An automated report generation tool for the data understanding phase

Juha Vesanto and Jaakko Hollmén

Helsinki University of Technology
Laboratory of Computer and Information Science
P.O. Box 5400, FIN-02015 HUT, Finland
Juha.Vesanto@hut.fi, Jaakko.Hollmen@hut.fi

Abstract To successfully prepare and model data, the data miner needs to be aware of the properties of the data manifold. In this chapter, the outline of a tool for automatically generating data survey reports for this purpose is described. Such a report is used as a starting point for data understanding, acts as documentation of the data, and can easily be redone if necessary. The main focus is on describing the cluster structure and the contents of the clusters. The described system combines linguistic descriptions (rules) and statistical measures with visualizations. Whereas rules and mathematical measures give quantitative information, the visualizations give qualitative information on the data sets, and help the user to form a mental model of the data based on the suggested rules and other characterizations.

1 Introduction

The purpose of data mining is to find knowledge from databases where the dimensionality, complexity, or amount of data is prohibitively large for manual analysis. This is an interactive process which requires that the intuition and background knowledge of application experts are coupled with the computational efficiency of modern computer technology.

The CRoss-Industry Standard Process for Data Mining (CRISP-DM) [4] divides the data mining process to several phases. One of the first phases is data understanding, which is concerned with understanding the origin, nature and reliability of the data, as well as becoming familiar with the contents of the data through data exploration. Understanding the data is essential in the whole knowledge discovery process. Proper data preparation, selection of modeling tools and evaluation processes is only possible if the miner has a good overall idea, or a mental model, of the data.

The data exploration is usually done by interactively applying a set of data exploration tools and algorithms to get an overview of the properties of the data manifold. However, understanding a single data set is often not enough. Because of the iterative nature of the knowledge discovery process, several different data sets and preprocessing strategies need to be considered and explored. The task of data understanding is engaged again and again. Therefore, the tools used for data understanding should be as automated as possible.

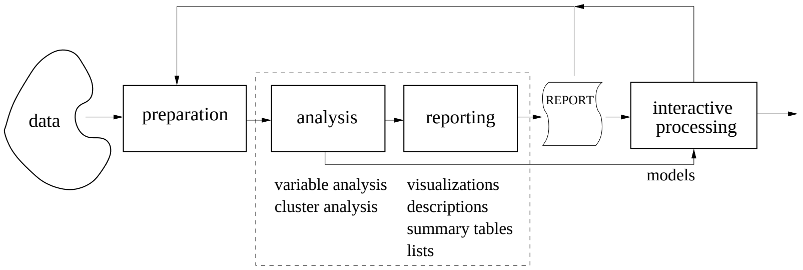


Figure1. Creating understanding of the data in a survey cycle. The data is prepared and fed into the analysis system which generates the data survey report. Based on the findings in the report and possibly further insights based on interactive investigation, the data miner may either proceed with the next data mining phase, or prepare the data set in a better, alternative fashion and, with a push of a button, make a new report. The area within the dashed box corresponds to the implemented report generation system.

1.1 Automated analysis of table-format data

This chapter presents a selection of techniques and associated presentation templates to automate part of the data understanding process. The driving goal of the work has been to construct a framework where an overview and initial analysis of the data can be executed automatically, without user intervention. The motivation for the work has come from a number of data mining projects, in process industry for example [1], where we have repeatedly encountered the data understanding task when new data sets and/or preprocessing strategies have been considered.

While statistical and numerical program packages provide a multitude of techniques that are similar to those presented here, the novelty of our approach is to combine the techniques and associated visualizations into a coherent whole. In addition, using such program packages requires considerable time and expertise. An automated approach used in this chapter has a number of desirable properties:

- the analysis is easy to execute again (and again and ...),
- the required level of technical know-how of the data miner is reduced when compared to fully interactive data exploration, and
- the resulting report acts as documentation that can be referred to later.

Of course, an automatically performed analysis can never replace the flexibility and power inherent in an interactive approach. Instead, we consider the report generation system described here to provide an advantageous starting point for such interactive analysis (see Figure 1).

The nature of the report generation system imposes some requirements for the applied methods. The algorithms should be computationally light,

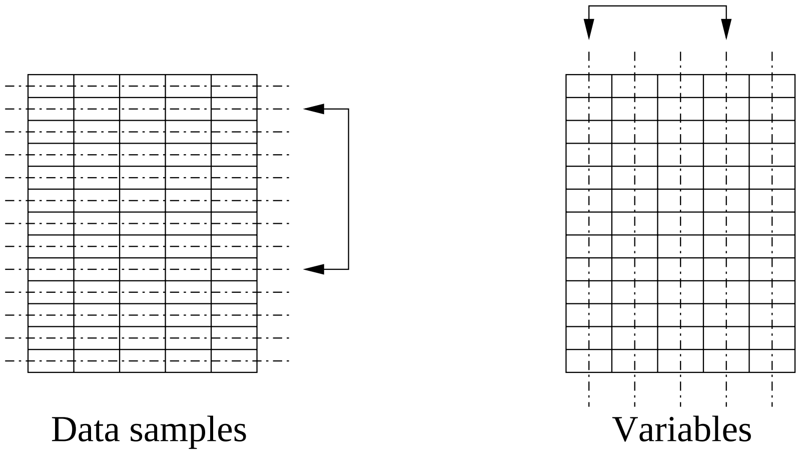


Figure2. Table-format data can be investigated both in terms of samples and in terms of variables. Samples are analyzed to find groups of similar items. Variables are analyzed to find groups of related items. Sample analysis is covered in Section 2 and variable analysis in Section 3.

