6

Data preparation

Business intelligence systems and mathematical models for decision making can achieve accurate and effective results only when the input data are highly reliable. However, the data extracted from the available primary sources and gathered into a data mart may have several anomalies which analysts must identify and correct.

This chapter deals with the activities involved in the creation of a high quality dataset for subsequent use for business intelligence and data mining analysis. Several techniques can be employed to reach this goal: data validation, to identify and remove anomalies and inconsistencies; data integration and transformation, to improve the accuracy and efficiency of learning algorithms; data size reduction and discretization, to obtain a dataset with a lower number of attributes and records but which is as informative as the original dataset. For further readings on the subject and for basic concepts of descriptive statistics, see the notes at the end of Chapter 7.

6.1 Data validation

The quality of input data may prove unsatisfactory due to *incompleteness*, *noise* and *inconsistency*.

Incompleteness. Some records may contain missing values corresponding to one or more attributes, and there may be a variety of reasons for this. It may be that some data were not recorded at the source in a systematic way, or that they were not available when the transactions associated with a record took place. In other instances, data may be missing because of malfunctioning recording devices. It is also possible that some data were deliberately removed during previous stages of the gathering process because they were deemed

Business Intelligence: Data Mining and Optimization for Decision Making *Carlo Vercellis*
© 2009 John Wiley & Sons, Ltd. ISBN: 978-0-470-51138-1

incorrect. Incompleteness may also derive from a failure to transfer data from the operational databases to a data mart used for a specific business intelligence analysis.

Noise. Data may contain erroneous or anomalous values, which are usually referred to as *outliers*. Other possible causes of noise are to be sought in malfunctioning devices for data measurement, recording and transmission. The presence of data expressed in heterogeneous measurement units, which therefore require conversion, may in turn cause anomalies and inaccuracies.

Inconsistency. Sometimes data contain discrepancies due to changes in the coding system used for their representation, and therefore may appear inconsistent. For example, the coding of the products manufactured by a company may be subject to a revision taking effect on a given date, without the data recorded in previous periods being subject to the necessary transformations in order to adapt them to the revised encoding scheme.

The purpose of data validation techniques is to identify and implement corrective actions in case of incomplete and inconsistent data or data affected by noise.

6.1.1 Incomplete data

To partially correct incomplete data one may adopt several techniques.

Elimination. It is possible to discard all records for which the values of one or more attributes are missing. In the case of a supervised data mining analysis, it is essential to eliminate a record if the value of the target attribute is missing. A policy based on systematic elimination of records may be ineffective when the distribution of missing values varies in an irregular way across the different attributes, since one may run the risk of incurring a substantial loss of information.

Inspection. Alternatively, one may opt for an inspection of each missing value, carried out by experts in the application domain, in order to obtain recommendations on possible substitute values. Obviously, this approach suffers from a high degree of arbitrariness and subjectivity, and is rather burdensome and time-consuming for large datasets. On the other hand, experience indicates that it is one of the most accurate corrective actions if skilfully exercised.

Identification. As a third possibility, a conventional value might be used to encode and identify missing values, making it unnecessary to remove entire records from the given dataset. For example, for a continuous attribute that assumes only positive values it is possible to assign the value $\{-1\}$ to all missing data. By the same token, for a categorical attribute one might replace missing values with a new value that differs from all those assumed by the attribute.

Substitution. Several criteria exist for the automatic replacement of missing data, although most of them appear somehow arbitrary. For instance, missing values of an attribute may be replaced with the mean of the attribute calculated for the remaining observations. This technique can only be applied to numerical attributes, but it will clearly be ineffective in the case of an asymmetric distribution of values. In a supervised analysis it is also possible to replace missing values by calculating the mean of the attribute only for those records having the same target class. Finally, the maximum likelihood value, estimated using regression models or Bayesian methods, can be used as a replacement for missing values. However, estimate procedures can become rather complex and time-consuming for a large dataset with a high percentage of missing data.

6.1.2 Data affected by noise

The term *noise* refers to a random perturbation within the values of a numerical attribute, usually resulting in noticeable anomalies. First, the outliers in a dataset need to be identified, so that subsequently either they can be corrected and regularized or entire records containing them are eliminated. In this section we will describe a few simple techniques for identifying and regularizing data affected by noise, while in Chapter 7 we will describe in greater detail the tools from exploratory data analysis used to detect outliers.

