

# Table of contents

## **1. Introduction**

## **2. Data Types and Representations (Text 2.1)**

## **3. Basic Statistical Descriptions of a Single Data Variable (Text 2.2)**

3.1. Computation of Measures (Text: 4.2.4)

3.2. Measuring the Central Tendency: Mean, Median and Mode (Text 2.2.1)

3.3. Measuring the Dispersion of Data: Range, Quartile, Variance, Standard Deviation, and Interquartile Range (Text 2.2.2)

3.4. Exercises

## **4. Measuring Correlation amongst Two Variables (not in Text)**

## **5. Measuring Similarity and Dissimilarity of Multivariate Data (Text 2.4)**

5.1. Proximity Measures for Nominal Attributes (Text 2.4.2)

5.2. Proximity Measures for Binary Attributes (Text 2.4.3)

5.3. Dissimilarity of Numeric Data: Normalisation (Text 3.5.2)

5.4. Dissimilarity of Numeric Data: Minkowski Distance (Text 2.4.4)

5.5. Proximity Measures for Ordinal Attributes (Text 2.4.5)

5.6. Dissimilarity for Attributes of Mixed Types (Text 2.4.6)

5.7. Cosine Similarity for Sparse Vectors (Text 2.4.7)

5.8. Exercises

# 1. Introduction

Most of this material is derived from the text, Han, Kamber and Pei, Chapter 2, or the corresponding powerpoint slides made available by the publisher. Where a source other than the text or its slides was used for the material, attribution is given. Unless otherwise stated, images are copyright of the publisher, Elsevier.

Here, we introduce some important concepts that are used right across data mining methods and the KDD process.

**ACTION: If you would like a pre-recorded lecture for this topic, then watch this. Or you can dive straight into the next sections -- it is all there.**

44 minutes

introweek2 (Copy)

RESOURCES • HEALTH AND WELLBEING • LIBRARY • WATTLE SUPPORT • ENGLISH (EN) •

Kerry Taylor  
Student

on campus only" class="bookprev" href="https://phn7id=15710218&chapterId=313336"> <

2. Data Types and Representations (Text 2.1)

Data is generally structured as **objects**, with each object described by some kind of **properties**, or **attributes**, or **features**, of some real-world or abstract entity. Some of those **attributes** may include relationships to other objects.

- Example: a Medical database may include information about **patient** objects and **treatment** objects with relations connecting patients to the treatments they are given
- Example: Attributes of a patient may include name, gender and birth date. Attributes of a treatment may include treatment name, responsible doctor, and date.

The objects may alternatively be called *samples*, *examples*, *instances*, *data points*, *observations*, *rows*, *tuples*, *records*, or *vectors*. The attributes may be called *fields*, *variables*, *dimensions*, *properties*, or *features*. This will depend on the data model or data mining discipline you are working with, and you need to be comfortable with all of those names.

- 1. Introduction on campus only
- 2. Data Types and Representations (Text 2.1)
- 3. Basic Statistical Descriptions of a Single Data Variable (Text 2.2)
  - 3.1. Computation of Measures (Text 4.2.4)
  - 3.2. Measuring the Central Tendency: Mean, Median and Mode (Text 2.2.1)
  - 3.3. Measuring the Dispersion of Data:

0:00/44:05

CC 1x

讨论 0

扩展

## 2. Data Types and Representations (Text 2.1)

Data is generally structured as **objects**, with each object described by some kind of **properties**, or **attributes**, or **features**, of some real-world or abstract entity. Some of those attributes may include **relationships** to other objects.

- Example: a Medical database may include information about patient objects and treatment objects with relations connecting patients to the treatments they are given
- Example: Attributes of a patient may include name, gender and birth date. Attributes of a treatment may include treatment name, responsible doctor, and date.

The objects may alternatively be called *samples*, *examples*, *instances*, *data points*, *observations*, *rows*, *tuples*, *records*, or *vectors*. The attributes may be called *fields*, *variables*, *dimensions*, *properties*, or *features*. This will depend on the data model or data mining discipline you are working with, and you need to be comfortable with all of those names.

Commonly, a **data model** (also called a **meta-model**) dictates how these concepts are materialised in a data structure, such as the relational data model, for example. Alternatively, a **data model** could refer to a model that instantiates a particular meta-model for some domain of application, for example, the particular relational schema that is used in your hospital's medical records system.

In this course, we will be employing the relational data model, multidimensional data models, a transactional data model, simple tables, graph and ontology data models, and others.

### Attribute Types

The type of an attribute defines the way its values for some object are represented, the ways it can be manipulated, and appropriate methods for mining.

#### Nominal

- categories, states, classes, or “names of things”
- also called **categorical**
- e.g. Hair\_color = {auburn, black, blond, brown, grey, red, white}
- e.g. marital status, occupation, ID numbers, postcodes

#### Binary

- Nominal attribute with only 2 states (0 and 1)
- Symmetric binary: both outcomes equally important
  - e.g., gender
- Asymmetric binary: outcomes not equally important.
  - e.g., medical test (positive vs. negative). e.g. attended class (no, yes)
  - Convention: assign 1 to most important outcome (e.g., HIV positive)

#### Ordinal

- Values have a meaningful order (ranking) but magnitude between successive values is not known.
- e.g. Size = {small, medium, large}, grades, army rankings
- Commonly used in surveys e.g. {very poor, poor, moderate, good, excellent}
- Can be derived by discretising numeric quantities into ordinal categories (e.g. ANU marks -> grades or age -> child/adult).