robust, and require no user-defined parameters. They should also be generic enough so that their results are of interest in most cases. Naturally, what is interesting depends on the problem and the data domain. In the implemented system, the domain has been restricted to unsupervised analysis of unordered, numerical vector (table-format) data, see Figure 2. In contrast, supervised estimation of one or more output variables, analysis of purely categorical variables, and analysis of data sequences or time-series are not considered. In a more complete system also these, especially the issues in supervised estimation, should be addressed.

However, simply applying the analysis algorithms to the data is not enough: the knowledge must be transferred to the data miner. This is accomplished through a data survey report. The report consists of visualizations and numerical or linguistic descriptions organized according to a predefined template into summary tables and lists. Visualizations are a very important part of the report since they allow the display of large amounts of detailed information in a coherent manner, and give the data miner a chance to validate the quantitative descriptions with respect to the actual data.

1.2 Related work

In [15], Pyle introduces the concept of data survey as a tool for getting a feel of the data manifold. The emphasis is on variable dependency analysis using entropy and related measures, and on identifying problems in the data.

Unfortunately, only rather general guidelines are given, and cluster analysis is handled very briefly.

Cluster analysis, and especially characterization of the contents of the clusters is one of the key issues in this chapter. The clusters can be characterized by listing the variable values typical for each cluster using, for example, characterizing rules [18]. Another approach is to rank the variables in the order of significance [12,16,17]. In this chapter, both approaches are used.

In [14], a report generation system KEFIR is described. It automatically analyzes data in large relational databases, and produces a report on the key findings. The difference to our work is that KEFIR compares a data set and *a priori* given normative values, and tries to find and explain deviations between them, and thus requires considerable setting up. The system described in this chapter, on the contrary, is applied when the data miner starts with a single unfamiliar data set.

In this sense, the recent work by Bay and Pazzani [2] is much closer to certain parts of this work. They examine the problem of mining contrast sets, where the fundamental problem is to find out how two groups differ from each other, and propose an efficient search algorithm to find conjunctive sets of variables and values which are meaningfully different in the two groups. The difference to our work is that they are concerned with categorical variables, and want to find all relevant differences between two arbitrary groups. In this work, the data is numerical, and the two groups are always two clusters, or a cluster and the rest of the data.

In the implemented system, metric clustering techniques are used, so the input data needs to be numerical vector-data (which may have missing values). Another possibility would be to use conceptual clustering techniques [13], which inherently focus on descriptions of the clusters. However, conceptual clustering techniques are rather slow, while recent advances have made metric clustering techniques applicable to very large data sets [27,7].

1.3 Contents

The chapter is organized as follows. In this Section, the methodology and basic motivation for the implemented system has been described. In Sections 2 and 3, the analysis methods for samples and variables, respectively, are described. In Section 4, the overall structure of the data survey report is explained. The system data set collected by the authors is used throughout the chapter to illustrate the different aspects of the report and is more closely described in Appendix A. In Section 5, a publicly available insurance company data set [21] is analyzed in order to describe the use of the data survey report. The work is summarized in Section 6.

2 Sample analysis

The relevant questions in terms of samples are: Are there natural groups, i.e. clusters, in the data? What kind of segments can be formed, and what are their properties?

2.1 Projection

A qualitative idea of the cluster structure in the data is acquired by visualizing the data using vector projection methods. Projection algorithms try to preserve distances or neighborhoods of the samples, and thus encode similarity information in the projection coordinates. There are many different kinds of projection algorithms, see for example [11], but in the proposed system, a classical linear projection based on principal component analysis (PCA) is used. In PCA, the directions are found which account for most of the variance in the data. This is done by calculating the eigenvectors $\mathbf{e}_1, ..., \mathbf{e}_d$ and corresponding eigenvalues $\lambda_1, ..., \lambda_d$ of the covariance matrix of the data, and ordering them by decreasing eigenvalues $\lambda_1 > \lambda_2 > ... > \lambda_d$. The first direction $\mathbf{e}_1$ accounts for most — specifically $\frac{\lambda_1}{\sum_i \lambda_i} 100\%$ — of the variance in the data, the second for the second largest amount, and so on. By projecting data to the space spanned by the first few eigenvectors as much of the variance is preserved as possible. The sum of the corresponding eigenvalues gives the amount of variance preserved in the projection, and thus indicates the error made in the low-dimensional projection. For example, a projection to a 2-dimensional plane is defined as:

$$\mathbf{y} = \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \end{bmatrix} \mathbf{x}. \tag{1}$$

The advantages of PCA projection include computational efficiency, and the ability to project new data points (e.g. cluster centers) easily. It also allows the use of a scree plot for easily understandable validation of the projection, see Figure 3a.

