1 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 prepare and model data successfully, the data miner needs to be aware of the properties of the data manifold. In this paper, the outline of a tool for automatically generating data survey reports for this purpose is described. The report combines linguistic descriptions (rules) and statistical measures with visualizations. Together these provide both quantitative and qualitative information and help the user to form a mental model of the data. The main focus is on describing the cluster structure and the contents of the clusters. The data is clustered using a novel algorithm based on the Self-Organizing Map. The rules describing the clusters are selected using a significance measure based on the confidence on their characterizing and discriminating properties.

1.1 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 [17]. Proper data preparation, selection of modeling tools and evaluation processes is only possible if the miner has a good overall idea, 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 — a specific collection of variables and samples — 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 repeatedly. Therefore, the tools used for data understanding should be as automated as possible.

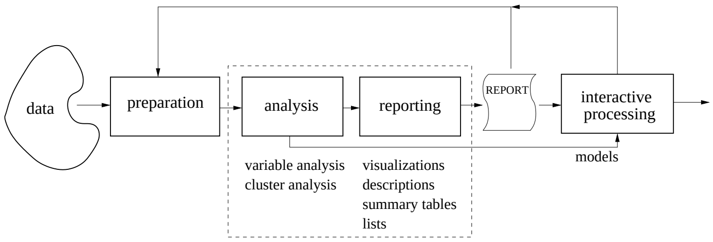


Figure1.1. Data understanding as an iterative process. The data is prepared and fed into the analysis system which generates the data survey report. Based on the findings in the report and possibly further insights based on interactive investigation, the data miner may either proceed with the next data mining phase, or prepare the data set better and, with a push of a button, make a new report. The area within the dashed box corresponds to the implemented system.

1.1.1 Automated analysis of table-format data

This paper 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 both new data sets and alternate 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 paper 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 1.1.

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

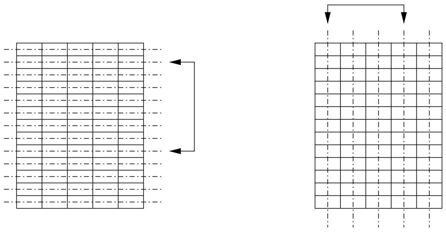


Figure1.2. Table-format data can be investigated both in terms of samples and in terms of variables. Samples are analyzed to find clusters or groups of similar items (Section 1.2). Variables are analyzed to find groups of related items (Section 1.3).

interesting depends on the problem and the data domain. In the implemented system, the domain has been restricted to unsupervised analysis of numerical table-format data, see Figure 1.2. 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.

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 a short overview for a quick look at the major properties of the data, followed by a more detailed description of each variable and cluster. The elements of the report are visualizations and numerical or linguistic descriptions organized 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 [21].

1.1.2 Related work

In [17], Pyle introduces the concept of data survey 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 paper. The clusters can be characterized by listing the variable values typical for each cluster using, for example, characterizing rules [8]. Another approach is to rank the variables in the order of significance [13,18,20]. In this paper, both approaches are used.

In [16], a report generation system KEFIR is described. It automatically analyses 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 paper, on the other hand, 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, for example between males and females. In this work, the data is numerical, and the two groups are always two clusters, or a cluster and the rest of the data. This simplifies the task considerably since the items in the clusters are always internally similar.

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 [15], 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 [7].

1.1.3 Organization and example data set

The paper is organized as follows. In Section 1.1, the methodology and basic motivation for the implemented system has been described. In Sections 1.2 and 1.3, the analysis methods for samples and variables, respectively, are described. In Section 1.4, the overall structure of the data survey report is explained. The work is summarized in Section 1.5.

Throughout the paper, a simple real-world data set is used to illustrate the elements of the report. It 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 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` (read blocks per second), `wblks` (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). In all, 1908 data vectors were collected.

1.2 Sample analysis

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

1.2.1 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 [12], but in the proposed system, a linear projection to the plane spanned by two principal eigenvectors is used. This is because of computational efficiency, and to be able 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 1.3a.

Like spatial coordinates, colors can also be used to encode similarity information [27,25,10]. Instead of similar positions, data samples close to each other get similar colors. While the resulting similarity encoding is not as accurate as the one produced by spatial projection, it is useful for linking multiple visualizations together, or when the position information is needed for other purposes. 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. These colors are used consistently in various visualizations throughout the report to indicate which cluster the sample belongs to.

1.2.2 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. This may represent the inherent structure of the data set better than a single direct partitioning. Equally important from data understanding point of view is that the hierarchy also allows the data to be investigated at several levels of granularity.