#### Numeric Interval-Scaled

- Could be integer or real
- Measured on a scale of equal-sized units
- Values have order

- No true zero-point
- e.g, temperature in °C, calendar dates
- We can reason about the difference amongst values, but not about the multiples or ratios of values.
  - e.g., 20°C is 10°C warmer than 10°C but it is not twice as warm as 10°C.
- Are not very common in practice

#### Numeric Ratio-Scaled

- Could be integer or real
- Inherent zero-point
- A ratio scale has all the properties of interval-scaled attributes and the additional property of permitting reasoning over multiples of values
- We can speak of values as being a multiple of another in the same unit of measurement (10 K° is twice as high as 5 K°).
- We can also reason about the difference between values, as for interval-scaled.
- e.g., temperature in Kelvin, length, counts, monetary quantities

### Discrete versus Continuous Attributes

Another classification of attributes, more often used in machine learning:

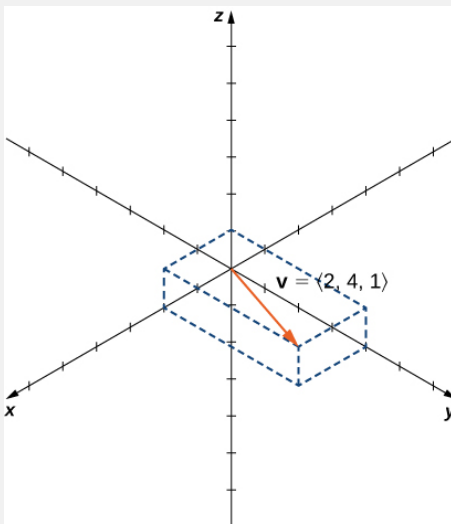
#### Discrete Attribute

- Has only a finite or countably infinite set of values
- e.g., postcodes, profession, or the set of words in a collection of documents
- Sometimes, represented as integer variables
- Binary, nominal and ordinal attributes are a special case of discrete attributes

#### Continuous Attribute

- real numbers as values
- e.g., temperature, height, weight, circumference
- In practice, real values can only be measured and represented using a finite number of bits (so in that sense are approximated by discrete values)
- Typically represented as floating-point variables

Commonly, an object with continuous attributes is called a *feature vector* and is identified with a point or vector in multidimensional space. For example,  $\mathbf{v} = (2.0, 4.0, 1.0)$  is a vector in 3-dimensional Euclidean space which can be sketched as follows.



>Object  $\mathbf{v}$  shown as a red vector in 3-dimensional space. Source: <https://math.libretexts.org>

### 3. Basic Statistical Descriptions of a Single Data Variable (Text 2.2)

Simple statistics can help us to understand the data we are working with, and by that to select appropriate cleaning and data mining methods, and to interpret data mining results. We will look at the **central tendency** (*what is the middle value?*), and **dispersion** (*how uniform are the values? and how are they spread?*).

These should mostly be concepts you have seen before.

But before that, we look briefly at the cost of computing these and other functions in data mining.

## 3.1. Computation of Measures (Text: 4.2.4)

When analysis or data mining needs to operate over big data, we need to think about the computational cost of all stages of the processing.

When assessing the character of data, when summarising data, and when developing measures of similarity and interestingness of data and data patterns, some measures are computationally more difficult to compute than others.

The classification of measure functions as *distributive*, *algebraic* or *holistic* is useful to understand their computational complexity.

### Distributive

Suppose the data is partitioned into  $n$  sets, and the function is applied to each of those sets. Then, suppose the function is applied to those  $n$  results. If the result is the same as applying the function to all of the data without partitioning, then the function is *distributive*. e.g., `count()`, `sum()`, `min()`, `max()`

Such computations can be made efficient by distributing the computation.

### Algebraic

A function is *algebraic* if it can be computed by a function with  $M$  arguments (where  $M$  is a bounded integer), each of which is obtained by applying a distributive function.

e.g. `average()` can be computed from `sum()` and `count()`.

e.g. `min_N()` and `max_N` ( $N$  minimum/maximum values for integer  $N$ )

e.g. `standard_deviation()`

Such computations can take advantage of the distributive sub-functions to be made efficient.

### Holistic

A function is *holistic* if there is no constant bound on the storage size needed to describe sub-functions. That is, there is no algebraic function with  $M$  arguments (where  $M$  is a bounded integer) that characterises the computation.

e.g. `median()`, `mode()`, `rank()`

Such computations are difficult to compute efficiently and often efficient *approximations* are used instead.

## 3.2. Measuring the Central Tendency: Mean, Median and Mode (Text 2.2.1)

Let  $X$  be an attribute (e.g. *salary*).

Let  $x_1, x_2, \dots, x_n$  be a set of  $n$  observations or observed values for  $X$ .