Like spatial coordinates, colors can also be used to encode similarity information [26,23,9]. In the implemented system, the colors are assigned from the hues on a color circle by a simple projection of cluster centroids onto the circle. A color coding can be constructed by defining a smooth coloring in a low-dimensional manifold, and projecting the data onto this manifold, for example as follows:

1. A 1-dimensional SOM (see Section 2.2 below) is trained using the data. The trained SOM forms a principal curve going through the data manifold.
2. A color from the color hue circle (from the HSV color model, see for example [26], with hue = ϕ, saturation = 1 and value = 1) is assigned to each map unit i of the 1-dimensional SOM. The colors can be assigned

equidistant from each other $\phi_i = 2\pi i/M$ or the distances between neighboring prototypes can be taken into account: $\phi_i = 2\pi \sum_{j=1}^{i} \|\mathbf{m}_{i+1} - \mathbf{m}_i\| / \sum_{j=1}^{M-1} \|\mathbf{m}_{i+1} - \mathbf{m}_i\|$.

3. Each data sample picks the same color as its BMU.

These colors are used consistently in various visualizations throughout the report.

2.2 Clustering

Clustering algorithms, see for example [6], provide a more quantitative analysis of the natural groups that exist in the data. In real data, however, clusters are rarely compact, well-separated groups of objects — the conceptual idea that is often used to motivate clustering algorithms. Apart from noise and outliers, clustering may depend on the level of detail being observed. Therefore, instead of providing a single partitioning of the data, the implemented system constructs a cluster hierarchy, see Figure 3b. This may represent the inherent structure of the data set better than a direct partitioning. Equally important from data understanding point of view is that it also allows the data to be investigated at several levels of granularity.

Base clusters In the implemented system, the Self-Organizing Map (SOM) is used for clustering [10]. The SOM is a collection of prototype vectors $\mathbf{m}$, between which a neighborhood relation h is defined. The neighborhood relation defines a structured lattice, usually a two-dimensional, rectangular or hexagonal lattice of map units. After initializing the prototype vectors with, for example, random values, training takes place. Training a Self-Organizing Map from data is divided to two steps, which are applied alternately. First, a best-matching unit (BMU) or a winner unit b_i is searched, which minimizes the Euclidean distance between a data sample $\mathbf{x}_i$ and the map unit prototypes $\mathbf{m}_j$

$$b_i = \arg\min_j \|\mathbf{x}_i - \mathbf{m}_j\|. \tag{2}$$

Then, new prototypes are calculated as:

$$\mathbf{m}_j = \frac{\sum_{i=1}^{n} h_{b_i j} \mathbf{x}_i}{\sum_{i=1}^{n} h_{b_i j}}, \tag{3}$$

where $h_{b_i j}$ is the neighborhood strength between map units b_i and j, and n is the number of data samples. This is the batch training algorithm for the SOM.

After quantizing the data using a SOM with a few hundred map units, the map units are clustered. Thus, in the second phase only a few hundred objects need to be clustered instead of all the original data samples. This

2-phase strategy reduces the computational complexity of the clustering considerably [24]. In addition, the SOM is useful as a convenient projection of the data cloud, see Section 3.

To cluster the units of the SOM, a widely used technique is the U-matrix [20]. It is, in effect, a measure of the local probability density of the data in each map unit. Thus, the local minima of the U-matrix — map units for which the distance matrix value is lower than that of any of their neighbors — can be used to identify cluster centers. In [22], the rest of the map units were assigned to the cluster whose center was closest. This procedure is simple and fast, but it also makes the implicit assumption that the border between two clusters lies on the middle point between their cluster centers. We use an enhanced version based on region-growing. This procedure provides a partitioning of the map into a set of base clusters, the number of which is equal to the number of local minima on the distance matrix [25].

Cluster hierarchy Starting from the base clusters, some agglomerative clustering algorithm is used to construct the initial cluster hierarchy. Agglomerative clustering algorithms start from some initial set of c clusters and successively join the two clusters closest to each other (in terms of some distance measure), until there is only one cluster left. This produces a binary tree with $2c - 1$ clusters.

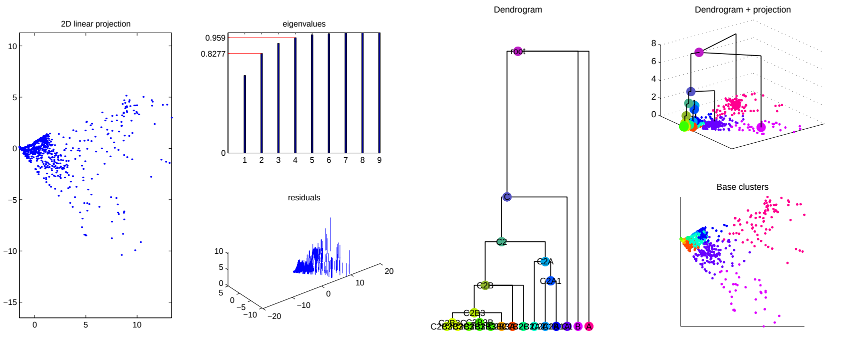
Since the binary structure does not necessarily reflect the properties of the data set, a number of the clusters in the initial hierarchy will be superfluous and need to be pruned out. This can be done by hand using some kind of interactive tool [3], or in an automated fashion using some cluster validity index to prune out the improper clusters. In the implemented system, the following procedure is applied:

1. Start from root (top level) cluster.
2. For the cluster c under investigation, generate different sub-cluster sets. A sub-cluster set may contain either sub-clusters of cluster c or sub-clusters of c's sub-clusters (sub-sub-clusters).
3. Each sub-cluster set defines a partitioning of the data in the investigated cluster. Investigate each partitioning using some clustering validity measure, for example Davies-Bouldin index [5] or other similar index [25].
4. Select the best sub-cluster set (for example the one with minimum I_{gap}), and prune the corresponding intermediate clusters.
5. Select an uninvestigated cluster (if any), and continue from step 2.