The easiest way to identify outliers is based on the statistical concept of *dispersion*. The sample mean $\bar{\mu}_j$ and the sample variance $\bar{\sigma}_j^2$ of the numerical attribute $\mathbf{a}_j$ are calculated. If the attribute follows a distribution that is not too far from normal, the values falling outside an appropriate interval centered around the mean value $\bar{\mu}_j$ are identified as outliers, by virtue of the central limit theorem. More precisely, with a confidence of $100(1-\alpha)\%$ (approximately 96% for $\alpha = 0.05$) it is possible to consider as outliers those values that fall outside the interval

$$(\bar{\mu}_j - z_{\alpha/2}\bar{\sigma}_j, \bar{\mu}_j + z_{\alpha/2}\bar{\sigma}_j), \tag{6.1}$$

where $z_{\alpha/2}$ is the $\alpha/2$ quantile of the standard normal distribution. This technique is simple to use, although it has the drawback of relying on the critical assumption that the distribution of the values of the attribute is bell-shaped and roughly normal. However, by applying Chebyshev's theorem, described in Chapter 7, it is possible to obtain analogous bounds independent of the distribution, with intervals that are only slightly less stringent. Once the outliers have been identified, it is possible to correct them with values that are deemed more plausible, or to remove an entire record containing them.

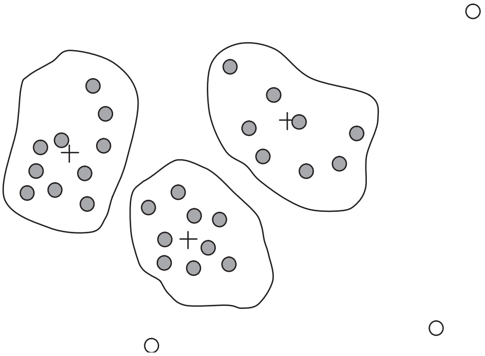

Figure 6.1 Identification of outliers using cluster analysis

An alternative technique, illustrated in Figure 6.1, is based on the distance between observations and the use of clustering methods. Once the clusters have been identified, representing sets of records having a mutual distance that is less than the distance from the records included in other groups, the observations that are not placed in any of the clusters are identified as outliers. Clustering techniques offer the advantage of simultaneously considering several attributes, while methods based on dispersion can only take into account each single attribute separately.

A variant of clustering methods, also based on the distances between the observations, detects the outliers through two parametric values, p and d, to be assigned by the user. An observation $\mathbf{x}_i$ is identified as an outlier if at least a percentage p of the observations in the dataset are found at a distance greater than d from $\mathbf{x}_i$.

The above techniques can be combined with the opinion of experts in order to identify actual outliers with respect to regular observations, even though these fall outside the intervals where regular records are expected to lie. In marketing applications, in particular, it is appropriate to consult with experts before adopting corrective measures in the case of anomalous observations.

Unlike the above methods, aimed at identifying and correcting each single anomaly, there exist also regularization techniques which automatically correct anomalous data. For example, simple or multiple regression models predict the value of the attribute $\mathbf{a}_j$ that one wishes to regularize based on other variables existing in the dataset. Once the regression model has been developed, and the corresponding confidence interval around the prediction curve has been calculated, it is possible to substitute the value computed along the prediction curve for the values of the attribute $\mathbf{a}_j$ that fall outside the interval.

A further automatic regularization technique, described in Section 6.3.4, relies on data discretization and grouping based on the proximity of the values of the attribute $\mathbf{a}_j$.

6.2 Data transformation

In most data mining analyses it is appropriate to apply a few transformations to the dataset in order to improve the accuracy of the learning models subsequently developed. Indeed, outlier correction techniques are examples of transformations of the original data that facilitate subsequent learning phases. The principal component method, described in Section 6.3.3, can also be regarded as a data transformation process.

6.2.1 Standardization

Most learning models benefit from a preventive *standardization* of the data, also called *normalization*. The most popular standardization techniques include the *decimal scaling* method, the *min-max* method and the *z-index* method.

Decimal scaling. Decimal scaling is based on the transformation

$$x'_{ij} = \frac{x_{ij}}{10^h}, \tag{6.2}$$

where h is a given parameter which determines the scaling intensity. In practice, decimal scaling corresponds to shifting the decimal point by h positions toward the left. In general, h is fixed at a value that gives transformed values in the range $[-1, 1]$.

Min-max. Min-max standardization is achieved through the transformation

$$x'_{ij} = \frac{x_{ij} - x_{\min,j}}{x_{\max,j} - x_{\min,j}}(x'_{\max,j} - x'_{\min,j}) + x'_{\min,j}, \tag{6.3}$$

where

$$x_{\min,j} = \min_i \; x_{ij}, \quad x_{\max,j} = \max_i \; x_{ij}, \tag{6.4}$$

are the minimum and maximum values of the attribute $\mathbf{a}_j$ before transformation, while $x'_{\min,j}$ and $x'_{\max,j}$ are the minimum and maximum values that we wish to obtain after transformation. In general, the extreme values of the range are defined so that $x'_{\min,j} = -1$ and $x'_{\max,j} = 1$ or $x'_{\min,j} = 0$ and $x'_{\max,j} = 1$.

