Distance Measure

A particular object, which other objects in the same data set are more similar and which are more dissimilar.

A common approach to associate a number with the similarity (and dissimilarity) between two objects is to use distance measures.

The most similar objects have the smallest distances between them, and the most dissimilar have the largest distances.

The way we compute the distance between objects depends on the scale type of its attributes: whether they are **quantitative or qualitative**.

Differences between Values of Common Attribute Types

The difference between two values for the same attribute, here named *a* and *b*, will be denoted as *d*(*a*, *b*). For quantitative attributes, one can calculate the absolute difference:

Example 5.1 For example, the difference in age between Andrew ($a = 55$) and Carolina ($b = 37$) is $55 - 37 = 18$. Note, that even if we change the order of the values ($a = 37$ and $b = 55$) the result is the same.

If the attribute type is qualitative, we use distance measures suitable for the given type. If the qualitative attribute has ordinal values, we can measure the difference in their positions as:

$$d(a, b) = (pos_a - pos_b)/(n - 1)$$

where *n* is the number of different values, and pos_a and pos_b are the positions of the values *a* and *b*, respectively, in a ranking of possible values.

Distance Measures for Objects with Quantitative Attributes

Several distance measures are particular cases of the Minkowski distance. The Minkowski distance for two *m*-dimensional objects *p* and *q* with quantitative attributes is given by:

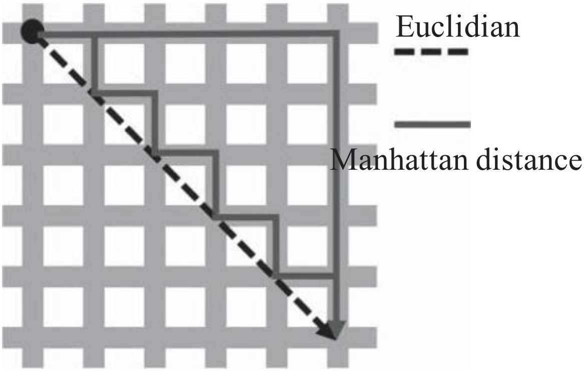
$$_{k}{}^{r}$$

where *m* is the number of attributes, while p_k and q_k are the values of the *k* th attribute for objects *p* and *q*, respectively. Variants are obtained using different values for *r*. For example, for the Manhattan distance, $r = 1$, and for the Euclidean distance, $r = 2$.

The Manhattan distance is also known as the city block or taxicab distance, since if you are in a city and want to go from one place to another, it will measure the distance traveled along the streets.

The Euclidean distance may sound familiar to those who know Pythagoras's theorem, which measures the size of the longest side of a right-angled triangle.

Figure 5.3 Euclidean and Manhattan distances.



The Euclidean distance, with length 7.07, is represented by the diagonal line. The other highlighted line is the Manhattan distance, of length 10.

There are also other attribute types that are neither quantitative nor quali- tative, but are often encountered in data mining. These attribute types, here termed “non-conventional’, include:

- biological sequences
- time series
- images
- sound
- video.

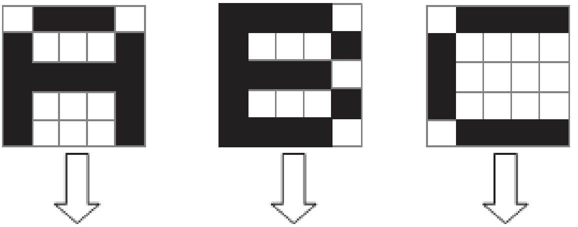
All these non-conventional attribute types can be converted into quantitative or qualitative types [20]. Among these non-conventional attribute types, the most common are sequences (text, biological and time series) and images.

Distance Measures for Non-conventional Attributes

The Hamming distance can be used for sequences of values and these values are usually characters or binary values. A binary value (or binary number) is either 1 or 0, meaning true or false, in general. The Hamming distance is the number of positions at which the corresponding characters or symbols in the two strings are different.

For example, the Hamming distance between the “Tom” and “Tim” is 1.

To calculate the distance between images, two distinct approaches can be used. In the first, features associated with the application can be extracted from the images. For example, for a face recognition task, the distance between the eyes can be extracted. Ultimately, each image is represented by a vector of real numbers, where each element corresponds to one particular feature.



0	1	1	1	0
1	0	0	0	1
1	1	1	1	1
1	0	0	0	1
1	0	0	0	1

1	1	1	1	0
1	0	0	0	1
1	1	1	1	0
1	0	0	0	1
1	1	1	1	0

0	1	1	1	1
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
0	1	1	1	1

	1st row					2nd row					3rd row					4th row					5th row				
A	0	1	1	1	0	1	0	0	0	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	1
B	1	1	1	1	0	1	0	0	0	1	1	1	1	1	0	1	0	0	0	1	1	1	1	1	0
C	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	1	1

Figure 5.5 Transformation of images of size 5 × 5 pixels into matrices and vectors.