In the implemented system, the Self-Organizing Map (SOM) is used for clustering [11,26]. First, the data is quantized using a SOM with a few hundred map units. After this the map units are clustered. This is done based on the U-matrix [23] technique which 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, as done in [24]. We use an enhanced version based on region-growing:

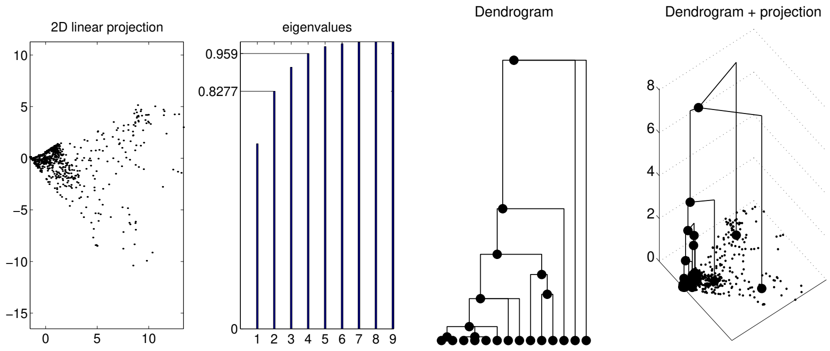
1. The local minima in the distance matrix are found.
2. Each local minima is set to be one cluster: $C_i = \{\mathbf{m}_i\}$. All other map units $\mathbf{m}_j$ are left unassigned.
3. The distances $d(C_i, \{\mathbf{m}_j\})$ from each cluster C_i to (the cluster formed by) each unassigned map unit $\mathbf{m}_j$ are calculated.
4. The unassigned map unit with smallest distance is found and assigned to the corresponding cluster.
5. Continue from step 3 until all map units have been assigned.

This procedure provides a partitioning of the map into a set of c base clusters, where c is the number of local minima on the distance matrix. Overall, this 2-phase strategy reduces the computational complexity of the clustering considerably because only a few hundred objects need to be clustered instead of all the original data samples [26]. In addition, the SOM is useful as a convenient projection of the data cloud, see Section 1.3.

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 $2c-1$ distinct 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. In the implemented system, the following procedure is applied:

1. Start from root cluster.
2. For the cluster under investigation, generate different kinds of sub-cluster sets by replacing each cluster in the sub-cluster set by its own sub-clusters.
3. Each sub-cluster set defines a partitioning of the data in the investigated cluster. Investigate each partitioning using Davies-Bouldin index:
$$I_{DB} = \frac{1}{n} \sum_{i=1}^{n} \max_{j \neq i} \left\{ \frac{\Delta_i + \Delta_j}{d(i,j)} \right\}, \tag{1.1}$$
where n is the number of clusters in the sub-cluster set, Δ_i is internal dispersion of sub-cluster i and $d(i,j)$ is the distance between sub-clusters i and j [5]. The index gets low values for well-separated and compact clusters.
4. Select the sub-cluster set with minimum value for I_{DB}, and prune the corresponding intermediate clusters.
5. Select an uninvestigated and unpruned cluster, and continue from step 2.

In Figure 1.3b, the clusters and cluster hierarchy are presented using dendrograms which are linked with the projection results using projection coordinates. In the actual report, colors are used to provide a much more efficient linking.



(a) (b)

Figure1.3. Projection visualization (a) and cluster hierarchy (b). The projection gives an idea of the shape of the data manifold. It is accompanied by the scree plot of cumulative eigenvalues. This gives 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. In (b), the cluster hierarchy is visualized using a dendrogram. The figure on the right links the dendrogram to the projection visualization by showing the dendrogram starting from the 2-dimensional projection coordinates. In the actual report, the linking is further enhanced by coloring each data sample with the color of the (base) cluster it belongs to.

1.2.3 Cluster characterization

Descriptive statistics — for example means, standard deviations and histograms of individual variables — can be used to list the values typical for each cluster. Not all variables are equally interesting or important, however. Interestingness can be defined as deviation from the expected [16,9]. It can be measured for each cluster as the difference between distributions of variable values in the cluster versus the whole data either using probability densities or some more robust measures, for example standard or mean deviation. 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 [22] to describe the values in each cluster:

$$R_i : \mathbf{x} \in C_i \Leftrightarrow x_k \in [\alpha_k, \beta_k] \tag{1.2}$$

where x_k is the value of variable k in the 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 concern 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.

