3. Building data – Dealing with variables
3.1 Preparation with variables and their values

Not only binary encoding (for the nominal), but some other transforms may be useful. Such are ”traditional” statistical transforms as taking logarithm on values of a variable to transform it into quite linear.

Having described the types of variables, we now turn our attention to types of actions taken to prepare variables to some of the problems that need to be addressed.

Clearly it is important to have enough representative data from which to draw any conclusions about what need to be done and to execute data mining. We need enough data to be able to construct representative models for, e.g., classification.

The data items, cases or instances are represented as vectors **x** and **y** that are rows of two-dimensional data matrix (array or table) **D**. There are *p* variables or columns in **D**. The number of the rows is *n*. These are called cases, instances, measurements or observations.

Usually *n* is greater than *p*. Sometimes *p* can also be a large number when good examples are genetical data from bioinformatics and also a text document in information retrieval, where variables are typically (relative) frequencies of semantically meaningful words such as some nouns, adjectives or verbs.

$$\mathbf{D} = \begin{pmatrix} x_{11} & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & x_{ij} & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & x_{np} \end{pmatrix}$$

For a while, let us use Vertigo data from Table 2.1 as an example.

Variables have to be capable to represent phenomena measured. Commonly, domain experts are responsible for these matters. For instance, the ordinal variables in Table 2.1 include 4 to 8 values such as {0,..,5}≡{no attacks, 1-15 s, >15 s - 5 min, >5 min – 4 h, >4 – 24 h, 1-5 days} for variable ’duration of attack’ or {0,..,3} ≡{no handicap, slight handicap, moderate handicap, severe handicap} for ’severity of tinnitus’. The questions were carefully designed and considered for the patient inquiry form by the experienced medical experts. However, a data miner has to have at least some understanding about them and communicate actively with domain experts to understand underlying phenomena.

Removing variables

The basic information about a variable comprises the number of distinct values (if needed, for reals these can first be discretized into number intervals) and the frequency count of each distinct value. From this information it is easy to determine whether there are missing values and the distribution of different values. If there is only a single constant value, the variable is removed. We may say that this way the variable included no information in the sense that it did not aid to separate patients to different disease classes or classes in general.

Vertigo data set also contained many such original variables that consisted of no known values, they were entirely missing. In Vertigo data set there were originally such variables as ’lues serology’ (syphilis) for which 100% of values were missing or ’decibel levels of 50% speech discrimination for right and left ear’ for which 81% of values were missing. Such variables were naturally omitted.

How many missing values in per cent are a tolerable number for a data set to be modeled? This depends on the properties of a data set. This could be sometimes only 5% or 10% for small data sets, say, less than 300 instances (also depends on classes), but in some ”favorable” situations they could be 20%, 30% or exceptionally even more.

In the vertigo data set there was a single exceptional variable [14] ’hearing loss type’ that was missing in more than 60% of cases. Nevertheless, the variable could not be removed since it was in the group of the five most important variables of the whole data set. It may be called diagnostic variable in the medical context, i.e., which an otologist (oto=ear) making a decision had to know to be able to infer whether the patient had or had not an otological disorder called ’sudden deafness’, one of 6 to 10 disease classes in question.

Notwithstanding very many missing values, we were able to include this variable, since those missing values appeared for patients who have those other diseases or disorders than ’sudden deafness’. As a matter of fact, this indicated the reason of missing values. When most patients had other diseases, the otologists did not investigate this variable at all for most of them. At the same time, when the values were missing, this did no harm, because the current variable was obviously essentially less important for those other diseases.

This extraordinary variable of the Vertigo data set indicated that its values were not missing *randomly* but on purpose, since often the physician had deemed not to need this piece of information and was right as seen later. Such an approach is possible, since as an experienced expert the physician had some hypothesis from patients with specific symptoms, and typically the hypothesis hit the final best matching diagnosis.

Often variable values are not missing randomly, but there is some perhaps rational reason for the missing. This is useful and sometimes even crucial to know and to understand as was in Vertigo data set.

Data may be sparse, in other words, there are relatively frequent missing data in it. Then some variables are removed, but for some of them missing values can be estimated on the basis of known data. This is called *imputation*. While imputing incomplete data, the data miner has to be careful not to create artificial, distorting properties to a data set.