***z*-index.** z-index based standardization uses the transformation

$$x'_{ij} = \frac{x_{ij} - \bar{\mu}_j}{\bar{\sigma}_j}, \tag{6.5}$$

where $\bar{\mu}_j$ and $\bar{\sigma}_j$ are respectively the sample mean and sample standard deviation of the attribute $\mathbf{a}_j$. If the distribution of values of the attribute $\mathbf{a}_j$ is roughly normal, the z-index based transformation generates values that are almost certainly within the range $(-3, 3)$.

6.2.2 Feature extraction

The aim of standardization techniques is to replace the values of an attribute with values obtained through an appropriate transformation. However, there are situations in which more complex transformations are used to generate new attributes that represent a set of additional columns in the matrix **X** representing the dataset $\mathcal{D}$. Transformations of this kind are usually referred to as *feature extraction*. For example, suppose that a set of attributes indicate the spending of each customer over consecutive time intervals. It is then possible to define new variables capable of capturing the trends in the data through differences or ratios between spending amounts of contiguous periods.

In other instances, the transformations may take even more complex forms, such as *Fourier transforms*, *wavelets* and *kernel* functions. The use of such methods will be explained within the classification methods called *support vector machines* in Chapter 10.

Attribute extraction may also consist of the creation of new variables that summarize within themselves the relevant information contained in a subset of the original attributes. For example, in the context of image recognition one is often interested in identifying the existence of a face within a digitalized photograph. There are different indicators intended for the synthesis of each piece of information contained in a group of adjacent pixels, which make it easier for classification algorithms to detect faces.

6.3 Data reduction

When dealing with a small dataset, the transformations described above are usually adequate to prepare input data for a data mining analysis. However, when facing a large dataset it is also appropriate to reduce its size, in order to make learning algorithms more efficient, without sacrificing the quality of the results obtained.

There are three main criteria to determine whether a data reduction technique should be used: *efficiency*, *accuracy* and *simplicity* of the models generated.

Efficiency. The application of learning algorithms to a dataset smaller than the original one usually means a shorter computation time. If the complexity of the algorithm is a superlinear function, as is the case for most known methods, the improvement in efficiency resulting from a reduction in the dataset size may be dramatic. As described in Chapter 5, within the data mining process it is customary to run several alternative learning algorithms in order to identify the most accurate model. Therefore, a reduction in processing times allows the analyses to be carried out more quickly.

Accuracy. In most applications, the accuracy of the models generated represents a critical success factor, and it is therefore the main criterion followed in order to select one class of learning methods over another. As a consequence, data reduction techniques should not significantly compromise the accuracy of the model generated. As shown below, it may also be the case that some data reduction techniques, based on attribute selection, will lead to models with a higher generalization capability on future records.

Simplicity. In some data mining applications, concerned more with interpretation than with prediction, it is important that the models generated be easily translated into simple rules that can be understood by experts in the application domain. As a trade-off for achieving simpler rules, decision makers are sometimes willing to allow a slight decrease in accuracy. Data reduction often represents an effective technique for deriving models that are more easily interpretable.

Since it is difficult to develop a data reduction technique that represents the optimal solution for all the criteria described, the analyst will aim for a suitable trade-off among all the requirements outlined.

Data reduction can be pursued in three distinct directions, described below: a reduction in the number of observations through *sampling*, a reduction in the number of attributes through *selection* and *projection*, and a reduction in the number of values through *discretization* and *aggregation*.

6.3.1 Sampling

A further reduction in the size of the original dataset can be achieved by extracting a sample of observations that is significant from a statistical standpoint. This type of reduction is based on classical inferential reasoning. It is therefore necessary to determine the size of the sample that guarantees the level of accuracy required by the subsequent learning algorithms and to define an adequate sampling procedure. Sampling may be *simple* or *stratified* depending on whether one wishes to preserve in the sample the percentages of the original dataset with respect to a categorical attribute that is considered critical.

Generally speaking, a sample comprising a few thousand observations is adequate to train most learning models. It is also useful to set up several independent samples, each of a predetermined size, to which learning algorithms should be applied. In this way, computation times increase linearly with the number of samples determined, and it is possible to compare the different models generated, in order to assess the robustness of each model and the quality of the knowledge extracted from data against the random fluctuations existing in the sample. It is obvious that the conclusions obtained can be regarded as robust when the models and the rules generated remain relatively stable as the sample set used for training varies.