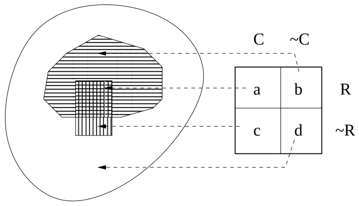


Figure1.4. The correspondence between cluster C and rule R. The cluster is the vertically shaded area, and the rule (or classification model) is the horizontally shaded area. On the right is the corresponding confusion matrix: a is the number of samples which are in the cluster, and for which the rule is true. In contrast, d is the 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 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 rule R_i can be divided to two different cases, a characterizing and a differentiating rule:

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

The validity of R_i 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 rules — in effect to select the variables and associated values for α_k and β_k for each rule — one can use statistics of the values in the clusters, as in [22]. 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 miss-classifications, for example:

$$s_1 = \frac{a+d}{a+b+c+d}, \tag{1.3}$$

$$s_2 = \frac{a}{a+b}\frac{a}{a+c}, \tag{1.4}$$

$$s_3 = \frac{a}{a+b+c}, \tag{1.5}$$

where a, b, c and d are from the confusion matrix in Figure 1.4.

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 can be interpreted as mutual confidence, since it 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, the advantage of s_3 is 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.

The report elements to describe the clusters are shown in Figure 1.5 and Table 1.1. The former shows the most significant combined 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.

1.3 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. In the implemented system, 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. However, this 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 each marker is defined by the value of a selected variable in the

Table1.1. Cluster summary for the cluster depicted below. First column is the name of the variable, followed by its statistics in the cluster, the corresponding rule, and validity measures for the rule. The columns under 'Single' show the validity of a single-variable rule. The variables are in the order of decreasing s_2 value. The columns under 'Cumulative' show the validity of the rule formed as conjunction of several variables. The most significant conjunctive rule (with $s_2 = 0.93$) consists of five variables `intr`, `idle`, `usr`, `blks` and `wio`.

Name	**Min**	**Mean±std**	**Max**	**Rule**	P^d	P^c	s_2	P^d	P^c	s_2
	Statistics				Single			Cumulative		
intr	0.98	1.8±0.36	2.9	$\in [0.78, 3.1]$	77%	97%	0.75	77%	97%	0.75
idle	2.9	3.9±0.2	4.5	$\in [3.3, 4.3]$	65%	94%	0.62	88%	94%	0.83
usr	0.4	1.5±0.23	2.9	$\in [1.2, 2.3]$	62%	95%	0.59	91%	94%	0.86
sys	0.42	0.82±0.26	2.3	$\in [0.71, 1.4]$	70%	60%	0.42			
opkts	0.03	0.087±0.29	2.9	< 2.9	20%	99%	0.2			
ipkts	0.038	0.098±0.26	2.4	< 2.9	20%	99%	0.2			
blks	0.092	0.19±0.21	1.3	< 1.6	20%	97%	0.2	97%	94%	0.91
wio	0	0.12±0.28	2	< 1.1	20%	96%	0.19	99%	93%	0.93
wblks	0.25	0.29±0.1	0.75	< 1.4	19%	97%	0.19			

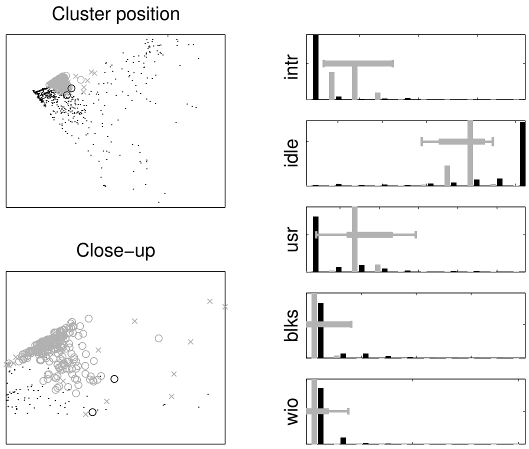


Figure1.5. Visualization of the most significant rule for a cluster. Samples belonging to the cluster are drawn with gray, while all other samples are black. On the left, the projection of the data (an overview and a close-up). The samples for which the rule is true are indicated with marker 'o'. The samples which belong to the cluster but for which the rule is not true are indicated with marker 'x'. In this case, the values for a, b, c and d are 324, 2, 23 and 1559, respectively. On the right are the histograms of the five variables involved in the most significant rule: their distribution in the cluster and in the rest of the data are indicated with vertical bars. The horizontal bar shows the range of the rule (thick part) and the range of values in the cluster (narrow part).

corresponding data sample. By using one component plane for each variable, the whole data set can be visualized. Relationships between variables can be seen as similar patterns in identical places on the component planes.