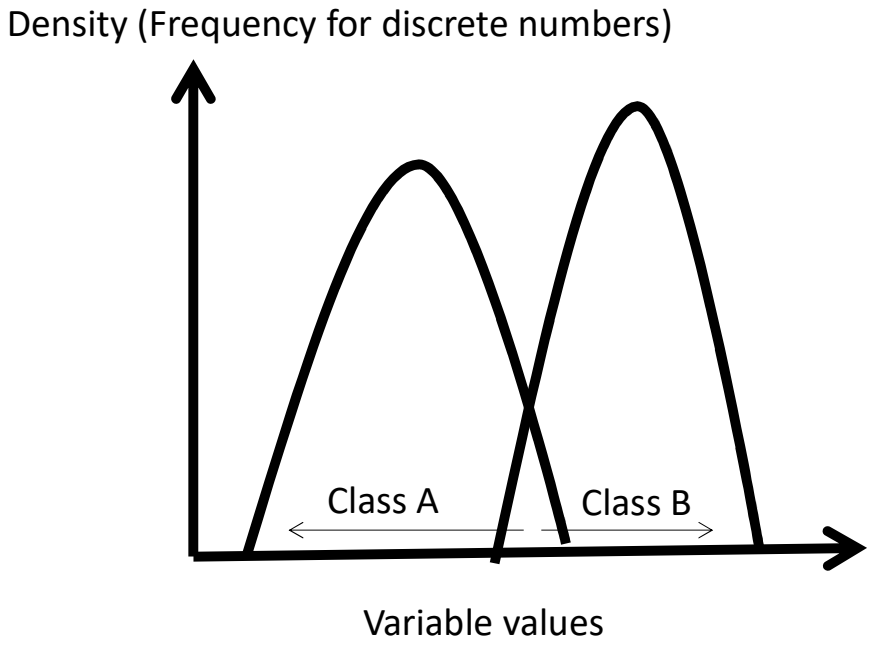
Imputation

To impute missing values in a *training* or *learning set*, i.e., subset of a data set, that is used for the construction of a model, we usually know the *class labels* of the training cases. Thus this information is best to utilize and to impute missing values class by class for each variable including missing values. This is essential, since the values of a variable present in the different classes are more or less different. This is natural, since if there were no difference, such a variable would be useless for separating different classes. See Fig. 3.1.

For real applications of machine learning algorithms, a new case due to be classified with the model built, if imputation is necessary, an imputed value has to be computed from the whole training data of the variable in question.

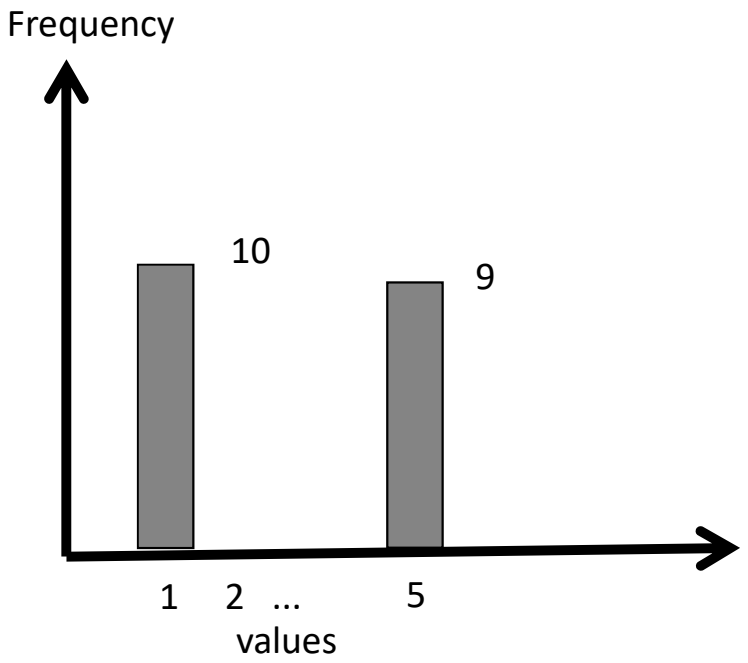
If we only compute and test models, the above approach is not, of course, necessary, but we can impute ”freely” based on the entire data set.

Fig. 3.1. When the variable is important for the separation of data classes, i.e., for classification, its values have at least somewhat different distributions for different classes. In the figure, the variable would be very useful if a slight part of values only overlapped along with the two classes



Imputation by simple methods contains the use of central values for missing variables inside each class. This process depends on the type of a variable. For the nominal, mode as the most frequent value is used. This also concerns binary variables. (Mode can be used for any variable type.) For ordinal variables we use median, the centermost value. For other quantitative variables we can use their medians and means, but statistically median is seen slightly better although this is a quite theoretical approach. However, their difference may occasionally be significant. See Fig. 3.2. For data mining purpose, median is ”more reliable” for imputation, since it represents the value that really exists in the distribution. Fig. 3.3 shows how an *outlier* affects the mean, but not the median.

Fig. 3.2. There are 10 values of 1 and 9 values of 5 for the current hypothetical variable. The mean is (10*1+9*5)/19 ≈ 2.89. The median is 1. The mean would not be the very best estimate here for a missing value, because it would be non-existent in the actual distribution and it would not be even ”close” to the actual values, but in a ”gap”. Thus, median is better here. It is always a value present in the data.



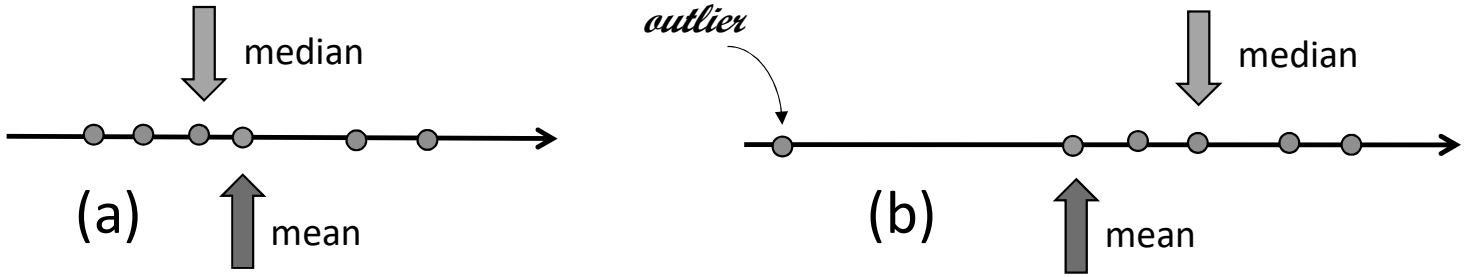


Fig. 3.3. Median of data set {*x*(1*)*, *x*(1), …, *x*(*N*)} is useful. Note its robustness property which manifests as no change even in the presence of one or sometimes even more outliers. (a) A ”normal”, nice situation and (b) with an outlier that distorts the mean. In fact, the median is the solution of the following optimization problem:

$$\min_{i=1,..,N} \sum_{k=1}^{N} \left| x(k) - x(i) \right| = \sum_{k=1}^{N} \left| x(k) - median \right|$$

Those central values above are appropriate for imputation if there are only relatively few missing values in each class. If a variable within some class does contain very many missing values, its mode, mean or median would produce a constant imputed value for many cases that is less ideal; for instance, a half of all values of the variable of a class is a constant.

The weakness of the preceding way is the use of a constant that makes data instances ”more similar” to each other. Thefore, it may be better to generate imputed values in more ”intelligent” ways, for instance, with regression analysis. In Fig. 3.4 there is an example of three variables of car data for which linear regression was computed.

If we know a value of predictor variable *X* and have computed a regression model, we can predict value for *Y* for imputation.

The linear regression technique involves discovering *joint variability* of the two variables *Y* and *X* using this to determine which values of the predicted variable match values of the predictor variable. Joint variability, the measure of the way one variable varies as another varies, allows the prediction with the linear regression model computed as usual, where the means of *x* and *y* are used to calculate coefficient *a*.

$$Y = a + bX$$

$$b = \frac{n\sum_{i=1}^{n} x(i)y(i) - \sum_{i=1}^{n} x(i)\sum_{i=1}^{n} y(i)}{n\sum_{i=1}^{n} x(i)^2 - \left(\sum_{i=1}^{n} x(i)\right)^2}$$

$$a = \bar{y} - b\bar{x}$$

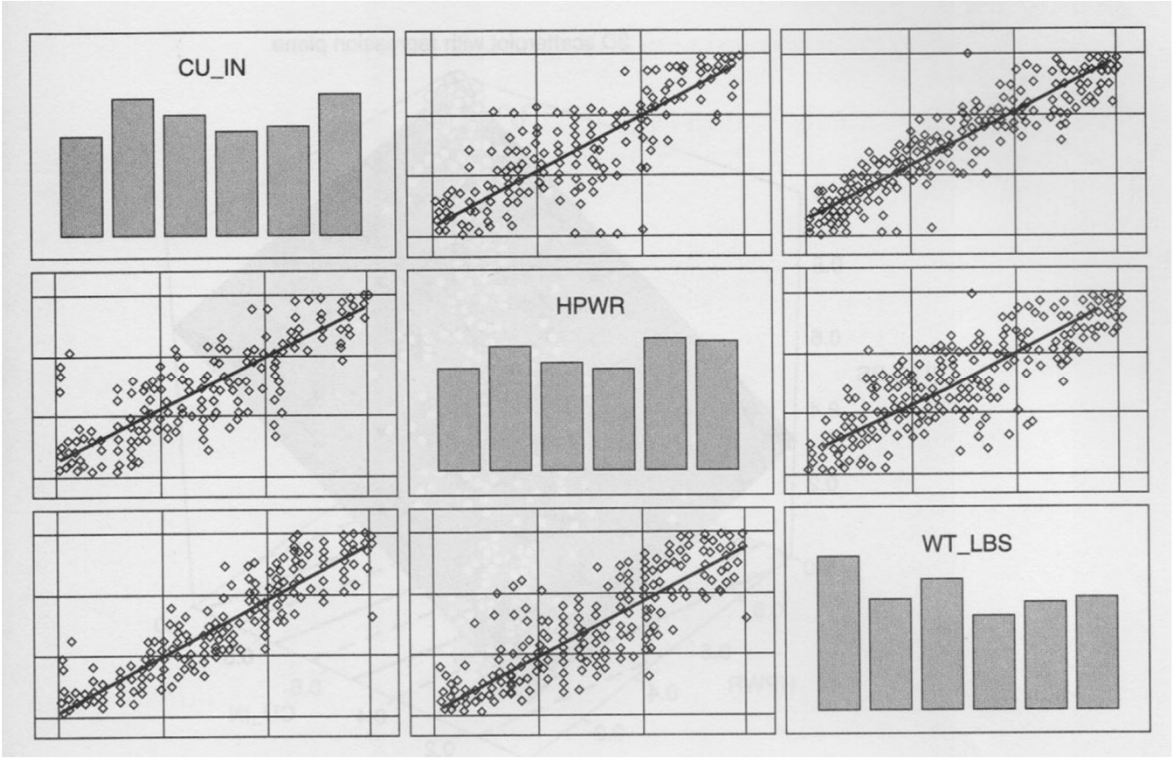


Fig. 3.4. There are three variables CU_IN (cubic inches), HPWR (horsepower) and WT_LBS (weight). The columns correspond to the predictor and the rows the predicted variable. The bar charts show the distributions of the variables.