6.3.2 Feature selection

The purpose of *feature selection*, also called *feature reduction*, is to eliminate from the dataset a subset of variables which are not deemed relevant for the purpose of the data mining activities. One of the most critical aspects in a learning process is the choice of the combination of predictive variables more suited to accurately explain the investigated phenomenon.

Feature reduction has several potential advantages. Due to the presence of fewer columns, learning algorithms can be run more quickly on the reduced dataset than on the original one. Moreover, the models generated after the elimination from the dataset of uninfluential attributes are often more accurate and easier to understand.

Feature selection methods can be classified into three main categories: *filter* methods, *wrapper* methods and *embedded* methods.

Filter methods. Filter methods select the relevant attributes before moving on to the subsequent learning phase, and are therefore independent of the specific algorithm being used. The attributes deemed most significant are selected for learning, while the rest are excluded. Several alternative statistical metrics have been proposed to assess the predictive capability and relevance of a group of attributes. Generally, these are monotone metrics in that their value increases or decreases according to the number of attributes considered. The simplest filter method to apply for supervised learning involves the assessment of each single attribute based on its level of correlation with the target. Consequently, this lead to the selection of the attributes that appear mostly correlated with the target.

Wrapper methods. If the purpose of the data mining investigation is classification or regression, and consequently performances are assessed mainly in terms of accuracy, the selection of predictive variables should be based not only on the level of relevance of each single attribute but also on the specific learning algorithm being utilized. Wrapper methods are able to meet this need, since they assess a group of variables using the same classification or regression algorithm used to predict the value of the target variable. Each time, the algorithm uses a different subset of attributes for learning, identified by a search engine that works on the entire set of all possible combinations of variables, and selects the set of attributes that guarantees the best result in terms of accuracy. Wrapper methods are usually burdensome from a computational standpoint, since the assessment of every possible combination identified by the search engine requires one to deal with the entire training phase of the learning algorithm. An example of the use of a wrapper method for attribute selection is given in Section 8.5 in the context of multiple linear regression models.

Embedded methods. For the embedded methods, the attribute selection process lies *inside* the learning algorithm, so that the selection of the optimal set of attributes is directly made during the phase of model generation. Classification trees, described in Chapter 10, are an example of embedded methods. At each tree node, they use an evaluation function that estimates the predictive value of a single attribute or a linear combination of variables. In this way, the relevant attributes are automatically selected and they determine the rule for splitting the records in the corresponding node.

Filter methods are the best choice when dealing with very large datasets, whose observations are described by a large number of attributes. In these cases, the application of wrapper methods is inappropriate due to very long computation times. Moreover, filter methods are flexible and in principle can be associated with any learning algorithm. However, when the size of the problem at hand is moderate, it is preferable to turn to wrapper or embedded methods which afford in most cases accuracy levels that are higher compared to filter methods.

As described above, wrapper methods select the attributes according to a search scheme that inspects in sequence several subsets of attributes and applies the learning algorithm to each subset in order to assess the resulting accuracy of the corresponding model. If a dataset contains n attributes, there are 2^n possible subsets and therefore an exhaustive search procedure would require excessive computation times even for moderate values of n. As a consequence, the procedure for selecting the attributes for wrapper methods is usually of a heuristic nature, based in most cases on a *greedy* logic which evaluates for each attribute a relevance indicator adequately defined and then selects the attributes based on their level of relevance.

In particular, three distinct myopic search schemes can be followed: *forward*, *backward* and *forward–backward* search.

Forward. According to the forward search scheme, also referred to as *bottom-up* search, the exploration starts with an empty set of attributes and subsequently introduces the attributes one at a time based on the ranking induced by the relevance indicator. The algorithm stops when the relevance index of all the attributes still excluded is lower than a prefixed threshold.

Backward. The backward search scheme, also referred to as *top-down* search, begins the exploration by selecting all the attributes and then eliminates them one at a time based on the preferred relevance indicator. The algorithm stops when the relevance index of all the attributes still included in the model is higher than a prefixed threshold.

Forward–backward. The forward–backward method represents a trade-off between the previous schemes, in the sense that at each step the best attribute among those excluded is introduced and the worst attribute among those included is eliminated. Also in this case, threshold values for the included and excluded attributes determine the stopping criterion.

The various wrapper methods differ in the choice of the relevance measure as well as well as the threshold preset values for the stopping rule of the algorithm.