In Figure 3b, the clusters and cluster hierarchy are presented in three visualizations linked with each other and with the projection results.

2.3 Cluster characterization

Descriptive statistics — for example means, standard deviations and histograms of individual variables — can be used to list the typical values for



(a) (b)

Figure3. Projection visualization (on left) and cluster hierarchy (on the right). The projection gives an idea of the shape of the data manifold. The figure is accompanied by the scree plot of eigenvalues, and a plot of projection residuals. These give an idea of the inherent dimensionality of the data set, and the reliability of the 2-dimensional projection. In this case, the projection covers 83% of the total variance. The 95% coverage limit would be reached in 4-dimensional projection (of the total of 9 dimensions). The cluster hierarchy is visualized using a dendrogram with colors and names of the clusters also indicated. The two smaller figures in the right panel (b) link the dendrogram to the projection visualization by showing the dendrogram starting from the 2-dimensional projection coordinates. Each point is colored with the color of the (base) cluster it belongs to.

each cluster. Not all variables are equally interesting or important, though. Interestingness can be defined as deviation from the expected [14,8]. It can be measured for each cluster as the difference between variable distributions in the cluster versus the whole data either using probability densities or some more robust measures, for example standard deviation [17]. Each cluster can then be characterized by using a list of the most important variables and their descriptive statistics.

Another frequently employed method is to form characterizing rules [19,18] to describe the values in each cluster:

$$R_i : \ \mathbf{x} \in C_i \Leftrightarrow x_k \in [\alpha_k, \beta_k] \tag{4}$$

where x_k is the value of variable k in sample vector $\mathbf{x}$, C_i is the investigated cluster, and $[\alpha_k, \beta_k]$ is the range of values allowed for the variable according to the rule. These rules may be expressed in terms of single variables like R_i above, or be conjunctions of several variable-wise rules in which case the rule forms a hypercube in the input space.

The main advantage of such rules is that they are compact, simple and therefore easy to understand. The problem is of course that clusters often do not coincide well with the rules since the edges between clusters are not necessarily in parallel with the edges of the hypercube. In addition, the cluster may include some uncharacteristic points or outliers. Therefore the rules should be accompanied by validity information. The rules R_i can be divided to two different cases, characterizing rules (R_i^c) and differentiating rules (R_i^d):

$$\begin{aligned} R_i^c: &\quad \mathbf{x} \in C_i \Rightarrow x_k \in [\alpha_k, \beta_k] \\ R_i^d: &\quad x_k \in [\alpha_k, \beta_k] \Rightarrow \mathbf{x} \in C_i. \end{aligned}$$

The validity with respect to each case can be measured using confidence: $P_i^c = P(x_k \in [\alpha_k, \beta_k] \,|\, C_i)$ and $P_i^d = P(C_i \,|\, x_k \in [\alpha_k, \beta_k])$, respectively.

To form the characterizing rules — in effect to select the low and high limits of the range — one can use statistics of the values in the clusters [19]. Another approach is to optimize the rules with respect to their significance. The optimization can be interpreted as a two-class classification problem between the cluster and the rest of the data. The boundaries in the characterizing rule can be set by maximizing a function which gets its highest value when there are no misclassification's, for example:

$$s_1 = \frac{a+d}{a+b+c+d}, \tag{5}$$

$$s_2 = \frac{a}{a+b}\frac{a}{a+c}, \tag{6}$$

$$s_3 = \frac{a}{a+b+c}, \tag{7}$$

where a, b, c and d are from the truth table in Figure 4.

The first function s_1 is simply the classification accuracy. It has the disadvantage that if the number of samples in the cluster is much lower than in the whole data set (which is very often the case), s_1 is dominated by the need to classify most of the samples as false. Thus, the allowed range of values in the rule may vanish entirely. However, when characterizing the (positive) relationship between rule R and cluster C, the samples belonging to d are not really interesting. As pointed out in [2], traditional rule-based classification techniques are not well suited for the characterization task.

The two latter measures consider only cases a, b and c. The second measure s_2 is the product of the confidences $s_2 = P_i^c P_i^d$. The third measure is its approximation $s_3 \approx s_2$ when $a \gg b+c$. Compared to s_2, s_3 has the advantage of a clearer interpretation. It is the ratio of correctly classified samples when the case d is ignored, whereas s_2 is the product of two such ratios.

Apart from characterizing the internal properties of the clusters, it is important to understand how they differ from the other, especially neighboring clusters. For the neighboring clusters, the constructed rules may be quite similar, but it is still important to know what makes them different. To do this, rules can be generated using the same procedure as above, but taking only the two clusters into account. In this case, however, both clusters are interesting, and therefore s_1 should be used.