Naturally, more complicated relations of three or more variables may be needed. Then we may use multiple linear regression as in Fig. 3.5.

Linearity is, however, often oversimplification, because in nature nonlinearity prevails typically. Thus, it can be good to first test statistically whether there are clearly nonlinear relations between variables. One of the easiest ways is to try nonlinear regressions and see if the fit is improved as the nonlinearity of the expression increases. This is not foolproof, because nonlinearities may be complicated.

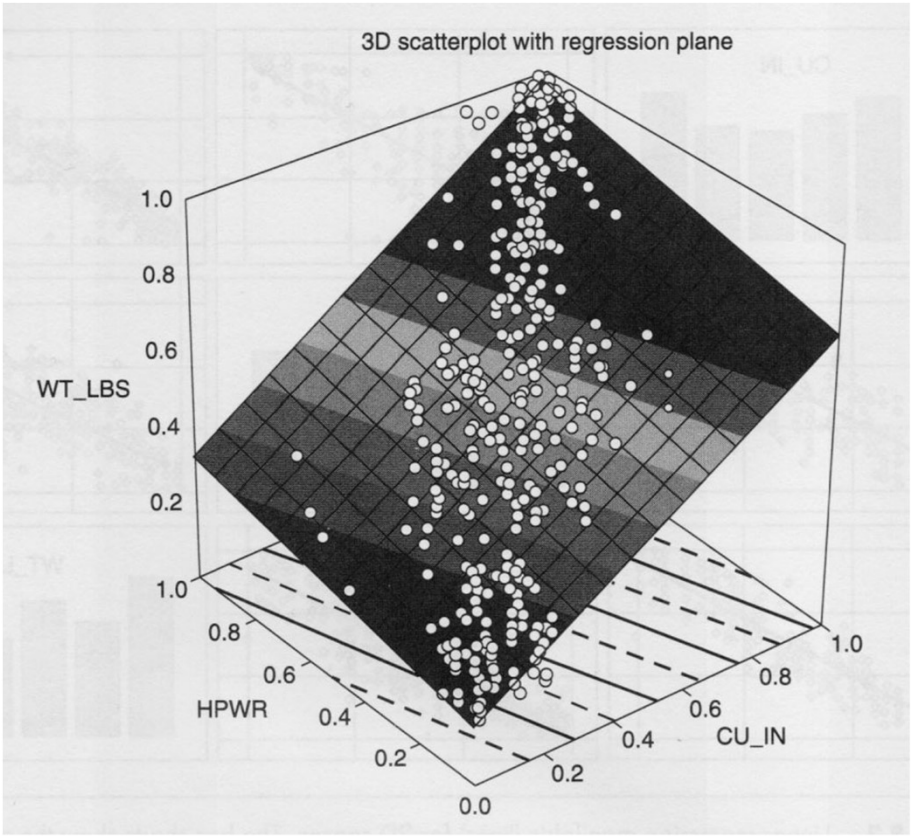


Fig. 3.5. Three-dimensional multiple linear regression. The manifold is a flat plane. The variables are those from the preceding figure.

Another imputation approach is to apply the distribution of the values of a variable and randomly take imputed values according to the probabilities defined by the distribution. For example, if the distribution were such as those in Fig. 3.1, an imputed value at the top of a curve would be more probable than another from its sides.

If there are a very small number of missing values for a variable, even a uniform distribution of values (each value equally probable) may sometimes be used and these ”guessed” values may serve well. For instance, we noticed that for a medical data set of female urinary incontinence there were some variables for which this approach worked well for missing values of a couple of diseases classes.[5]

[5]J. Laurikkala et al., Analysis of the imputed female urinary incontinence data for the evaluation of expert system parameters, Computers in Biology and Medicine, 31, 239-257, 2001

The reason was that sometimes a few variables were not important for the certain diseases. For this reason, physicians had not always investigated those variables, i.e., the reason for missing values was not random at all.

Let us look at the above-mentioned data set in Table 3.1 which included the data of 594 patients. In Table 3.1 the data set was divided into two parts: the main data (training set) of 529 patients and the test set of 65 patients. The latter was randomly taken from the whole data, but following approximately the class distribution of four disease classes and the fifth class of the normal (no disease). That kind of division is often used for training and testing in machine learning, and this theme will be considered subsequently.

Table 3.1

Variables for diagnosis of female urinary incontinence: numbers of missing values.
* removed

Variable	Values	Main *n*	%	Testing *n*	%
UVA	No, yes	1	0.2	0	0
US	Low 0–6, high 7–20	144	27.2	0	0
PVR	Normal 0–100, high >100	108	20.4	2	3.1
PMU	Low 0.0–0.5, high 0.6-1.0	335	63.3	27	41.5
CYM	Normal, abnormal	111	21.0	3	4.6
PTR	Abnormal 0–89%, normal ≥90%	183	34.6	14	21.5
MUCP	Negative ≤0, positive >0	127	24.0	13	20
SS	No, yes	116	21.9	2	3.1
UVJ	Normal, abnormal	191	36.1	2	3.1
UF*	Abnormal <20, normal ≥20	513	97.0	59	90.8
CYP*	Normal, abnormal	478	90.4	56	86.2
SSY	No, yes	2	0.4	1	1.5
CLU	No, yes	0	0	1	1.5
DV	No, yes	0	0	1	1.5
USY	No, yes	3	0.6	1	1.5
Age*	Integer	7	1.3	0	0
Total		2319	27.4	182	17.5

Three variables included great numbers of missing values. Two variables were left out, UF and CYP, which included very high numbers of missing values. In addition, variable ’age’ was also removed (obviously the physicians did not see it important).