6.3.3 Principal component analysis

Principal component analysis (PCA) is the most widely known technique of attribute reduction by means of projection. Generally speaking, the purpose of this method is to obtain a projective transformation that replaces a subset of the original numerical attributes with a lower number of new attributes obtained as their linear combination, without this change causing a loss of information. Experience shows that a transformation of the attributes may lead in many instances to better accuracy in the learning models subsequently developed.

Before applying the principal component method, it is expedient to standardize the data, so as to obtain for all the attributes the same range of values, usually represented by the interval $[-1, 1]$. Moreover, the mean of each attribute $\mathbf{a}_j$ is made equal to 0 by applying the transformation

$$\tilde{x}_{ij} = x_{ij} - \frac{1}{m}\sum_{i=1}^{m} x_{ij}. \tag{6.6}$$

Let $\mathbf{X}$ denote the matrix resulting from applying the transformation (6.6) to the original data, and let $\mathbf{V} = \mathbf{X}'\mathbf{X}$ be the covariance matrix of the attributes (for a definition of the covariance and variance matrices, see Section 7.3.1). If the correlation matrix is used to develop the principal component analysis method instead of the covariance matrix, the transformation (6.6) is not required.

Starting from the n attributes in the original dataset, represented by the matrix $\mathbf{X}$, the principal component method derives n orthogonal vectors, namely the *principal components*, which constitute a new basis of the space $\mathbb{R}^n$. Principal components are better suited than the original attributes to explain fluctuations in the data, in the sense that usually a subset consisting of q principal components, with $q < n$, has an information content that is almost equivalent to that of the original dataset. As a consequence, the original data are projected into a lower-dimensional space of dimension q having the same explanatory capability.

Principal components are generated in sequence by means of an iterative algorithm. The first component is determined by solving an appropriate optimization problem, in order to explain the highest percentage of variation in the data. At each iteration the next principal component is selected, among those vectors that are orthogonal to all components already determined, as the one which explains the maximum percentage of variance not yet explained by the previously generated components. At the end of the procedure the principal components are ranked in non-increasing order with respect to the amount of variance that they are able to explain.

Let $\mathbf{p}_j$, $j \in \mathcal{N}$, denote the n principal components, each of them being obtained as a linear combination $\mathbf{p}_j = \mathbf{X}\mathbf{w}_j$ of the available attributes, where the weights $\mathbf{w}_j$ have to be determined. The projection of a generic example $\mathbf{x}_i$ in the direction of the weights vector $\mathbf{w}_j$ is given by $\mathbf{w}'_j\mathbf{x}_i$. It can easily be seen that its variance is given by

$$\begin{aligned} \mathrm{E}[\mathbf{w}'_j\mathbf{x}_i - \mathrm{E}[\mathbf{w}'_j\mathbf{x}_i]]^2 &= \mathrm{E}[(\mathbf{w}'_j(\mathbf{x}_i - \mathrm{E}[\mathbf{x}_i]))^2] \\ &= \mathbf{w}'_j\,\mathrm{E}[(\mathbf{x}_i - \mathrm{E}[\mathbf{x}_i])'(\mathbf{x}_i - \mathrm{E}[\mathbf{x}_i])]\mathbf{w}_j \\ &= \mathbf{w}'_j\mathbf{V}\mathbf{w}_j. \end{aligned} \tag{6.7}$$

The first principal component $\mathbf{p}_1$ represents a vector in the direction of maximum variance in the space of the original attributes and therefore its weights may be obtained by solving the quadratic constrained maximization problem

$$\max_{\mathbf{w}_1} \quad \{\mathbf{w}'_1\mathbf{V}\mathbf{w}_1 : \ \mathbf{w}'_1\mathbf{w}_1 = 1\}, \tag{6.8}$$

where the unit norm constraint for $\mathbf{w}_1$ is introduced in order to derive a well-posed problem. By introducing the Lagrangian function

$$L(\mathbf{w}_1, \lambda_1) = \mathbf{w}'_1\mathbf{V}\mathbf{w}_1 - \lambda_1(\mathbf{w}'_1\mathbf{w}_1 - 1), \tag{6.9}$$

and applying the Karush–Kuhn–Tucker conditions, the solution of the maximization problem reduces to the solution of the system

$$\frac{\partial L(\mathbf{w}_1, \lambda_1)}{\partial \mathbf{w}_1} = 2\mathbf{V}\mathbf{w}_1 - 2\lambda_1 \mathbf{w}_1 = \mathbf{0}, \tag{6.10}$$

$$\frac{\partial L(\mathbf{w}_1, \lambda_1)}{\partial \lambda_1} = 1 - \mathbf{w}_1'\mathbf{w}_1 = 0, \tag{6.11}$$