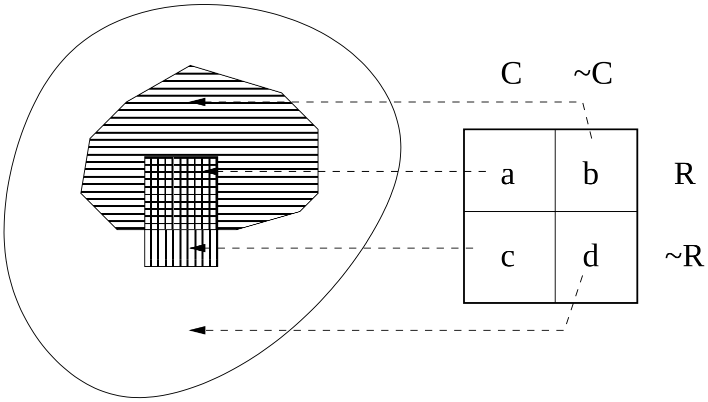
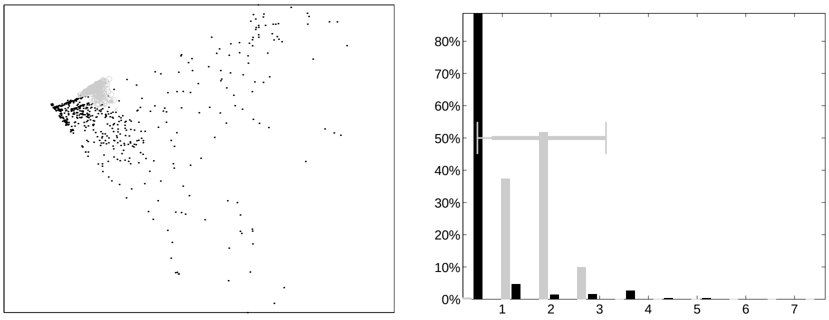


Figure4. The four-way truth table of cluster C and rule R. The cluster is the horizontally shaded area, and the rule (or classification model) is the vertically shaded area. On the right is the corresponding confusion matrix: a is the number of samples which are in the cluster, and for which the rule is true. In contrast, d is number of samples which are out of the cluster, and for which the rule is false. Ideally, the off-diagonal elements in the matrix should be zero.

The report elements to describe the clusters are shown in Figure 5 and Table 1. The former shows the most significant rule visualized with projection and histograms, and the latter a summary table of variable values and associated descriptive rules of each variable, ordered by the significance s_2 of the variable.

Table1. Rule summary for the cluster in Figure 5 in the system data. Variables are listed in order of decreasing significance as measured by s_2. The columns in the middle indicate the properties for the variable-wise rules, and the columns on the right for a conjunctive rule formed of the indicated variables starting from the top. The "diff" columns are confidences in the differentiating property of the rule P^d and "char" column in the characterizing property P^c.

Variable	**Rule**	**diff**	**char**	s_2	**diff**	**char**	s_2
			single			**cumulative**	
`intr`	[0.78,3.1]	75%	99%	0.745	75%	99%	0.745
`idle`	[3.3,4.3]	65%	98%	0.639	87%	98%	0.856
`usr`	[1.2,2.3]	61%	98%	0.605	90%	98%	0.885
`sys`	[0.71,1.4]	69%	62%	0.425			
`blks/s`	< 0.82	22%	98%	0.217	100%	96%	0.96
`wio`	< 1.1	21%	100%	0.214			
`ipkts`	< 0.97	20%	100%	0.201			
`opkts`	< 0.97	20%	100%	0.2			
`wblks/s`	< 1.4	20%	100%	0.198			



(a) A cluster of system data set

(b) Rule for `intr`

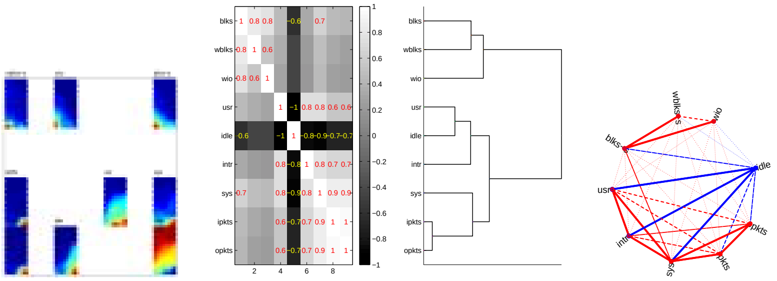
Figure5. A cluster of the system data set. In (a) the PCA projection of the data is shown, with the cluster indicated by the gray markers and the rest of the data with black markers. In (b), the histogram corresponding to the most significant variable is shown (see Table 1). The gray vertical bars are the histogram for the cluster, and the black the histogram for the rest of the data. The horizontal bar indicates the range allowed by the rule (thick part) and the real range of values in the cluster (thin part).

3 Variable analysis

The relevant questions with respect to variables are: What are the distribution characteristics of the variables? Are there pairs or groups of dependent variables? If so, how do they depend on each other?

The distributions of individual variables can be characterized by simple descriptive statistics, for example histograms. The histogram bins are formed either based on the unique values of the variable, if there are at most 10 unique values, or by dividing the range between minimum and maximum values of the variable to 10 equally sized bins.