Thereafter, the highest numbers of missing values were for PMU, PTR and UVJ. Imputation was finally also done for these despite many missing values. This exceptional situation was possible, since these variables were known well by the physicians and obviously the values were not missing randomly, but mostly in such classes where their missing did not do harm for diagnosing. (PMU was really exceptional, even questionable. Thus, it can be an example better not to imitate.)

Nearest neighbor estimators which are later viewed in detail can be efficient for imputation. It depends on the assumption that representative near neighbors can be found despite the fact that one or more dimensional values are missing. Nearest neighbors or cases are searched for in the variable space on the basis of some distance measure, e.g., *Euclidean distance*.

For example, let us assume that there is a case x=(1,1,?), where ? represents the missing value, and its nearest neighbor in the data set is a=(1,0.9,2) according to some distance measure. We form the subspace of the first two variables in Fig. 3.6 and then impute the missing value with 2 of the third variable.

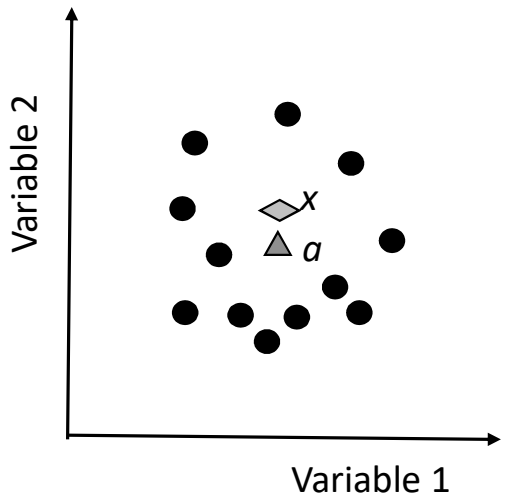


Fig. 3.6. A nearest neighbor searching for the missing value imputation. The nearest neighbor *a* gives an estimate. Here we assumed that the nearest neighbor of *x* was searched for from a subset where the values of Variables 1 and 2 of the cases used were known.

Expectation-maximum (EM) algorithm is a sophisticated method.

Also, machine learning methods such as neural networks have been applied efficiently to imputation.[6] The target of this study was to classify appendicitis patients. [7] In addition, nearest neihgbor searching, variable means and even random values were utilized. In this special application, differences of classification results after imputation with various methods were small, but there was virtually only one variable imputed (leucosyte counts, 18.9% missing), because for the other variables there were very few missing values (9, 4 or less). Thus, this was a simple and successful application as to imputation. There were 1333 patients with 14 variables altogether.

[6]E. Pesonen, M. Eskelinen and M. Juhola, Treatment of missing data values in a neural network based on decision support system for acute abdominal pain, Artificial Intelligence in Medicine, 13, 139-146, 1998.

[7]E. Pesonen, J. Ikonen, M. Juhola and M. Eskelinen, Parameters for a knowledge base for acute appendicitis, Methods of Information in Medicine, 33, 220-226, 1994.

Dimensionality reduction

For large data sets where there are perhaps thousands of variables it may be necessary to reduce their number. There can be two reasons for this. First, some variables are quite meaningless or useless, say, for classification. We can explore their ”uselfulness”. For instance, a constant or dead variable is useless. A remarklable number of missing values may lead to the removal of some variables.

For information retrieval, words that occur rarely or very frequently are typically left out (relative word counts here as variables), because it is seen that they are not so capable to separate digital text documents from each other than those originating from the middle part of a frequency distribution of words.

In other words, if a certain string, word in its basic form of a natural language occurs rarely, in few documents only, it does not cover much information to separate a great number of different documents. On the other hand, if a word occurs within almost any document of the set, it is too frequent and does neither separate well different documents. The objective here is to search for fairly similar or dissimilar documents, i.e., to classify or cluster documents.

Second, a great number of variables are often as such the reason to reduce that number, because it may incur so great running times for classification task that the reducement of variables is needed.

There are different algorithms to select variables to keep important or useful variables and drop out poor ones. Another approach is to compute ”artificial” variables, in other words, to construct a new coordinate (new ”variables”) system through a mathematical transform. Instances or cases are then – in a favorable situation – located so that some of new variables are very useful and some other only slightly. The latter are then possible to leave out, i.e., to reduce dimensionality. Principal component analysis is such a technique.

We will return to these techniques in subsequent sections.

3.2 Basic computation techniques for data

Distance and similarity measures

There are several distance measures from which the best known is the *Euclidean distance*. The question is often how similar instances are or the opposite how dissimilar they are. Let us give instances or cases as vectors $\mathbf{x}=(x_1,\ldots, x_p)$ and $\mathbf{y}=(y_1,\ldots, y_p)$ where p is the number of the variables.