which can be rewritten as

$$(\mathbf{V} - \lambda_1 \mathbf{I})\mathbf{w}_1 = \mathbf{0}, \tag{6.12}$$

subject to the unit norm condition $\mathbf{w}_1'\mathbf{w}_1 = 1$, where $\mathbf{I}$ is the identity matrix. The solution of the maximization problem is therefore given by $\mathbf{w}_1 = \mathbf{u}_1$, where $\mathbf{u}_1$ is the eigenvector of unit norm associated with the maximum eigenvalue λ_1 of the covariance matrix $\mathbf{V}$. Since the variance we wish to maximize is given by

$$\mathbf{w}_1'\mathbf{V}\mathbf{w}_1 = \lambda_1 \mathbf{w}_1'\mathbf{w}_1 = \lambda_1 \mathbf{u}_1'\mathbf{u}_1 = \lambda_1, \tag{6.13}$$

the first principal component is obtained by means of the eigenvector $\mathbf{u}_1$, associated with the maximum eigenvalue λ_1 of $\mathbf{V}$, through the relation $\mathbf{p}_j = \mathbf{X}\mathbf{u}_j$.

The second principal component may be determined by solving an optimization problem similar to (6.8), adding the condition of orthogonality to the previously obtained principal component, expressed by the constraint

$$\mathbf{w}_2'\mathbf{u}_1 = 0. \tag{6.14}$$

Proceeding in an iterative way, it is possible to derive the n principal components starting from the eigenvectors $\mathbf{u}_j$, $j \in \mathcal{N}$, of $\mathbf{V}$ ordered by non-increasing eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$, through the equalities $\mathbf{p}_j = \mathbf{X}\mathbf{u}_j$. The variance of the principal component $\mathbf{p}_j$ is given by $\text{var}(\mathbf{p}_j) = \lambda_j$.

The n principal components constitute a new basis in the space $\mathbb{R}^n$, since the vectors are orthogonal to each other. Therefore, they are also uncorrelated and can be ordered according to a relevance indicator expressed by the corresponding eigenvalue. In particular, the first principal component explains the greatest proportion of variance in the data, the second explains the second greatest proportion of variance, and so on. If $\mathbf{U}$ denotes the $n \times n$ matrix whose columns are the eigenvectors $\mathbf{u}_j$, $j \in \mathcal{N}$, and $\mathbf{P}$ indicates the $n \times n$ matrix whose columns are the principal components $\mathbf{p}_j$, $j \in \mathcal{N}$, then the equality $\mathbf{P} = \mathbf{X}\mathbf{U}$ holds true. The total variance of the principal components is equal to the total variance of the original attributes, that is,

$$\sum_{j=1}^{n} \text{var}(\mathbf{p}_j) = \sum_{j=1}^{n} \lambda_j = \text{tr}(\mathbf{V}) = \sum_{j=1}^{n} \text{var}(\mathbf{a}_j). \tag{6.15}$$

The interpretation of the principal components may be obtained from the coefficients of the vector $\mathbf{w}_j = \mathbf{u}_j$ which express their relationship with the original attributes. To this end, notice that the principal component $\mathbf{p}_h$ assumes the form

$$\mathbf{p}_h = u_{h1}\mathbf{a}_1 + u_{h2}\mathbf{a}_2 + \cdots + u_{hn}\mathbf{a}_n. \tag{6.16}$$

The coefficient u_{hj} can be therefore interpreted as the weight of the attribute $\mathbf{a}_j$ in determining the component $\mathbf{p}_h$. The greater the absolute value of u_{hj} is, the more the component $\mathbf{p}_h$ is characterized by the attribute $\mathbf{a}_j$. At the same time, $\text{var}(\mathbf{p}_h) = \lambda_h$ represents a measure of the proportion of total variance explained by the principal component $\mathbf{p}_h$. For this reason, the index

$$I_q = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_q}{\lambda_1 + \lambda_2 + \cdots + \lambda_n} \tag{6.17}$$

expresses the percentage of total variance explained by the first q principal components and provides an indication of the amount of information preserved by the first q components. In order to determine the number of principal components to be appropriately used, it is possible to go on until the level of overall importance I_q of the considered components exceeds a threshold $I_{\min}$ deemed reasonable, in relation to the properties of the dataset. The number of principal components is therefore determined as the smallest value q such that $I_q > I_{\min}$.

An example of application of principal component analysis

To illustrate the application of the principal component method, consider the *mtcars* dataset, described in Appendix B, which contains 11 attributes representing the main characteristics of different types of cars.

Table 6.1 indicates the overall importance of the principal components. It displays for each component $\mathbf{p}_h$ the standard deviation, given by $\sqrt{\lambda_h}$, the proportion of variance explained by each component, equal to $\lambda_h / \sum_j \lambda_j$, and the cumulative proportion explained by the first components up to $\mathbf{p}_h$ included. The analysis shows that the first two components explain 84% of the total variation in the data while the first 5 explain 94%.