In the implemented system, the component planes are based on the SOM rather than the original data. This is because the quantization performed by the SOM reduces the effect of noise and outliers, and the SOM projection focuses locally such that the behavior of the data can be seen irrespective of the local scale, see Figure 1.6a. However, care must be taken since the perceived correlations may actually be a product of quantization rather than a true dependency [14].

A more quantitative measure of the dependency between pairs of variables is the correlation coefficient. It is robust, computationally light and can be efficiently visualized as a matrix, see Figure 1.6c. Selected correlations are also visualized on an association graph, see Figure 1.6b. In the implemented system, the correlation coefficients are also used as feature vectors for each variable. Using them, the variables are clustered, and ordered, to indicate groups of dependent variables [25]. In all three visualizations in Figure 1.6, the variables have been ordered such that related variables are near each other.

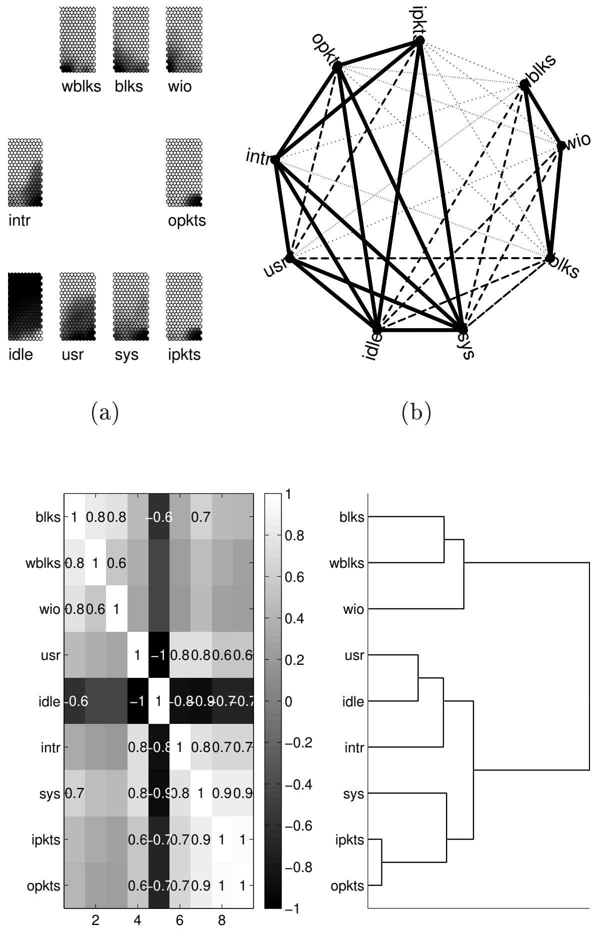
1.4 Data survey report

The current implementation of the system is a Matlab script which has been installed on a web server as a cgi-bin script. The user just provides the data, and the analysis is performed automatically. The resulting data survey report is provided both as a hypertext document in HTML and in printable form in Postscript. In this paper, only some examples of the report contents are shown. However, the actual report with full details and color figures can be found from: `http://www.cis.hut.fi/juuso/datareport/`.

1.4.1 The report structure

The report starts with an overview part, where the top-level results of both variable and cluster analysis are shown. The overview provides a quick look at the properties of 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 paper (Figures 1.3 and 1.6). From Figure 1.3a, one can see that the 2-dimensional projection preserves the structure of the system data set very well. Most of the data is tightly packed in a single region, and the rest of the data is dispersed along two main directions. From Figure 1.3b, one can see that there are three main operational states in the system, one of which divides further to several sub-states. From Figure 1.6 one can see that there



(c)

Figure1.6. (a) Component planes with black and white corresponding to high and low values, respectively. Relationships between variables can be seen as similar patterns in identical places on the component planes. (b) Association graph. On the association graph, high positive (≥ 0.75) and high negative (≤ -0.75) correlations are shown with thick lines. The thinner dashed lines indicate smaller coefficients. (c) Correlation coefficient matrix, and the dendrogram resulting from clustering the variables using an agglomerative clustering algorithm. In each figure, the variables have been automatically ordered such that variables closely related in terms of the correlation coefficients are close to each other.

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 characterizations of individual clusters, both their internal properties and their differences to closest neighboring clusters. These allow the reader to get immediately some further information on interesting details. For example, the cluster depicted in Table 1.1 and Figure 1.5 is mainly characterized by relatively high level of interruptions and user operations, very low amount of disk operations, and a moderate level of idle time.

1.4.2 Computational load

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: $\mathcal{O}(cd)$. The computational complexity of variable analysis is $\mathcal{O}(d^3)$ because of clustering the variables. Since typically $n \gg m \gg c$, the overall complexity of the analysis is $\mathcal{O}((nm+d^2)d)$. For the example data set, the analysis takes about 0.5 minutes on a Linux workstation with Pentium II 350 MHz processor and 256 MB of memory.