Arithmetic Mean (or, simply *Mean*)

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Weighted (Arithmetic) Mean

Used when some values are more important or more significant than others (e.g. occurrence frequency, or weightings for most important people represented in the data). Also called **weighted average**.

$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}$$

Can also use **trimmed mean** that removes a portion of the highest and lowest values and compute the mean of the remainder - to avoid being pushed off centre by extreme values.

Median

- The middle value of an ordered set of values (or the average of the middle two values if there is an even number of values).
- Represents the centre better than the mean when data is skewed (asymmetrical) towards one end or has extreme values.
- Works for ordinal data as well as numeric data.
  - But if there is an even number of values then it is not unique: the two middlemost values and anything in between those values
- Can be expensive to compute but can be approximated by interpolation when data values are grouped into ranges with frequencies.

Mode

- Defined as value that occurs most frequently in the data
- Works for numeric and nominal or ordinal data
- May not be unique - data sets with multiple modes are called unimodal, bimodal, trimodal (1,2,3 modes respectively)
- An empirical formula that applies to *unimodal* data that is at most moderately skewed, meaning that the mode can be easily approximated from mean and median:

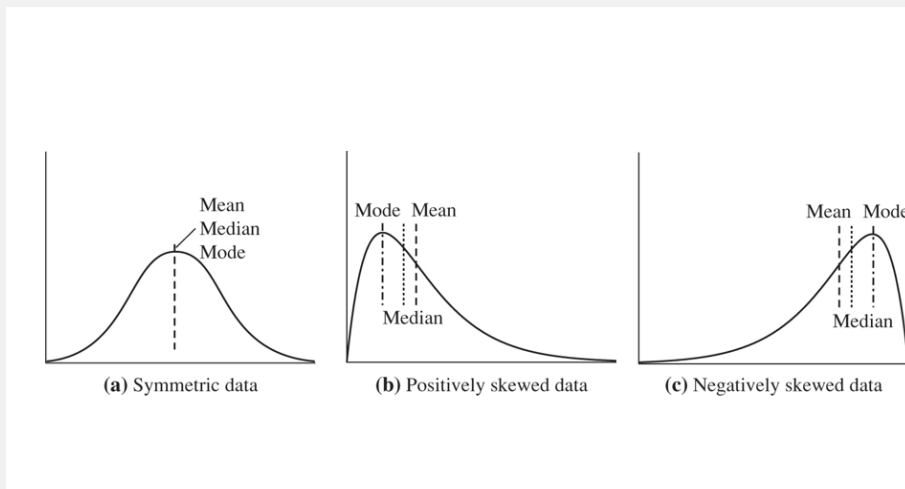
$$mean - mode \approx 3 \times (mean - median)$$

Midrange

- Defined as the average of the largest and smallest values
- Cheap to compute as indication of central tendency of the data

Skewness

The figure illustrates the main central tendency measures of symmetric and skewed data.



Skewness is a measure of asymmetry in data. As seen here the sign and magnitude of the difference between mean and mode is an simple indicator of skewness in the data. There are many other ways to measure skewness.



### 3.3. Measuring the Dispersion of Data: Range, Quartile, Variance, Standard Deviation, and Interquartile Range (Text 2.2.2)

Let  $X$  be a **numeric** attribute (e.g. *salary*).

Let  $x_1, x_2, \dots, x_n$  be a set of  $n$  observations or observed values for  $X$ .

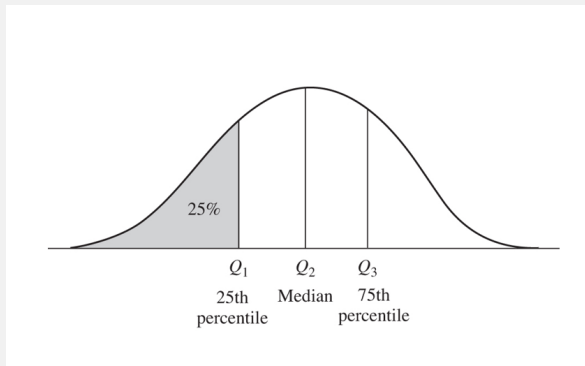
#### Range

The range is the difference between the largest and smallest values, i.e.  $\max(X) - \min(X)$

#### Quartiles

**Quantiles** are data values that split the data into (nearly) equal-sized increasing-order subsets.

- Given an integer  $k$  such that  $0 < k < q$ , the  $k$ th  $q$ -quantile is the value  $x$  such that
  - at most  $k/q$  of the values are less than  $x$  and
  - at most  $(q-k)/q$  are greater than  $x$
- There are  $q-1$   $q$ -quantiles.
- The 2-quantile splits the ordered data into 2 parts and is also called the **median**.
- The 4-quantiles are called **quartiles**.
- The 100-quantiles are called **percentiles**.