Dependencies between variables are best detected from ordinary scatter plots, for example from a scatterplot matrix which consists of several subgraphs where each variable is plotted against each other variable. Of course, such visualization technique has the deficiency that the number of pairwise scatter plots increases quadratically with the number of variables. A more efficient, if less accurate, technique is to use component planes. A component plane is a colored scatter plot of the data, where the positions of the markers are defined using a projection such that similar data samples are close to each other. The color of the markers are defined by the values of a variable in the data samples. By using one component plane for each variable, the whole data set can be visualized, see Figure 6a. Relationships between variables can be seen as similar patterns in identical places in the component



(a)

(b)

(c)

Figure6. (a) Component planes. Relationships between variables can be seen as similar patterns on the component planes. (b) Correlation coefficient matrix, and the dendrogram resulting from clustering of the variables using an agglomerative clustering algorithm. (c) Association graph. On the association graph, high positive (negative) correlations are shown with red (blue) lines.

planes. The projection made by SOM works very well with this technique, since the projection focuses locally such that the behavior of the data can be seen irrespective of the local scale.

A more quantitative measure of the dependency between pairs of variables $\{i, j\}$ is the correlations coefficient c_{ij}:

$$c_{ij} = \frac{1}{\sigma_i \sigma_j} \sum_{k=1}^{n} (x_{ki} - \mu_i)(x_{kj} - \mu_j) \tag{8}$$

where μ_i and σ_i are the mean and standard deviations of variable i. It is robust, computationally light and can be efficiently visualized as a matrix, see Figure 6b. Selected correlations are also visualized on an association graph, see Figure 6c. In the implemented system, the correlation coefficients are also used as feature vectors for each variable: $\mathbf{v}_i = [\,|c_{i1}|, |c_{i2}|, ..., |c_{id}|\,]$. Using them and some clustering or projection method (see Section 2), the variables can be clustered (or ordered) to indicate groups of dependent variables [23]. In all three visualizations in Figure 6, the variables have been ordered such that related variables are near each other.

4 Data survey report

The system was implemented as a Matlab script. It was installed on a web server as a cgi-bin script, and thus it could be run remotely through any web browser. The user provided the data and optionally a few other parameters.

The resulting report was provided both as a hypertext document in HTML and in printable form in PostScript/PDF.

The report starts with an overview part, where the top-level results of both variable and cluster analysis are given. The overview provides a quick look at the data. It is short, only 2-4 pages in length, and consists mainly of visualizations so that it can be understood at a glance.

Most of the elements of the overview have already been shown earlier in the chapter (Figures 3a and b, 6a and c). From Figure 3a, one can see that a two-dimensional projection preserves the structure of the system data set very well. Most of the data is tightly packed in a single region, and there is a fan of dispersed data. From Figure 3b, one can see that there are three main operational states in the system, one of which divides further to several sub-states. From Figure 6 one can see that there are two main groups of variables, those involved with disk operations, and the rest.

In addition, the overview part has a table of descriptive statistics for each variable, and a list of most significant rules for each cluster (not shown). From the latter one could see that the main operational states correspond to a normal operation state and two different high load states where either the amount of disk operations is high (`wio` > 2.1) or there is a lot of network traffic (`ipkts` > 1.9). Further investigation reveals that the normal operation state divides to several different types, for example totally idle state, state for mainly system operations, and state for mainly user operations.

The overview is followed by more detailed information of all variables, and clusters (individual cluster characterizations). These allow the reader to get immediately some further information on interesting details. For example, the cluster depicted in Table 1 and Figure 5 is mainly characterized by relatively high level of interruptions, and a moderate level of idle time.

The computational complexity of quantization using SOM is $\mathcal{O}(nmd)$, where n is the number of data points, m is the number of map units, and d is the vector dimension. The complexity of clustering the SOM is $\mathcal{O}((m-c)(c+1)d)$, where c is the number of base clusters. The computational complexity of the hierarchical clustering phase is $\mathcal{O}(c^2d)$. The search for the best rule is an optimization problem which can be solved, for example, using (fast) exhaustive search techniques like the ones introduced in [2]. In the implemented system, though, a much less complex greedy search is used which is linear in computational complexity with respect to the number of variables. Thus, its computational complexity is $\mathcal{O}(cd)$. The computational complexity of variable analysis is $\mathcal{O}(d^3)$ because of clustering the variables. Thus, the overall complexity is of the order of $\mathcal{O}((nm + c^2 + d^2)d)$.

The system runs in main memory, which of course limits the possible size of the data set. For a data set of size 4000 samples and 40 variables, the report generation takes 15 minutes on a Linux workstation with Pentium II 350 MHz processor and 256 MB of memory. The majority of the time is spent to writing the images to files, which is unfortunately slow in Matlab.

The actual analysis only takes about 2 minutes. The heaviest part of the analysis is clustering, which scales linearly with the number of data samples. Thus, the system scales well to larger data sets, too.

5 Case study: caravan insurance policy data set

The second data set used in this chapter is a publicly available insurance company data set used in a recent CoIL challenge [21]. In this data mining competition, two separate tasks were given: predicting and explaining caravan insurance policy ownership. The samples represent customers and the variables different aspects of the customers or their behavior. We do not attempt to give an answer to either of the questions posed in the competition directly, but rather to demonstrate how our data survey report can be used in the initial phases of the data analysis project. In the following, we describe the generation of the report and give a rough idea how to make valuable observations about the data set with the help of the data survey report.

