
5. Select a model and train it
6. Fine tune your model
7. Present your solution
8. Launch, monitor and maintain your system

CRISP DM (The **CR**oss **I**ndustry **S**tandard **P**rocess for **D**ata **M**ining)

Reference: https://www.datascience-pm.com/crisp-dm-2/



The content of the slides are prepared from different textbooks.

References:

Proximity Measures:

- https://towardsdatascience.com/measures-of-proximity-in-data-mining-machine-learning-e9baaed1aafb

Chi-Square Goodness of Fit readings:

- https://www.mygreatlearning.com/blog/understanding-goodness-of-fit-test/
- https://machinelearningmastery.com/chi-squared-test-for-machine-learning/
- https://towardsdatascience.com/machine-learning-chi-square-test-in-evaluating-predictions-486404dd5bc
- https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223
- https://www.analyticsvidhya.com/blog/2019/11/what-is-chi-square-test-how-it-works/ (stepwise calculation)
- https://medium.com/wenyi-yan/a-simple-explanation-to-understand-chi-square-test-1814fa261499 (step wise calculation simple)

Correlation:

https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

Extra Read:

https://towardsdatascience.com/types-of-data-sets-in-data-science-data-mining-machine-learning-eb47c80af7a

Thank you..