Table 6.2 shows the values of the coefficients $w_{hj} = u_{hj}$; equivalently, it indicates the eigenvectors of the covariance matrix $\mathbf{V}$. The first component, which alone explains 60% of the variance, is negatively correlated with the attributes $\{cyl, disp, wt, carb\}$, whose meaning is explained in Appendix B, while it is positively correlated with all the other attributes.

Finally, Figure 6.2 shows through a *scree plot* the decreasing progression of the variance explained by the principal components. The chart may prove helpful in identifying the relevant components to be included in the transformed dataset.

Table 6.1 Overall importance of principal components in the *mtcars* dataset

	Comp.1	**Comp.2**	**Comp.3**	**Comp.4**
Standard deviation	2.5706809	1.6280258	0.7919578	0.5192277
Proportion of variance	0.6007637	0.2409516	0.0570179	0.0245088
Cumulative proportion	0.6007637	0.8417153	0.8987332	0.9232420
	Comp.5	**Comp.6**	**Comp.7**	**Comp.8**
Standard deviation	0.4727061	0.4599957	0.3677798	0.3505730
Proportion of variance	0.0203137	0.0192360	0.0122965	0.0111728
Cumulative proportion	0.9435558	0.9627918	0.9750883	0.9862612
	Comp.9	**Comp.10**	**Comp.11**	
Standard deviation	0.2775727	0.2281127	0.1484735	
Proportion of variance	0.0070042	0.0047304	0.0020040	
Cumulative proportion	0.9932654	0.9979959	1.0000000	

Table 6.2 Principal component coefficients for the *mtcars* dataset

	Comp.1	**Comp.2**	**Comp.3**	**Comp.4**	**Comp.5**	**Comp.6**
mpg	0.363	−0.226	−0.103	−0.109	0.368	0.754
cyl	−0.374	−0.175	0.169	0.231	−0.846	
disp	−0.368	0.257	−0.394	−0.336	0.214	0.198
hp	−0.33	−0.249	0.14	−0.54	0.222	−0.576
drat	0.294	−0.275	0.161	0.855	0.244	−0.101
wt	−0.346	0.143	0.342	0.246	−0.465	0.359
qsec	0.2	0.463	0.403	0.165	−0.33	0.232
vs	0.307	0.232	0.429	−0.215	−0.6	0.194
am	0.235	−0.429	−0.206	−0.571	−0.587	−0.178
gear	0.207	−0.462	0.29	−0.265	−0.244	0.605
carb	−0.214	−0.414	0.529	−0.127	0.361	0.184
	Comp.7	**Comp.8**	**Comp.9**	**Comp.10**	**Comp.11**	
mpg	0.236	0.139			0.125	
cyl					0.141	
disp					−0.661	
hp	0.248				0.256	
drat						
wt					0.567	
qsec	−0.528	−0.271			−0.181	
vs	−0.266	0.359	−0.159			
am						
gear	−0.336	−0.214				
carb	−0.175	0.396	0.171		−0.32	

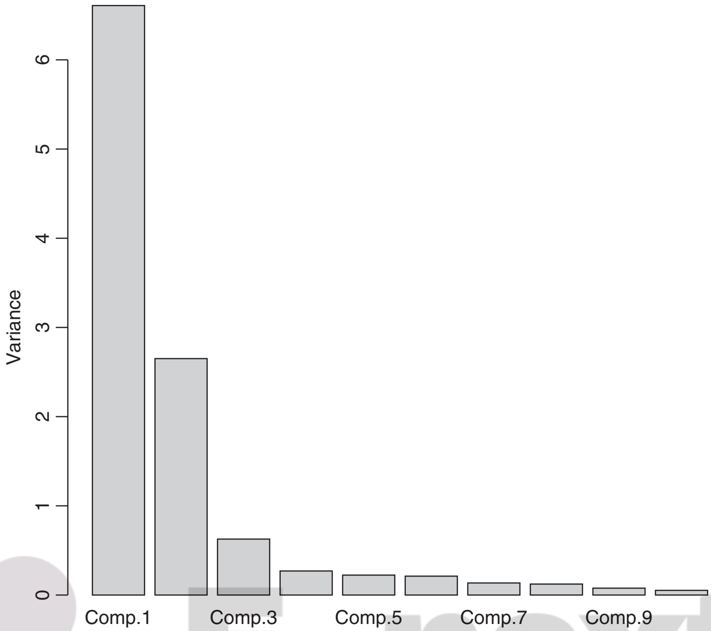

Figure 6.2 A scree plot for the principal components in the mtcars dataset

6.3.4 Data discretization

The general purpose of data reduction methods is to obtain a decrease in the number of distinct values assumed by one or more attributes. Data discretization is the primary reduction method. On the one hand, it reduces continuous attributes to categorical attributes characterized by a limited number of distinct values. On the other hand, its aim is to significantly reduce the number of distinct values assumed by the categorical attributes.