Creating a data survey report begins by inputting the data set to the data survey generator. As indicated in Section 4, the system has been implemented using a interface on a Web server, so the table-format data can be uploaded to the generator using a Web based interface. The data survey report itself is a hypertext document, which allows for navigating between overviews and detailed descriptions as well as between clusters and their sub-clusters with minimal effort. The report starts with an index document that lists all the names of the variables along with their summary statistics. Summary statistics used are the minimum, maximum, mean, standard deviation, and entropy of the recorded variables. In addition, number of missing values, number of unique values and whether all values have integer values are shown. This helps finding out possibly categorical data or discretized variables. This is the case in the insurance data set, which is easily seen from the report. The data set size of 9822 samples and 86 variables is visible on the report header.

The main document is followed by an overview. The overview contains a hierarchical table resembling a dendrogram, where clusters and the cluster hierarchy are represented. The entries in the table indicate how many samples are contained in each level of the cluster hierarchy. Also, the most descriptive rules are listed. For instance, two clusters near the top of the hierarchy are described by the descriptive rules. The descriptive rules for the cluster A are

$$\text{Number of third party insurance (agriculture)} >= 0.5$$
$$\text{Contribution family accidents insurance policies} < 2.5$$

and for the cluster B, the descriptive rules are

$$\begin{aligned}\text{Contribution third party insurance (agriculture)} &< 1.5\\ \text{Contribution tractor policies} &< 1.5\\ \text{Contribution trailer policies} &< 1.5\end{aligned}$$

Majority of the customers (97 %) belong to the cluster B, also indicated in the tables. Focusing on the B clusters in the hierarchy, it is valuable to know how each cluster differs from its peer clusters (in effect, the clusters which have the same parent in the cluster hierarchy). The report indicates a single most discriminative rule according to which each peer cluster differs from the present cluster. In the case of sub-clusters of cluster B called B1 and B2, the discriminative rule dictates the B1 to have a significant rule

$$\text{Contribution family accidents insurance policies} >= 1$$

in contrast to the present cluster B2. Projection of the cluster-specific data complements this view. Navigating back to the top level, projection visualization as in the Figure 3 is presented.

As the description above demonstrates, working with the data survey report is highly interactive, and the implementation as a hypertext document greatly facilitates working with the report. The user navigates through the document searching for suggestive information to be used in the later stages of the data analysis process. It is important to realize that the data survey report serves as a initial tool to gain understanding through suggestive information on the structure of the data manifold. The results should be used as an initial step, and care should be taken in inferring knowledge from the report.

6 Conclusion

In the initial phases of a data analysis project, the data miner should have some perception of the data, or a mental model of it, to be able to formulate models in the latter phases of the project successfully. Helping to reach this goal, an implemented system for automatically creating data survey reports on numerical table-format data sets has been described. The system applies a set of generic data analysis algorithms — variable and variable relation analysis, projection, clustering, and cluster description algorithms — to the data and writes a report which can be used as the starting point for data understanding and as a reference later on. The system integrates linguistic descriptions (rules) and statistical measures with visualizations. Visualizations provide qualitative information of the data sets, and give an overview of the data. The visualizations also help in assessing the validity of the proposed measures, clusters and descriptive rules. The report provides a coherent, organized document about the properties of the data, making it preferable to applying the same algorithms to the original data sets in an unorganized manner and in different formats.

In our experience, the implemented system succeeds in automating a lot of the initial effort done in the beginning of a typical data analysis project. In fact, the system has been built on top of our experience in many collaborative data analysis projects with industrial partners involving real-world data. In all, we feel that the current version should have general appeal to a wide variety of projects and should help in gaining an initial understanding for successful modeling in many domains.

A System data

The first data set used throughout this chapter to illustrate the resulting data survey report is a simple 9-dimensional real-world data set. The *system data set* describes the operation of a single computer workstation in a networking environment. The workstation was used for daily activities of a research scientist ranging from computationally intensive (data analysis) tasks to editing of programs and publications. The data set has been collected by the authors at their research institution.

The number of variables recorded was 9. Four of the variables reflect the volumes of network traffic and five of them the CPU usage in relative measures. The variables for measuring the network traffic were `blks/s` (read blocks per second), `wblks/s` (written blocks per second), `ipkts` (the number of input packets) and `opkts` (the number of output packets). Correspondingly, the central processing unit activities were measured with variables `usr` (time spent in user processes), `sys` (time spent in system processes), `intr` (time spent handling interrupts), `wio` (CPU was idle while waiting for I/O), `idle` (CPU was idle and not waiting for anything). Whereas the network traffic is unconstrained (within reasonable bounds), the full capacity of the CPU is always divided between activities. Therefore, the five last measurements add up to the full, unit capacity of the CPU. In all, 1908 data vectors were collected.