If $s(\mathbf{x},\mathbf{y})$ denotes *similarity*, computed somehow, with subsequent distance or similarity measures, and we define the maximum similarity to be equal to 1 (identical cases) and the minimum similarity equal to 0 (not alike at all), we are able to define *dissimilarity ds* with monotonically decreasing transformation.

Possible transformations are:

$$ds(\mathbf{x},\mathbf{y}) = 1 - s(\mathbf{x},\mathbf{y})$$

$$ds(\mathbf{x},\mathbf{y}) = \sqrt{2(1 - s(\mathbf{x},\mathbf{y})}$$

We may conceive that distance is quite the same as similarity-dissimilarity when we give exact rules how to compute these. Nevertheless, distance is equal to 0 for identical cases and more for more or less different cases.

Proximity is the general concept for all similarity and distance measures.

Metric is another important concept. A distance measure is metric when it satisfies the following conditions:

Metric

The conditions of the metric

(1) $d(\mathbf{x},\mathbf{y}) > 0$ if $\mathbf{x} \neq \mathbf{y}$ positivity

(2) $d(\mathbf{x},\mathbf{y}) = 0$ if $\mathbf{x} = \mathbf{y}$ reflexivity

(3) $d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x})$ symmetry

(4) $d(\mathbf{x},\mathbf{y}) \leq d(\mathbf{x},\mathbf{z}) + d(\mathbf{z},\mathbf{y})$ for all vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ triangle inequality

Euclidean distance is metric. (Note that the term metric is sometimes used ”wildly”. Even if in the name of a distance there were the word metric, it would not necessarily be true.)

If a distance measure is a metric, it produces well defined information about dissimilarity or similarity, distances, between measurements or cases.

The measure assumes some degree of *commensurability* between different variables in vectors **x**, **y** and **z**. For instance, we cannot use Euclidean distance with nominal variables as is possible for binary and quantitative variables. We cannot say what the Euclidean distance from the green apple would be from the yellow or red apple. Because equivalence relation merely is valid for them, we can only say that they are different if the color is the variable used.

The following distance measures can only be used for quantitative and binary variables (except Chi-square distance and Gower similarity).

3.3 Distance measures

Let us now define the *Manhattan* or *Block city* (or *Hamming*) distance, Euclidean distance and Tshebyschev distance also called L_1, L_2 and L_∞ metrics:

$$\text{Manhattan } d_1(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{p} |x_k - y_k| \qquad \text{Euclidean } d_2(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{k=1}^{p} (x_k - y_k)^2}$$

$$\text{Tshebyschev } d_3(\mathbf{x},\mathbf{y}) = \max_{\mathrm{k}} \left(|x_k - y_k| \right)$$

In general, these are called *Minkowski* distances or L_λ metric according to:

$$d_4(\mathbf{x},\mathbf{y}) = \left(\sum_{k=1}^{p} |x_k - y_k|^{\lambda} \right)^{1/\lambda}$$

There are other distance measures for quantitative variables. The preceding measures can be weighted which is useful if some variables are more important than others. Then the former have greater weights than those of the latter. (Appropriate weights have to be determined.)

$$\text{weighted Euclidean } d_{WE}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{k=1}^{p} w_k \left(x_k - y_k\right)^2}$$

Nonetheless, frequently the opposite transformation is made if some variables include an extensive scale, say, [1,100] and others small like [0,1]. Normalization is then often used to scale all, usually, into [0,1] or into the same scale in any case. This is made so that the variable of a more extensive scale would not dominate computation exceedingly. Normalization will subsequently be considered.

Let us use *Correlation coefficient* of the two vectors with variables x_k and y_k and their means occuring in the data set, when there are p variables.

$$r(\mathbf{x},\mathbf{y}) = \frac{\sum_{k=1}^{p}(x_k - \bar{x}_k)(y_k - \bar{y}_k)}{\sqrt{\sum_{k=1}^{p}(x_k - \bar{x}_k)^2 \sum_{i=1}^{p}(y_k - \bar{y}_k)^2}}$$

The correlation coefficient is a value from [-1,1]. A positive number indicates positive relationship between the cases. This is perfect if it is 1, but ”in nature” it hardly ever happens. It is the same for the opposite, -1. Here $\bar{x}_k = \bar{y}_k$ is the mean of variable *k* occuring in the (training) data set.

If there were 100 variables – for some strange reason - for height measures of an object and only one for its width, the 100 variables would dominate strongly results. We can discount the effect of 100 correlated variables by incorporating the covariance matrix in the distance definition. This leads to the *Mahalanobis* distance between two *p*-dimensional vectors as follows.

$$d_{MH}(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T \Sigma^{-1} (\mathbf{x}-\mathbf{y})}$$

Mahalanobis distance includes the inverse of $p \times p$ covariance matrix Σ that standardizes the data relative to Σ. *T* represents the transpose.