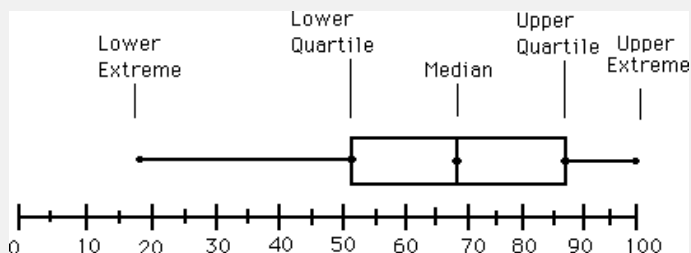


#### Interquartile range (IQR)

The distance between the first and third quartiles gives the range of values covered by the middle half of the data and is a simple indicator of spread.

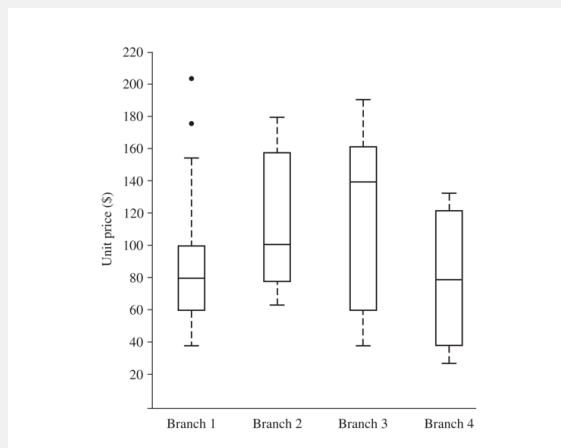
$$\text{IQR} = Q_3 - Q_1$$

#### Five-number Summary and Boxplots



- Five-number summary**, In order: **min**, **Q1**, **median=Q2**, **Q3**, **max**
- Convenient summary of data dispersion
- Illustrated in a **boxplot**.
- Boxplots can be used to visually compare sets of compatible data.
- Ends of the **box** are the quartiles; median is marked by a line; **whiskers** extend to min and max.

- **Outlier:** usually, a value below  $Q1 - 1.5 \text{ IQR}$  or above  $Q3 + 1.5 \text{ IQR}$ 
  - If there are outliers, then the whiskers commonly extend to  $1.5 \times \text{IQR}$  (instead of min, max) and outliers are plotted individually.
  - The "tailedness" or frequency of outliers compared to a central peak of the distribution in each direction is easily seen by the length of the whiskers and occurrence of individual outliers. A formal measure of tailedness called *kurtosis* can be calculated instead.



## Standard Deviation and Variance

- Used very commonly to indicate data dispersion
- A low standard deviation indicates that values tend to be close to the mean
- Is cheap to compute
- Measures deviation around the mean so should only be used when the mean is also used for centrality
- $\sigma = 0$  only when all values are the same
- An observation is unlikely to be more than several standard deviations from the mean.
- Formally (**Chebyshev's inequality**),

**At least  $(1 - 1/k^2) \times 100\%$  of the observations are no more than  $k$  standard deviations from the mean.**

Variance,  $s^2$  or  $\sigma^2$ :

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

Standard deviation,  $s$  or  $\sigma$

- is the square root of variance  $s^2$  (or  $\sigma^2$ )

## Population or Sample?

In the formula above, the so-called "n-1" method is used, due to the "n-1" term in the denominator. You will often see standard deviation or variance alternatively written with "n", the size of the data, instead of "n-1". For example, the course text uses that "n" form. Strictly, the "n-1" method is properly used for observational data that is understood to be a *sample* (say, collected by a survey) selected from the complete population, and the standard deviation computed this way is an *estimation* of the real standard deviation of the complete population, and this estimation minimises the expected error wrt the real standard deviation. On the other hand, the "n" method computes exactly the standard deviation of the observations at hand which are then assumed to be the complete *population* real data (say, collected by a population census). We use the "n-1" form here under the assumption that we are interested in estimating the spread of the intended population, not just the sample to hand. Of course, for very large data sets to which data mining is normally applied, the difference is too small to be a matter of concern.




## 3.4. Exercises

**ACTION: Try out these exercises:**

 [Exercises for Basic Statistical Descriptions of Data](#)

When you have had a go, you can check your answers against these worked answers:

 [Answers to Exercises for Basic Statistical Descriptions of Data](#)


## 4. Measuring Correlation amongst Two Variables (not in Text)

There are some measures commonly used to assess whether variables are **correlated**, that is that one can (approximately) be computed directly from the other.

### Pearson's Correlation

- Measures how well a straight line can be fitted to two continuous attributes (one on each axis).
- Also called Pearson's  $r$ , Pearson product-moment correlation, Pearson correlation coefficient, bivariate correlation
- Ranges from -1 (perfectly negatively correlated), through 0 (no correlation) to 1 (perfectly positively correlated).

**ACTION:** Always be aware that *correlation between attributes does not imply that either causes the other*. If you do not immediately understand this, or are in need of some light relief, do have a look at

 [Spurious Correlations](#)

For  $n$  objects  $(x_1, y_1, z_1, \dots), \dots (x_i, y_i, z_i, \dots), \dots (x_n, y_n, z_n, \dots)$  Pearson's  $r$  of the pair of attributes  $x$  and  $y$  can be calculated by:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

- This formulation relies on the [standard deviation formula for a sample](#), so we are assuming we are using a sample to estimate correlation in the real, full, population data here.
- For nominal attributes, the [chi-square test](#) (covered later in the course), that considers the frequency of co-occurrence of nominal values, should be used instead.