References

1. Esa Alhoniemi, Jaakko Hollmén, Olli Simula, and Juha Vesanto, *Process Monitoring and Modeling Using the Self-Organizing Map*, Integrated Computer-Aided Engineering **6** (1999), no. 1, 3–14.
2. Stephen D. Bay and Michael J. Pazzani, *Detecting group differences: Mining contrast sets*, Data Mining and Knowledge Discovery **5** (2001), no. 3, 213–246.
3. Eric Boudaillier and Georges Hebrail, *Interactive Interpretation of Hierarchical Clustering*, Intelligent Data Analysis **2** (1998), no. 3.
4. Pete Chapman, Julian Clinton, Thomas Khabaza, Thomas Reinartz, and Rüdiger Wirth, *The CRISP-DM process model*, Tech. report, CRISM-DM consortium, March 1999, `http://www.crisp-dm.org`.

5. David L. Davies and Donald W. Bouldin, *A Cluster Separation Measure*, IEEE Trans. on Pattern Analysis and Machine Intelligence **PAMI-1** (1979), no. 2, 224–227.
6. Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern classification*, second ed., John Wiley & Sons, 2001.
7. Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim, *CURE: an efficient clustering algorithm for large databases*, Proceedings of SIGMOD International Conference on Management of Data (New York), ACM, 1998, pp. 73–84.
8. R. Hilderman and H. Hamilton, *Knowledge discovery and interestingness measures: A survey*, Tech. Report CS 99-04, Department of Computer Science, University of Regina, October 1999.
9. Johan Himberg, *Enhancing SOM-based data visualization by linking different data projections*, Intelligent Data Engineering and Learning (IDEAL'98) (L. Xu, L. W. Chan, and I. King, eds.), Springer, 1998, pp. 427–434.
10. Teuvo Kohonen, *Self-Organizing Maps*, 2nd ed., Springer Series in Information Sciences, vol. 30, Springer, Berlin, Heidelberg, 1995.
11. Andreas König, *A survey of methods for multivariate data projection, visualization and interactive analysis*, Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems (IIZUKA'98) (T. Yamakawa and G. Matsumoto, eds.), World Scientific, October 1998, pp. 55–59.
12. Krista Lagus and Samuel Kaski, *Keyword selection method for characterizing text document maps*, Proceedings of ICANN99, Ninth International Conference on Artificial Neural Networks, vol. 1, IEE, London, 1999, pp. 371–376.
13. R. S. Michalski and R. Stepp, *Automated construction of classifications: Conceptual clustering versus numerical taxonomy*, IEEE Transactions on Pattern Analysis and Machine Intelligence **5** (1983), 396–410.
14. G. Piatetsky-Shapiro and C. Matheus, *The interestingness of deviations*, Proceedings of KDD'94, July 1994, pp. 25–36.
15. Dorian Pyle, *Data Preparation for Data Mining*, Morgan Kaufmann Publishers, 1999.
16. Andreas Rauber and Dieter Merkl, *Automatic labeling of self-organizing maps: Making a treasure-map reveal its secrets*, Proceedings of the 3rd Pasific-Area Conference on Knowledge Discovery and Data Mining (PAKDD'99), 1999.
17. Markus Siponen, Juha Vesanto, Olli Simula, and Petri Vasara, *An approach to automated interpretation of SOM*, Proceedings of Workshop on Self-Organizing Map 2001 (Nigel Allinson, Hujun Yin, Lesley Allinson, and Jon Slack, eds.), Springer, June 2001, pp. 89–94.
18. A. Ultsch, *Self-organized feature maps for monitoring and knowledge acquisition of a chemical process*, Proceedings of International Conference on Artificial Neural Networks (ICANN) 1993, September 1993, pp. 864–867.
19. A. Ultsch, G. Guimaraes, D. Korus, and H. Li, *Knowledge extraction from artificial neural networks and applications*, Proceedings of Transputer-Anwender-Treffen / World-Transputer-Congress (TAT/WTC) 1993 (Aachen, Tagungsband), Springer Verlag, September 1993, pp. 194–203.
20. A. Ultsch and H. P. Siemon, *Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis*, Proceedings of International Neural Network Conference (INNC'90) (Dordrecht, Netherlands), Kluwer, 1990, pp. 305–308.
21. P. van der Putten and M. van Someren (eds.), *Coil challenge 2000: The insurance company case*, Tech. Report 2000-09, Leiden Institute of Advanced Computer Science, 2000.

22. A. Vellido, P.J.G Lisboa, and K. Meehan, *Segmentation of the on-line shopping market using neural networks*, Expert Systems with Applications **17** (1999), 303–314.
23. Juha Vesanto, *SOM-Based Data Visualization Methods*, Intelligent Data Analysis **3** (1999), no. 2, 111–126.
24. Juha Vesanto and Esa Alhoniemi, *Clustering of the Self-Organizing Map*, IEEE Transactions on Neural Networks **11** (2000), no. 2, 586–600.
25. Juha Vesanto and Mika Sulkava, *Distance matrix based clustering of the self-organizing map*, Proceedings of the Twelfth International Conference on Artificial Neural Networks (ICANN'02), 2002, To appear.
26. Colin Ware, *Information visualization: Perception for design*, Morgan Kaufmann Publishers, 2000.
27. Tian Zhang, Raghu Ramakrishnan, , and Miron Livny, *Birch: An efficient data clustering method for very large databases*, Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (Montreal, Canada), 1996, pp. 103–114.