Covariance between two variables *X* and *Y* (an element in the matrix) for *n* items or cases is as follows with two components (variables) and their means.

$$\operatorname{cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}\left(x(i)-\bar{x}\right)\left(y(i)-\bar{y}\right)$$

In *Chi-square* distance function sum_k is the sum of all values for variable *k* occurring in the data set and $size_{\mathbf{x}}$ is the sum of all values in the vector **x**.

$$chi(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{p} \frac{1}{sum_k}\left(\frac{x_k}{size_{\mathbf{x}}} - \frac{y_k}{size_{\mathbf{y}}}\right)^2$$

Cosine measure is important in information retrieval, e.g., when documents are compared based on their (relative) word frequencies.

$$\mathrm{cm}(\mathbf{x},\mathbf{y}) = 1 - \cos(\mathbf{x},\mathbf{y}) = 1 - \frac{\sum_{k=1}^{p} x_k y_k}{\sqrt{\sum_{k=1}^{p} x_k^2 \sum_{k=1}^{p} y_k^2}} = 1 - \frac{\mathbf{x}\cdot\mathbf{y}}{|\mathbf{x}||\mathbf{y}|}$$

This was also given with vectors. (It is not metric, but this does not do harm, at least in information retrieval.) It takes advantage of cosine of the angle between vectors **x** and **y** in the *p*-dimensional variable space.

Similarity measures

The following is *Overlap similarity measure*:

$$s_1(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^{p} x_k y_k}{\min\left(\sum_{k=1}^{p} x_k^2, \sum_{k=1}^{p} y_k^2\right)}$$

Note that normally it is assumed vector entries to be nonnegative values here as well as for all distance measures. Also, we have to assume that no denominator is equal to zero. This is no distance measure, but similarity measure so that it is equal to 0 for vectors that

do not overlap, for example, **x**=(0,1,2) and **y**=(1,0,0), in other words, one of the two vectors always has a 0 in one of the two components.

If the vectors are identical, the value is 1.

However, a value of this measure may sometimes be ”illogically” even greater than 1.

Overlap similarity as well as the following s_2 to s_4 have their origin in measuring similarities between sets based on the intersection of the two sets. According to its name, the Overlap measure determines the degree to which the two sets overlap.

Similarity values are often used for document classification where numbers of semantically meaningful words or terms of a natural language are counted and used as variable values. Then irrelevant ”stop words” such as {a, an, the, and, or, ..} are first removed, because these do not tell about topics of documents.

Other similarity measures are Dice s_2, Jaccard s_3, and Cosine s_4 as follows:

$$s_2(\mathbf{x},\mathbf{y}) = \frac{2\sum_{k=1}^{p} x_k y_k}{\sum_{k=1}^{p} x_k^2 + \sum_{k=1}^{p} y_k^2} \quad s_3(\mathbf{x},\mathbf{y}) = \frac{\sum_{k=1}^{p} x_k y_k}{\sum_{k=1}^{p} x_k^2 + \sum_{k=1}^{p} y_k^2 - \sum_{k=1}^{p} x_k y_p} \quad s_4(\mathbf{x},\mathbf{y}) = \frac{\sum_{k=1}^{p} x_k y_k}{\sqrt{\sum_{k=1}^{p} x_k^2 \sum_{k=1}^{p} y_k^2}}$$

Cosine was here in the ”similarity” role, whereas previously in the ”distance” role.

Dice’s coefficient relates the overlap to the average size of the two sets together. Jaccard’s coefficient is used to measure the overlap of two sets as related to the whole set caused by the their union. The Cosine coefficient relates the overlap to the geometric mean of the two sets.

Gower similarity s_5 is:

$$s_5(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^{p} w_k u_k}{\sum_{k=1}^{p} w_k},$$

where $u_k = 1 - |x_k - y_k| / R_k$, $R_k = \max_k - \min_k$ for a quantitative variable and for a binary or nominal variable u_k is 1 if $x_k = y_k$ and 0 if $x_k \neq y_k$

Weight w_k is 1 if both values x_k and y_k are known. If one of them or both are unknown, it is 0. This is a new means to handle missing values where no imputation is performed. It is called ”pessimistic”, because we then assume the maximal dissimilarity for a variable, i.e., 0. R_k is a scaling or normalization value in which the maximum and minimum of kth variable of **x** and **y** are used.

3.4 Binary distance and similarity measures

Binary similarity is applied when ***all variables*** of data vectors are binary, in a way the simplest of all.

Table 3.2 A cross-classification of two binary variables

	k=1	k=0
j=1	$n_{1,1}$	$n_{1,0}$
j=0	$n_{0,1}$	$n_{0,0}$

For multivariate binary data we count the numbers of variables on which two objects take the same or different values. Consider Table 3.2 in which all p variables defined for objects j and k take values {0,1}. The entry $n_{1,1}$ for j=1 and k=1 denotes that there are $n_{1,1}$ variables such that j and k both have value 1.

Hamming distance counts how many pairs of the vector components, variables, are unequal.

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

Note that it is not necessary that the variables used here are binary. They can also be, for instance, characters or any symbols for which we use binary operation, equivalence relation. For example, Hamming distance for DNA strings ’ATCTT’ and ’ATGAA’ is 3. For binary variables or bit strings 010101 and 111011, it is 4.

Perhaps the most obvious similarity measure is the simple *Sokal and Michener matching coefficient* defined as

$$M(j,k) = \frac{n_{1,1} + n_{0,0}}{n_{1,1} + n_{1,0} + n_{0,1} + n_{0,0}}$$

the proportion of variables on which the objects have same value and where $n_{1,1} + n_{1,0} + n_{0,1} + n_{0,0} = p$, the total number of variables.

