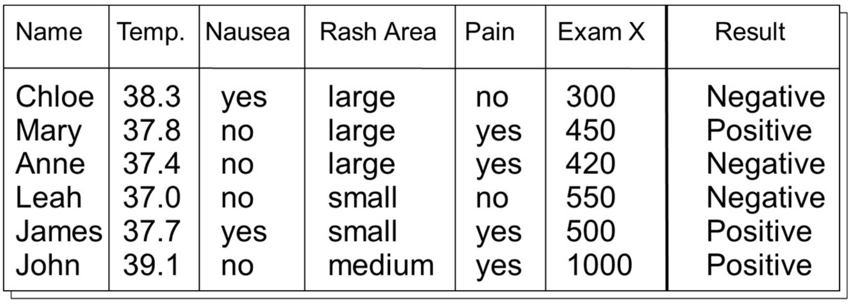
**University of Southern Denmark**
**IMADA**
Ricardo Campello

# DM583: Data Mining

## Exercise 2: Data Representation

### Exercise 2-1 Variable Types

a) Identify the types of variables in the above example assuming that Exam X is measuring the amount of certain cells per unit of volume and cannot be fractional. NOTE: Consider variable types according to the data mining taxonomy discussed in the course, NOT according to variable types in any particular programming language.



| Name | Temp. | Nausea | Rash Area | Pain | Exam X | Result |
|---|---|---|---|---|---|---|
| Chloe | 38.3 | yes | large | no | 300 | Negative |
| Mary | 37.8 | no | large | yes | 450 | Positive |
| Anne | 37.4 | no | large | yes | 420 | Negative |
| Leah | 37.0 | no | small | no | 550 | Negative |
| James | 37.7 | yes | small | yes | 500 | Positive |
| John | 39.1 | no | medium | yes | 1000 | Positive |

b) Identify appropriate types for the following variables:

- Number of certain blood cells in a blood sample
- The relative rank of each student of a class with respect to their final grades
- Length of an object as measured to any desired precision
- The hair color of an individual
- The postal code of an individual

Hint: think not only of the values that they possibly take, but also the operations that make sense for each of them.

### Exercise 2-2 Distances for Numerical Data

(a) Manually compute the Euclidean distance between every pair of data observations, $\mathbf{x}_i = [x_{i1}\ x_{i2}\ ...\ x_{i4}]$ and $\mathbf{x}_j = [x_{j1}\ x_{j2}\ ...\ x_{j4}]$, each of which is described by 4 numerical variables, as shown in the dataset below, then check your results by recomputing the distances using function `dist()` with argument `method = "euclidean"` in R:

|  | Variable 1 | Variable 2 | Variable 3 | Variable 4 |
|---|---|---|---|---|
| $x_1$ | 2 | -1 | 1 | 0 |
| $x_2$ | 7 | 0 | -4 | 8 |
| $x_3$ | 4 | 3 | 5 | 2 |
| $x_4$ | 5 | 10 | -1 | 5 |

(b) Repeat item (a) above now using Manhattan rather than Euclidean distance.

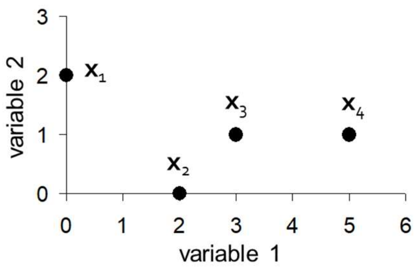
(c) Given a fixed reference point $\mathbf{x}_1$ on the plane, and another point $\mathbf{x}_2$ at a distance d($\mathbf{x}_1$,$\mathbf{x}_2$) from $\mathbf{x}_1$, it is easy to see that the set of all points that are at this very same distance from $\mathbf{x}_1$ form a circle with radius d($\mathbf{x}_1$,$\mathbf{x}_2$) centred at $\mathbf{x}_1$ if d is the **Euclidean** distance. What are the shapes formed if d is: **Manhattan**? **Suprema**?