## 5. Measuring Similarity and Dissimilarity of Multivariate Data (Text 2.4)

We need algorithmic ways to assess how alike or unlike objects are to others e.g. Are some set of taxpayers similar in their characteristics of age, employment status and income?

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

### Dissimilarity Matrix

- Common data structure used to represent dissimilarity amongst objects
- Used e.g. in main-memory based algorithms for clustering and nearest neighbour
- For  $n$  data points, an  $n \times n$  matrix data structure that shows dissimilarity or distance  $d(i,j)$  for all pairs of  $n$  objects  $i$ , and  $j$
- $d(i,j)$  is close to 0 when objects  $i$ , and  $j$  are near to each other, i.e. the distance is small.
- It is defined that  $d(i,j) = d(j,i)$
- It is defined that  $d(i,i) = 0$
- Therefore the matrix is **symmetric** or **triangular** with zeros along the diagonal and duplicates (often not represented at all) in the upper right triangle.
- **Single mode** or **one-mode** matrix because rows and columns represent the same things (i.e. objects).

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

- **Similarity**,  $sim(i,j)$  is often expressed as a function of dissimilarity, e.g. when  $d(i,j)$  is normalised to [0,1] we may have

$$sim(i,j) = 1 - d(i,j)$$

So far we have described proximity amongst pairs of objects, but each of those objects is comprised of values for attributes that may be of heterogeneous types. In the following we show how distances may be calculated for each attribute type, and then how to combine attribute distances of mixed types to give an overall object distance.



## 5.1. Proximity Measures for Nominal Attributes (Text 2.4.2)

A nominal attribute may have two or more states as values (if only two, then it is binary).

Let the number of states for a nominal attribute be  $M$ .

For convenience, we denote the states by integers  $1, 2, \dots, M$  but note that no ordering is implied here.

### Method 1: Simple matching

Let  $m$  be the number of matches i.e. the number of attributes for which object  $i$  and  $j$  have the same value.

Let  $p$  be the number of variables (i.e attributes for each object).

Then, define

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

Similarity may then be defined as

$$sim(i, j) = 1 - d(i, j) = m/p$$

### Method 2: Map nominals to a large number of binary attributes

- Create a new binary attribute for each of the  $M$  nominal states of the attribute.
- e.g. for attribute Colour = {red, blue, green}, introduce binary attributes Red, Blue and Green and map an object with Colour = red to the corresponding object with attribute Red = 1, attribute Blue = 0 and attribute Green = 0.
- Then use any method for the proximity of binary attributes.



## 5.2. Proximity Measures for Binary Attributes (Text 2.4.3)

- Binary attributes have only one of two values: 0 or 1
- The attributes may be symmetric or asymmetric: asymmetric means the 1-value state is more important than the 0-value state.
- Treating the values as numeric can be misleading
- The methods presented in this subsection require that all the attributes of the objects are binary

### Contingency Table

Let us consider a pair of objects  $i$  and  $j$  with a total number of attributes  $p$ .

Let  $q$  be the number of attributes that equal 1 for both  $i$  and  $j$ .

Let  $r$  be the number of attributes that equal 1 for object  $i$  but 0 for object  $j$ .

Let  $s$  be the number of attributes that equal 0 for object  $i$  but 1 for object  $j$ .

Let  $t$  be the number of attributes that equal 0 for both object  $i$  and object  $j$ .

Then  $p = q + r + s + t$  and we have the contingency table below.

		Object $j$		
		1	0	sum
Object $i$	1	$q$	$r$	$q + r$
	0	$s$	$t$	$s + t$
sum		$q + s$	$r + t$	$p$

### Symmetric binary dissimilarity

Then the dissimilarity is defined as

$$d(i, j) = \frac{r + s}{q + r + s + t}$$

That is, the proportion of non-matching attributes amongst all attributes.

### Asymmetric binary dissimilarity

The agreement of two 1 values (i.e. the agreement of the important values, a *positive match*) counts more than the agreement of two 0 values (i.e. a *negative match*). Dissimilarity is defined as

$$d(i, j) = \frac{r + s}{q + r + s}$$

The corresponding similarity, **asymmetric binary similarity**, is also called the **Jaccard coefficient** and is widely used.

It is also called *coherence*.

$$sim(i,j)=1-d(i,j)=sim_{Jaccard}(i,j)=\frac{q}{q+r+s}$$

Alternatively, viewing each object as set of asymmetric attributes, where each attribute is either present or absent (and so binary), the Jaccard similarity between two objects can be thought of as the size of the intersection of the two sets divided by the size of the union of the sets.

### Example of symmetric binary dissimilarity.

Consider this morbidity data.