Sometimes, however, it is inappropriate to include (0,0) cell (or (1,1)), depending on the meaning of 0 and 1. For instance, if the variables

express the presence (1) or absence (0) of certain properties, we may not care about all the irrelevant properties had by neither object. For example, in vector representations of text documents it may not be relevant that two documents *do not* contain thousands of specific terms. This leads to the modification of the matching coefficient, *Jaccard coeffcient* for binary variables.

$$J(j,k) = \frac{n_{1,1}}{n_{1,1} + n_{1,0} + n_{0,1}}$$

Dice coefficient is also used for binary variables.

$$D(j,k) = \frac{2n_{1,1}}{n_{1,1} + n_{1,0} + n_{0,1}}$$

There are more, such as *Russel and Rao*, and *Sokal and Sneath*.

$$RR(j,k) = \frac{n_{1,1}}{n_{1,1} + n_{1,0} + n_{0,1} + n_{0,0}} \qquad SS(j,k) = \frac{n_{1,1}}{n_{1,1} + 2\left(n_{1,0} + n_{0,1}\right)}$$

Furthermore, there are *Kulzinsky and Rogers*, and *Tanimoto* similarity measures.

$$KR(j,k) = \frac{n_{1,1}}{n_{1,1} + n_{0,1}} \qquad T(j,k) = \frac{n_{1,1} + n_{0,0}}{n_{1,1} + n_{0,0} + 2\left(n_{1,0} + n_{0,1}\right)}$$

In addition, there are *Yule* and *binary correlation* measures.

$$Y(j,k) = \frac{n_{1,1}n_{0,0} - n_{1,0}n_{0,1}}{n_{1,1}n_{0,0} + n_{1,0}n_{0,1}}$$

$$BC(j,k) = \frac{n_{1,1}n_{0,0} + n_{1,0}n_{0,1}}{\sqrt{(n_{1,1} + n_{1,0})(n_{0,1} + n_{0,0})(n_{1,1} + n_{0,1})(n_{1,0} + n_{0,0})}}$$

There are dozens of other binary measures, but perhaps those described so far are the most interesting.[8]

[8] S.-S. Choi et al., A survey of binary similarity and distance measures, Systemics, Cybernetics and Informatics, 8(1), 43- 48, 2010

3.5 Distance measures of mixed variables

In reality there are every now and then data that include different types of variables called mixed types. Then distance measures have to be constructed based on the slightly different approach. *Heterogenous Euclidean-Overlap Metric* (*HEOM*) is defined as shown below.

$$HEOM(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{k=1}^{p} d_k\left(x_k, y_k\right)^2}$$

Distance d_k is calculated differently depending on the type (scale) of a variable. Below the second formula concerns quantitative and the third nominal variables. Since *HEOM* is a distance function, the variables are handled oppositely to similarity functions. The distance between the values of a nominal variable is 0 when the values are the same and 1 otherwise.

$$d_k(x_k, y_k) = \begin{cases} 1, \text{if } x_k \text{ or } y_k \text{ or both are missing} \\ \dfrac{|x_k - y_k|}{R_k}, \text{where } R_k = \max_k - \min_k \\ overlap(x_k, y_k) = \begin{cases} 0, \text{if } x_k = y_k \\ 1, \text{if } x_k \neq y_k \end{cases} \end{cases}$$

HEOM was named metric by D. Aha invented it.[9] Since we noticed that the reflexivity condition is not true for it (very easy to prove), it should actually be called pseudometric to be precise.[10] This property yields an odd conclusion that ”distance from a vector to itself is not equal to 0” provided that there are one or more missing values within it.

[9]D.W. Aha, Instance-based learning algorithms, Machine Learning, 6, 37-66, 1991.

[10]M. Juhola and J. Laurikkala, On metricity of two heterogenous measures in the presence of missing values, Artificial Intelligence Review, 28, 163-178, 2007

Nevertheless, this oddness was more a theoretical nuisance since *HEOM* works well by separating efficiently cases (vectors) differing from each other. The same also appears in the following distance function, *HVDM*.

Yet, there is also a practical specific weakness in these not being entirely metric. If there are missing values in data vectors, they are not appropriate for searching for duplicates (identical cases), since their distance is not equal to 0. For example, if we use symbol * to indicate a missing value and there are vectors (1,2,3,*) and (1,2,3,*), the distance of these is not 0 even if we might see them as duplicates.

Heterogeneous Value Difference Metric (*HVDM*) is defined as follows. The second formula is for quantitative variables and the third for nominal variables:

$$HVDM(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{k=1}^{p} d_k(x_k, y_k)^2}$$

$$d_k(x_k, y_k) = \begin{cases} 1, \text{if } x_k \text{ or } y_k \text{ or both are missing} \\ \dfrac{|x_k - y_k|}{4\sigma_k}, \text{where } \sigma_k \text{ is standard deviation of the } k\text{th variable} \\ \sqrt{\sum_{q=1}^{c} \left| \dfrac{N_{k,x,q}}{N_{k,x}} - \dfrac{N_{k,y,q}}{N_{k,y}} \right|^2}, \text{where } N_{k,x,q} \text{ is the number of cases in the data set with value } x_k \\ \text{and class } q, N_{k,x} \text{ is the number of cases in the data set that have value } x_k \text{ and } c \text{ denotes} \\ \text{the number of classes.} \end{cases}$$