1.5 Summary and conclusions

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 dependency analysis, projection, clustering, and cluster description algorithms — to the data and writes a data survey 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 and incoherent manner.

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 [1,19]. 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.

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(1):3–14, 1999.
2. Stephen D. Bay and Michael J. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, July 2001.
3. Eric Boudaillier and Georges Hebrail. Interactive Interpretation of Hierarchical Clustering. *Intelligent Data Analysis*, 2(3), August 1998.
4. Pete Chapman, Julian Clinton, Thomas Khabaza, Thomas Reinartz, and Rüdiger Wirth. The CRISP-DM process model. Technical 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(2):224–227, April 1979.
6. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley & Sons, second edition, 2001.
7. Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: an efficient clustering algorithm for large databases. In *Proceedings of SIGMOD International Conference on Management of Data*, pages 73–84, New York, 1998. ACM.
8. Jiawei Han, Yandong Cai, and Nick Cercone. Knowledge discovery in databases: An attribute-oriented approach. In Li-Yan Yuan, editor, *Proceedings of the 18th International Conference on Very Large Databases*, pages 547–559, San Francisco, U.S.A., 1992. Morgan Kaufmann Publishers.
9. R. Hilderman and H. Hamilton. Knowledge discovery and interestingness measures: A survey. Technical Report CS 99-04, Department of Computer Science, University of Regina, October 1999.
10. Johan Himberg. A SOM based cluster visualization and its application for false coloring. In *Proceedings of International Joint Conference in Neural Networks (IJCNN) 2000*, Como, Italy, 2000.

11. Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 3rd edition, 1995.
12. Andreas König. A survey of methods for multivariate data projection, visualization and interactive analysis. In T. Yamakawa and G. Matsumoto, editors, *Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems (IIZUKA'98)*, pages 55–59. World Scientific, October 1998.
13. Krista Lagus and Samuel Kaski. Keyword selection method for characterizing text document maps. In *Proceedings of ICANN99, Ninth International Conference on Artificial Neural Networks*, volume 1, pages 371–376. IEE, London, 1999.
14. Jouko Lampinen and Timo Kostiainen. *Recent advances in self-organizing neural networks*, chapter Generative probability density model in the Self-Organizing Map. Springer Verlag, To appear.
15. R. S. Michalski and R. Stepp. Automated construction of classifications: Conceptual clustering versus numerical taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:396–410, 1983.
16. G. Piatetsky-Shapiro and C. Matheus. The interestingness of deviations. In *Proceedings of KDD'94*, pages 25–36, July 1994.
17. Dorian Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, 1999.
18. Andreas Rauber and Dieter Merkl. Automatic labeling of self-organizing maps: Making a treasure-map reveal its secrets. In *Proceedings of the 3rd Pasific-Area Conference on Knowledge Discovery and Data Mining (PAKDD'99)*, 1999.
19. Olli Simula, Jussi Ahola, Esa Alhoniemi, Johan Himberg, and Juha Vesanto. *Kohonen Maps (E. Oja and S. Kaski, eds.)*, chapter Self-Organizing Map in Analysis of Large-Scale Industrial Systems. Elsevier, 1999.
20. Markus Siponen, Juha Vesanto, Olli Simula, and Petri Vasara. An approach to automated interpretation of SOM. In Nigel Allinson, Hujun Yin, Lesley Allinson, and Jon Slack, editors, *Proceedings of Workshop on Self-Organizing Map 2001*, pages 89–94. Springer, June 2001.
21. Edward Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.
22. A. Ultsch, G. Guimaraes, D. Korus, and H. Li. Knowledge extraction from artificial neural networks and applications. In *Proceedings of Transputer-Anwender-Treffen / World-Transputer-Congress (TAT/WTC) 1993*, pages 194–203, Aachen, Tagungsband, September 1993. Springer Verlag.
23. A. Ultsch and H. P. Siemon. Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis. In *Proceedings of International Neural Network Conference (INNC'90)*, pages 305–308, Dordrecht, Netherlands, 1990. Kluwer.
24. A. Vellido, P.J.G Lisboa, and K. Meehan. Segmentation of the on-line shopping market using neural networks. *Expert Systems with Applications*, 17:303–314, 1999.
25. Juha Vesanto. SOM-Based Data Visualization Methods. *Intelligent Data Analysis*, 3(2):111–126, 1999.
26. Juha Vesanto and Esa Alhoniemi. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks*, 11(2):586–600, March 2000.
27. Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, 2000.