Name	Adult/Child	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	A	Y	N	P	N	N	N
Mary	C	Y	N	P	N	P	N
Jim	A	Y	P	N	N	N	N

Instead of using a contingency table as above, we simply count the non-matching attributes, which is equivalent.

$$\text{So, } d(Jack, Mary) = \frac{1+0+0+0+0+1+0}{7} = 0.29$$

$$\text{And } d(Jack, Jim) = \frac{0+0+1+1+0+0+0}{7} = 0.29$$

$$\text{And } d(Mary, Jim) = \frac{1+0+1+1+0+1+0}{7} = 0.57$$

### Example of asymmetric binary dissimilarity.

In this same example of morbidity data, the attribute Adult/Child is to be treated as a symmetric attribute and all the other attributes are treated as asymmetric. Now we calculate distance on the basis of the asymmetric attributes alone. Later, we will see how distance can be computed by including attributes of mixed types.

For the Y/N-coded asymmetric attributes, 'Y' should be considered as the significant value for which we will consider a positive match to contribute to similarity, a mismatch to contribute to dissimilarity, and a negative match of 'N' to be ignored. For this reason we map 'Y' to 1 and 'N' to 0 for the Y/N-coded attributes. Likewise, for the P/N-coded asymmetric attributes, the values 'P' for the asymmetric attributes should be considered as significant so we map 'P' to 1 and 'N' to 0 for the P/N-coded attributes.

$$\text{Therefore } d(Jack, Mary) = \frac{0+1}{2+0+1} = 0.33, \text{ by the formula above, or equivalently, by checking for matches of each attribute at a time in order, } d(Jack, Mary) = \frac{0+0+0+0+1+0}{1+0+1+0+1+0} = 0.33$$

$$\text{And } d(Jack, Jim) = \frac{1+1}{1+1+1} = 0.67, \text{ and } d(Jack, Jim) = \frac{0+1+1+0+0+0}{1+1+1+0+0+0} = 0.67.$$

$$\text{And } d(Mary, Jim) = \frac{1+2}{1+1+2} = 0.75, \text{ and } d(Mary, Jim) = \frac{0+1+1+0+1+0}{1+1+1+0+1+0} = 0.75.$$

## 5.3. Dissimilarity of Numeric Data: Normalisation (Text 3.5.2)

Numeric data is commonly *normalised* (also called *standardised*) before calculating distance. This is done by scaling all attributes to a common range such as [0.0,1.0] so that all attributes are weighted evenly. Otherwise, attributes with a greater range will have a greater effect on the distance measure.

### Normalisation of numeric data

Methods for normalisation are not covered here in depth. Here is one method, **min-max normalisation**, that normalises values by a linear transformation.

Each value for an attribute,  $v_i$  is normalised by replacing it with a value  $v_i'$  that falls within a desired range [ $new\_min$ ,  $new\_max$ ], defined as:

$$v_i' = (v_i - min) \frac{(new\_max - new\_min)}{max - min} + new\_min$$

where  $min$ ,  $max$  are the minimum and maximum values respectively in the original attribute and  $new\_min$ ,  $new\_max$  are the desired minimum and maximum values respectively in the transformed attribute.

If you want to ensure normalised values fall in [0.0,1.0], you can set  $new\_min$  to 0 and  $new\_max$  to 1.

## 5.4. Dissimilarity of Numeric Data: Minkowski Distance (Text 2.4.4)

Popular distance measures for **numeric** attributes are *Euclidean*, *Manhattan* and *Minkowski* measures that are closely related.

Data is commonly *normalised* (also called *standardised*) before calculating distance. This is done by scaling all attributes to a common range such as [0.0,1.0] so that all attributes are weighted evenly. Otherwise, attributes with a greater range will have a greater effect on the distance measure.

### Metric Measures

A measure is a **metric** when it satisfies the following properties that have some intuitive appeal.

- **Identity of indiscernibles:**  $d(i,i) = 0$  The distance from an object to itself is 0.
- **Symmetry:**  $d(i,j) = d(j,i)$  The distance between objects is the same irrespective of direction.
- **Triangle inequality:**  $d(i,j) \leq d(i,k) + d(k,j)$  Going directly from object  $i$  to object  $j$  is *no further than making a detour via any other object  $k$*
- **Non-negativity:**  $d(i,j) \geq 0$  Distance is a non-negative number, This is a consequence of the 3 properties above.

*Euclidean*, *Manhattan* and *Minkowski* measures are *metrics*.

### Minkowski Distance

Minkowski is a *family* of several popular metrics, where the parameter  $h$  selects a particular metric.

For  $p$ -dimensional data objects  $i = (x_{i1}, x_{i2}, \dots, x_{ip})$  and  $j = (x_{j1}, x_{j2}, \dots, x_{jp})$  i.e., two objects with  $p$  attributes

Minkowski distance of **order**  $h$  is:

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

### Manhattan (City Block) Distance

- Distance in blocks between any two points in a city, where corners cannot be cut.
- Also called **Hamming distance** when the  $x_{ij}$ s are bits: i.e. counts the number of bits that are different between two binary vectors
- Minkowski distance with  $h = 1$ .
- Also called the **L<sub>1</sub> norm**