(d) Given two records (i.e., data objects or observations) **p** and **q**, each of which is described by n=6 numerical variables, as follows:

$$\mathbf{p} = [1 \quad 2 \quad -3 \quad 2 \quad 0 \quad 8]$$
$$\mathbf{q} = [0 \quad 6 \quad 2 \quad -1 \quad 2 \quad 5]$$

Compute the Euclidean, Manhattan, and Suprema distances: (i) manually; (ii) using R

(e) Given a dataset with four data points (i.e., data objects or observations), each of which is described by n=2 numerical variables, as follows:



Compute the distance matrices for this dataset using Euclidean, Manhattan, and Suprema distances: (i) Manually (at least one non-diagonal entry of each matrix); and (b) Using R (the whole matrices)

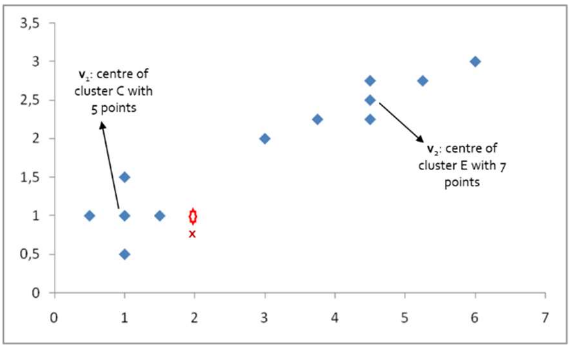
(f) Consider the *Ionosphere Radar Data*, which can be found online, e.g., from the archived UCI repository:

- Dataset description at: https://archive.ics.uci.edu/ml/datasets/Ionosphere
- Dataset as a CSV file (no variable names/header, "." as decimal point, and "," as separator) at: https://archive.ics.uci.edu/ml/machine-learning-databases/ionosphere/ionosphere.data

This dataset contains 351 records (rows) described by 34 numeric variables (columns). The 35th column is a categorical class label ("good" or "bad"). You are asked the following: read the file into a data frame in R, convert it to a numeric matrix getting rid of the class label (last column), then compute the Minkowski distances corresponding to the first 10 records (rows) of the remaining data matrix (with 34 columns), using the Minkowski parameter set to p = 1, p = 2, and p = ∞.

(g) Consider the following toy example seen in our lecture slides, involving a 2-dimensional dataset with two natural clusters of points (in blue diamonds), namely, cluster *C* (with 5 points at the bottom left) and cluster

*E* (with 7 points, at the top right), as well as an arbitrary reference point **x** (red circle):



```
C <- matrix(c(0.5, 1, 1, 0.5, 1, 1, 1, 1.5, 1.5, 1), 5, 2, byrow = TRUE) # Cluster C
v1 <- c(mean(C[,1]),mean(C[,2])) # Compute Centre of Cluster C (2-dimensional mean)
cov_C <- cov(C) # Compute Covariance Matrix of Cluster C
x <- c(2,1) # Point x
dxv1_sq <- mahalanobis(x, v1, cov_C) # [SQUARED] Mahalanobis dist. from x to v1
dxv1 <- sqrt(dxv1_sq) # Mahalanobis dist. from x to v1
```

With the R code above, we can compute the Mahalanobis distance from **x** = [2 1] to $\mathbf{v}_1$ = [1 1] as the centre of cluster *C*. The resulting distance is d(**x**,$\mathbf{v}_1$) = 2.82, or d(**x**,$\mathbf{v}_1$)$^2$ = 8. If the covariance matrix of cluster *E* is:

$$\text{Cov\_E} = \begin{bmatrix} 0.80 & 0.27 \\ 0.27 & 0.11 \end{bmatrix}$$

and the centre is $\mathbf{v}_2$ = [4.5 2.5], compute de distance from **x** to $\mathbf{v}_2$, d(**x**,$\mathbf{v}_2$), with the aid of function `mahalanobis()` in R. The result of the exercise above should be d(**x**,$\mathbf{v}_2$) = 5.53, or d(**x**,$\mathbf{v}_2$)$^2$ = 30.62. Notice that, in this case, **x** is much closer to Cluster *C* than it is to Cluster *E*. Repeat the computations using the formula directly, rather than the `mahalanobis()` function. **Note**: This function in R returns the *squared* Mahalanobis distance.