For instance, the weekly spending of a mobile phone customer is a continuous numerical value, which might be discretized into, say, five classes: low, [0 – 10) euros; medium low, [10 – 20) euros; medium, [20 – 30) euros; medium high, [30 – 40) euros; and high, over 40 euros.

As a further example applied to a categorical variable, consider the province of residence of each customer, and suppose it can assume a hundred distinct values. If instead of the province one uses the region of residence, the new attribute might take on twenty distinct values.

In both cases, the discretization process has brought about a reduction in the number of distinct values assumed by each attribute. The models that can be generated on the reduced dataset are likely to be more intuitive and less arbitrary. For instance, using a classification tree, it is possible to generate a rule of the form

> if spending is in the medium low range, and if a customer resides in region A, then the probability of churning is higher than 0.85.

This is much more interpretable than the rule

> if spending is in the [12.21, 14.79] euro range, and if a customer resides in province B, then the probability of churning is higher than 0.85,

which could have been generated for the original dataset.

The examples shown above suggest that discretization and reduction of the number of values taken by each attribute can improve the generalization capability of predictive models, thus making easier the interpretation of the rules obtained.

Among the most popular discretization techniques are *subjective subdivision*, *subdivision into classes* and *hierarchical discretization*.

Subjective subdivision. Subjective subdivision is the most popular and intuitive method. Classes are defined based on the experience and judgment of experts in the application domain.

Subdivision into classes. Subdivision into categorical classes may be achieved in an automated way using the techniques described below. In particular, the subdivision can be based on classes of equal size or equal width.

Hierarchical discretization. The third type of discretization is based on hierarchical relationships between concepts and may be applied to categorical attributes, just as for the hierarchical relationships between provinces and regions. In general, given a hierarchical relationship of the one-to-many kind, it is possible to replace each value of an attribute with the corresponding value found at a higher level in the hierarchy of concepts.

Subdivision into classes

The automated procedure of subdivision into classes consists of ordering in a non-decreasing way the values of the attribute $\mathbf{a}_j$ and grouping them into a predetermined number K of contiguous classes. It is possible to form the classes of either different *size* or different *width*.

In the first case, the m observed values available for the attribute $\mathbf{a}_j$ are distributed by placing $\lfloor m/K \rfloor$ or $\lceil m/K \rceil$ contiguous values in each class, so as to divide the m observed values almost equally among the K classes.

In the second case, the range of total variation between the minimum value and the maximum value taken by the attribute $\mathbf{a}_j$ is subdivided into K contiguous intervals and the observed values are placed in the class corresponding to the interval where they fall. This second procedure is less effective if the distribution of the values significantly moves away from the uniform distribution. Once the K classes have been constructed, each observed value of $\mathbf{a}_j$ is replaced by the average value of the corresponding class. As an alternative, instead of using the average value for regularization, it is possible to use the boundary value of the class that is closest to the original value taken by $\mathbf{a}_j$.

The following examples show the different regularization methods based on the subdivision into classes.

Example 6.1 – Subdivision into equal size classes. Ordered values of $\mathbf{a}_j$: 3, 4, 4, 7, 12, 15, 21, 23, 27.
Class 1: 3, 4, 4
Class 2: 7, 12, 15
Class 3: 21, 23, 27

Example 6.2 – Regularization by the mean value. Ordered values of $\mathbf{a}_j$: 3, 4, 4, 7, 12, 15, 21, 23, 27.
Class 1: 3.66, 3.66, 3.66
Class 2: 11.33, 11.33, 11.33
Class 3: 23.66, 23.66, 23.66

Example 6.3 – Regularization by boundary values. Ordered values of $\mathbf{a}_j$: 3, 4, 4, 7, 12, 15, 21, 23, 27.
Class 1: 3, 4, 4
Class 2: 7, 15, 15
Class 3: 21, 21, 27

Example 6.4 – Subdivision into equal width classes. Ordered values of $\mathbf{a}_j$: 3, 4, 4, 7, 12, 15, 21, 23, 27.
Class 1 - interval [3, 11): 3, 4, 4, 7
Class 2 - interval [11, 19): 12, 15
Class 3 - interval [19, 27]: 21, 23, 27