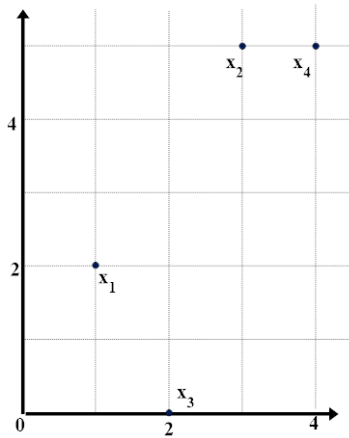
$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

### Euclidean Distance

- Distance in a straight line between any two points, cutting corners (*Pythagoras's Theorem*)
- Minkowski distance with  $h = 2$ .
- Also called the **L<sub>2</sub> norm**

$$d(i,j)=\sqrt{(|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+...+|x_{ip}-x_{jp}|^2)}$$

## Example: Data Matrix and Dissimilarity Matrix



**Data Matrix**

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

**Dissimilarity Matrix**  
(with **Euclidean Distance**)

	$x1$	$x2$	$x3$	$x4$
$x1$	0			
$x2$	3.61	0		
$x3$	5.1	5.1	0	
$x4$	4.24	1	5.39	0

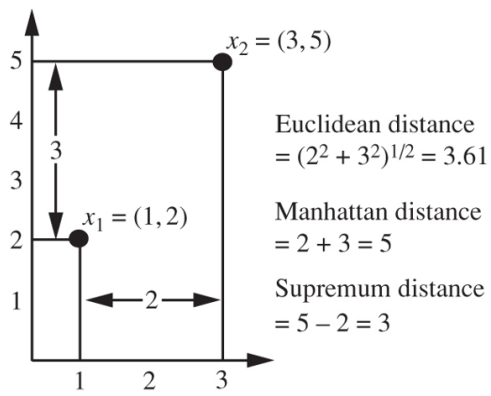
N.B. There is an error in this table above:  $d(x1,x3)=\sqrt{1+4}=2.23$

### Supremum Distance

- The maximum of all the individual attribute distances.
- Also called **Chebyshev distance** and the **uniform norm**.
- Minkowski distance with  $h \rightarrow \infty$ .
- Also called the  **$L_\infty$  norm** and  **$L_{max}$**

$$d(i,j)=\lim_{h \rightarrow \infty} \left( \sum_{f=1}^p |x_{if}-x_{jf}|^h \right)^{\frac{1}{h}} = \max_f^p |x_{if}-x_{jf}|$$

### Examples of Minkowski Distances



## Example: Minkowski Distance

Dissimilarity Matrices

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

### Manhattan ( $L_1$ )

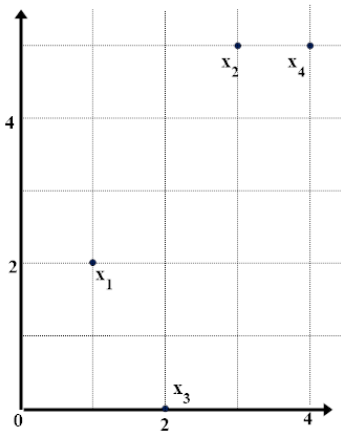
L	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

### Euclidean ( $L_2$ )

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

### Supremum

$L_\infty$	x1	x2	x3	x4
x1	0			
x2	3	0		
x3	2	5	0	
x4	3	1	5	0



## 5.5. Proximity Measures for Ordinal Attributes (Text 2.4.5)

Ordinal data has a meaningful ordering amongst values, so some pairs of values are closer together than others and this should be taken into account.

The approach maps ordinal values to numeric values so that numeric attribute methods can be used.

Let  $f$  be an attribute from a set of  $p$  ordinal attributes where  $\{x_{i1}, x_{i2}, x_{if}, \dots, x_{ip}\}$  describes the  $i$ th object of  $n$  objects.

Let  $M_f$  be the number of possible states for attribute  $f$  where  $f$  in  $\{1, \dots, p\}$  and let  $1, \dots, M_f$  be the ranking defined by the attribute values arranged in increasing order.

Then, to compute the dissimilarity with respect to attribute  $f$  do:

1. For each object  $i = \{x_{i1}, x_{i2}, x_{if}, \dots, x_{ip}\}$ , replace  $x_{if}$  by its corresponding rank  $r_{if}$  in  $\{1, \dots, M_f\}$ .
2. Normalisation: Scale the range of each attribute to  $[0, 1]$  so that each attribute has equal weight irrespective of the number of distinct states it has.

To do this, replace the rank  $r_{if}$  of the  $i$ th object in the  $f$ th attribute by the normalised rank  $z_{if}$  given by:

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

3. Compute **dissimilarity** by any distance measure for numeric attributes, using  $z_{if}$  to represent the  $f$  value for the  $i$ th object.

## 5.6. Dissimilarity for Attributes of Mixed Types (Text 2.4.6)

So far we have considered methods for dissimilarity for *nominal*, *symmetric binary*, *asymmetric binary*, *numeric* and *ordinal* attributes, but only when all attributes of an object are the *same type*. This is rarely the case in practice, where *mixed* attribute types are common.

One technique combines all the attributes into a single dissimilarity matrix, bringing all attributes to a common scale of the interval [0.0, 1.0].