Now, assume we moved point x slightly to position [2 1.5]. Re-compute distances d(**x**,$\mathbf{v}_1$) and d(**x**,$\mathbf{v}_2$), noticing that only point **x** and the Mahalanobis distances themselves need to be re-computed, nothing else. The result should be d(**x**,$\mathbf{v}_1$) = 3.16 (d(**x**,$\mathbf{v}_1$)$^2$ = 10) and d(**x**,$\mathbf{v}_2$) = 3.01 (d(**x**,$\mathbf{v}_2$)$^2$ = 9.10). That is, **x** is now closer to $\mathbf{v}_2$ (Ellipsoidal Cluster *E*) than to $\mathbf{v}_1$ (Spherical Cluster *C*).

**Notes and Hints**:

When using the formula, you can still possibly have the aid of R to perform algebraic operations:

- $$d(\mathbf{x},\mathbf{v}) = \sqrt{(\mathbf{x}-\mathbf{v})^T Cov(\mathbf{X})^{-1}(\mathbf{x}-\mathbf{v})}$$

Note that **X** in the equation above should be the subset (cluster) of interest, whereas Cov(**X**) is its covariance matrix, with inverse Cov(**X**)$^{-1}$. The inverse of a matrix, when it exists, can be computed in R, e.g., using command `solve()`.You can create arrays of any arbitrary dimensions, including 1-dimensional *row vectors* or *column vectors* (matrices with a single column or row) using the command `array()` in R. You can transpose a matrix using the `t()` command.

(h) Compute the Pearson and Spearman correlations between the following two (7-dimensional) numeric observations: **p** = [10 -3 0 4 1 0 3] and **q** = [1 1 4 -2 3 -1 4]. Do it both manually as well as in R.

(i) Compute the Cosine similarity between the following records, each of which is described by 8 numeric variables: **p** = [1 0 0 4 1 0 0 3] and **q** = [0 5 0 2 3 1 0 4]. Do it in R but explicitly applying the formula, rather than using any built-in function.

### Exercise 2-3 Proximity Measures for Categorical Data

(a) Consider the following dataset with 4 data objects described by 10 binary *categorical* variables each.

$$\mathbf{X}=\begin{array}{c}\mathbf{x}_1\\ \mathbf{x}_2\\ \mathbf{x}_3\\ \mathbf{x}_4\end{array}\begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&1\\1&0&0&1&1&0&1&0&1&1\\0&1&0&0&1&1&0&0&1&0\end{bmatrix}$$

Compute the Simple Matching Coefficient (SMC) and the Jaccard coefficient for pairs of observations in the dataset above (manually and in R).

(b) Discuss what conditions are required for SMC to reach its extreme values (0 and 1). Repeat for Jaccard.

(c) Suppose that observations are students, and each binary categorical variable indicates that a students is ("1") or not ("0") enrolled in a subject (course). There are n=20 variables (courses). You want to assess similarity between students based on the courses that they choose. Should you use SMC or Jaccard?

(d) Repeat item (c) above now assuming that the 20 variables are questions of a questionnaire that the students answer as TRUE ("1") or FALSE ("0"), and we want to assess similarity of students based on their answers to the same questions.

(e) Suppose we computed the *similarity* between two observations described by categorical variables and obtained SWC = 6/10 and Jaccard = 1/3. Convert these proximity values from similarity to *dissimilarity*.

### Exercise 2-4 Categorical to Numeric Conversion

(a) Consider a variable Size that takes on categorical ordinal values {Very Small, Small, Medium, Large, Very Large, Extra Large}, where we know that their semantic respect their intended relative orders. Without any further information about the application domain or context, propose a simple yet suitable conversion scheme to represent this ordinal variable as a numeric variable instead.

(b) Now consider a nominal categorical value Political_Affiliation that can take on values {Party A, Party B, Party C}. Show how this variable can be replaced by a numerical representation using a one-hot encoding scheme.