Let the data set contain  $p$  attributes of mixed type. The dissimilarity of objects  $i$  and  $j$  is defined as the **average of the normalised distances between each attribute** of the pair of objects. That is,

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

where:

$p$  is the number of attributes for each object, and  $f$  iterates of the attributes (or *features*), and  $x_{if}$  is the value for the  $f$ th attribute of object  $x_i$ ; and

indicator  $\delta_{ij} = 0$  if either

(1)  $x_{if}$  or  $x_{jf}$  is missing; or

(2)  $f$  is asymmetric binary and  $x_{if} = x_{jf} = 0$

or 1 otherwise; and

and depending on the type of  $f$  either:

(1) attribute  $f$  is binary or nominal and  $d_{ij}^{(f)} = 0$  if  $x_{if} = x_{jf}$ , or  $d_{ij}^{(f)} = 1$  otherwise

(2) attribute  $f$  is numeric and  $d_{ij}^{(f)}$  is a normalised distance measure for  $f$  on [0.0, 1.0].

(3) attribute  $f$  is ordinal and  $d_{ij}^{(f)}$  is determined from the normalised rank  $z_{if}$  treated as a numeric interval-scaled attribute

derived from  $x_{if}$  by computing rank  $r_{if}$  and  $z_{if} = \frac{r_{if} - 1}{M_f - 1}$

**Informally**, and leaving out the asymmetric binary case for simplicity, this means:

Treat each attribute according to its type as before. For **binary** and **nominal** assign 0 if it matches, 1 if it doesn't and then treat as numeric. For **ordinal** use the normalised rank and treat as numeric. For **numeric** use difference of values normalised to [0,1] (which can be simply their difference divided by the difference of the max and min values, or equivalently a normalised L1 norm of one variable).

Then add up all the one-attribute distances and divide by the number of attributes to put on [0,1] scale. This would be an L1 norm (normalised). Or combine all the one-attribute distances according to one of the other Minkowski distance metrics.

Example



Consider attributes numerical  $a_1$  with values ranging from 0 to 1, ordinal  $a_2$  with values 1 to 5, and nominal  $a_3$  with values drawn from {red, yellow, blue}.

Let the objects  $X = (0.1, 2, \text{red})$  and  $Y = (0.2, 1, \text{red})$ . Then to calculate distance,  $a_1$  is already normalised and treated as a numeric,  $a_2$  is translated to a normalised rank value and  $a_3$  is handled by simple matching.

The Manhattan (city block), L1 normalised distance( $X, Y$ ) =  $((0.2 - 0.1) + (1 - 0)/4 + (0)) / 3 = 0.35 / 3 = 0.117$

## 5.7. Cosine Similarity for Sparse Vectors (Text 2.4.7)

- Measures similarity between two vectors of a multidimensional space, determining whether they are pointing in a similar direction
- Used for **sparse numeric data** where previous methods do not work well ( where data is dominated by zeros and zero-matches are not interesting).
- Classic application is similarity of documents of words (information retrieval, text document clustering)
- Also used in biological taxonomy and gene expression in microarrays

### Cosine similarity

Given  $x$  and  $y$  are two vectors then

$sim(x, y) = cos(x, y) = \frac{x \bullet y}{\|x\| \|y\|}$  where  $\|x\|$  is the Euclidean norm for vector  $x$  and  $\bullet$  is the vector dot product.

- **Euclidean norm** is defined as the Euclidean distance,  $L_2$  norm, between  $x$  and the zero vector (i.e the square root of the sum of the squares of each  $x_i$  in  $x$ ).
- **Vector dot-product** over  $x, y$  is defined as  $\sum_{i=1}^n x_i \times y_i$
- Cosine similarity computes the cosine of the angle between the two vectors, ranging from 0 to 1. A value of 0 means the vectors are orthogonal and do not match at all. The closer the value to 1, the smaller the angle and so the greater the match.
- Cosine similarity is *not* a metric measure.
- It measures *similarity* but can be converted to *distance* by subtracting from 1 as it is already normalised to [0,1] by the Euclidean norm in the denominator as given above.

Example of cosine similarity.

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

Document	team	coach	hockey	baseball	soccer	penalty	score	win	loss	season
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

Find the similarity between Document1 and Document2.

$$d1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$$

$$d2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$$

$$d1 \bullet d2 = 5*3 + 0*0 + 3*2 + 0*0 + 2*1 + 0*1 + 0*1 + 2*1 + 0*0 + 0*1 = 25$$

$$\|d1\| = (5*5 + 0*0 + 3*3 + 0*0 + 2*2 + 0*0 + 0*0 + 2*2 + 0*0 + 0*0)^{0.5} = (42)^{0.5} = 6.481$$

$$\|d2\| = (3*3 + 0*0 + 2*2 + 0*0 + 1*1 + 1*1 + 0*0 + 1*1 + 0*0 + 1*1)^{0.5} = (17)^{0.5} = 4.12$$

$$sim(d1, d2) = cos(d1, d2) = 0.94$$

## 5.8. Exercises

**ACTION: Try out these exercises:**

 [Exercises for Proximity Measures](#)

When you have had a go, you can check your answers against these worked answers:

 [Answers to Exercises for Proximity Measures](#)