Exploring Conceptual Similarities of Objects for Analyzing Inconsistencies in Relational Databases

Gerd Stumme

Technische Universität Darmstadt, Fachbereich Mathematik, Schloßgartenstr. 7,
D–64289 Darmstadt; stumme@mathematik.tu-darmstadt.de

1 Introduction

On-Line Transaction Processing (OLTP) systems gather huge amounts of automatically generated data, often without sufficient integrity checking. Hence, during the knowledge discovery process, one discovers inconsistencies which contradict to constraints that have never been made explicit, but which exist only as personal knowledge of the analyst.

Conceptual Information Systems support the detection of such contradictions to expert knowledge which is not explicitly given. They provide a multidimensional conceptually structured view on data stored in relational databases ([5], [10]). Conceptual Information Systems are similar to On-Line Analytical Processing (OLAP) tools, but focus on qualitative (i.e. non-numerical) data ([6]). The analog to OLAP dimensions are *conceptual scales*. Each conceptual scale represents a conceptual hierarchy describing the semantics of the range of values for one or more attributes of the database scheme. The conceptual scales are visualized by so-called *Hasse diagrams* which indicate the subconcept-superconcept hierarchy on the concepts. Unlike statistical data mining tools, Conceptual Information Systems always present information about the individual objects. Hence the existence of outliers is not hidden in numerical values as, e. g., standard deviations. Conceptual Information Systems rely on the mathematical theory of *Formal Concept Analysis* ([12], [3]) which provides a formalization of the concept of 'concept'. It reflects an understanding of 'concept' which is first mentioned explicitly in the Logic of Port Royal in 1668 ([1]) and has been established in the German standards DIN 2330 and DIN 2331.

If inconsistencies have been discovered in the knowledge discovery process, then the task is to examine their causes. In cases where the detection of such inconsistencies was possible only by human interaction, it is very unlikely that the cause will be discovered by a fully automatic tool. Hence we promote a knowledge discovery support environment that supports a human-centered discovery process. In this paper we present the state of research about extending Conceptual Information Systems such that they support the process of analyzing possible causes for such inconsistencies.

2 Conceptual Information Systems

Concepts are necessary for expressing human knowledge. Therefore, the knowledge discovery process benefits from a comprehensive formalization of concepts which can be activated to represent knowledge coded in databases. *Formal Concept Analysis* ([12], [3]) offers such a formalization by mathematizing concepts which are understood as units of thought constituted by their extension and intension ([11]). For allowing a mathematical description of extensions and intensions, Formal Concept Analysis always starts with a *formal context*.

Definition. A *(formal) context* is a triple $\mathbb{K} := (G, M, I)$ where G and M are sets and I is a relation between G and M. The elements of G and M are called *objects* and *attributes*, respectively, and $(g, m) \in I$ is read *"the object g has the attribute m"*.

A *(formal) concept* is a pair (A, B) such that $A \subseteq G$, $B \subseteq M$ are maximal with $A \times B \subseteq I$. The set A is called the *extent* and the set B the *intent* of the concept (A, B). The *subconcept-superconcept-relation* is formalized by $(A_1, B_1) \leq (A_2, B_2) :\Longleftrightarrow A_1 \subseteq A_2$ $(\Longleftrightarrow B_1 \supseteq B_2)$. The set of all concepts of a context (G, M, I) together with this order relation is always a complete lattice, called the *concept lattice* of (G, M, I) and denoted by $\underline{\mathfrak{B}}(G, M, I)$.

Example 1. The cross-table in Figure 1 represents a formal context about the gates of Terminal 1 at Frankfurt Airport. The object set G comprises the gates, the attribute set M four different functionalities. The binary relation I is represented by the crosses. A cross at cell (g, m) (i.e., $(g, m) \in I$) means 'gate g has functionality m'.

	Terminal Gate	Bus Gate	Domestic Gate	International Gate
A1, B2, C1	X	X	X	
A2-5, A8, A9, B1, B3-9		X	X	
A10-21, A23	X		X	X
A22		X	X	X
B10	X		X	
B11-16, B30-35, B90, B91, C3, C10, C21-23		X		X
B20, B22-28, B41-48, C4-9	X			X
C2	X	X		X

Fig. 1. A formal context concerning gates at Frankfurt Airport

In the line diagram of the concept lattice in Fig. 2, each circle stands for a formal concept. The subconcept-superconcept hierarchy can be read by following

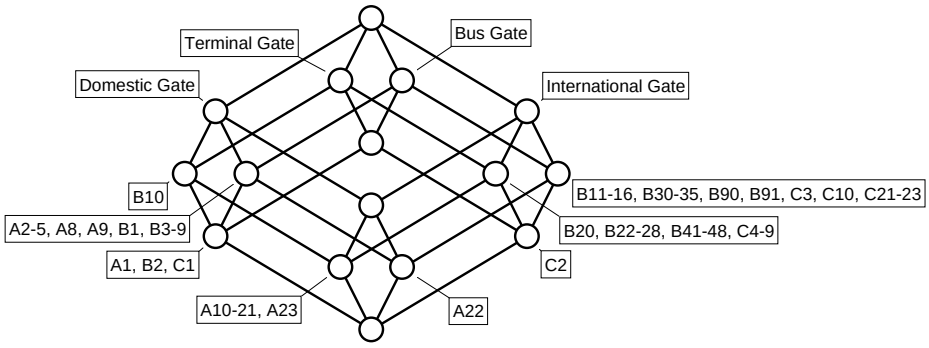

Fig. 2. The concept lattice of the formal context in Fig. 1

ascending paths of straight line segments. The intent [extent] of each concept is given by all attribute labels [object labels] reachable from that context by ascending [descending] paths of straight line segments. Hence, an object has an attribute if and only if they are linked by an ascending path. For instance, the leftmost concept in the diagram (which is labeled by B10) is the concept

$$(\{\text{B10}, \text{A1}, \text{B2}, \text{C1}\}, \{\text{Domestic Gate}, \text{Terminal Gate}\}) \ .$$

The fact that the lattice is a four-dimensional hypercube indicates that there are no implications between the four attributes. In general, the visualization by line diagrams supports the study of dependencies between the attributes.

In most applications, there are not only Boolean attributes in the databases. The conceptual model we use for this is a *many-valued context*. In database terms, a many-valued context is a relation of a relational database with one key attribute whose domain is the set G of objects. Throughout this paper, we consider only one (denormalized) database relation at a time.

In order to obtain a concept lattice from a many-valued context, it has to be translated into a one-valued (formal) context. The translation process is described by *conceptual scales*:

Definition. A *many-valued context* is a tuple $(G, M, (W_m)_{m \in M}, I)$ where G and M are sets of *objects* and *attributes*, resp., W_m is a set of *values* for each $m \in M$, and $I \subseteq G \times \bigcup_{m \in M}(\{m\} \times W_m)$ is a relation such that $(g, m, w_1) \in I$ and $(g, m, w_2) \in I$ imply $w_1 = w_2$.

A *conceptual scale* for an attribute $m \in M$ is a one-valued context $\mathbb{S}_m := (G_m, M_m, I_m)$ with $W_m \subseteq G_m$. The context $\mathbb{R}_m := (G, M_m, J_m)$ with $gJ_mn :\iff \exists w \in W_m\colon (g, m, w) \in I \wedge (w, n) \in I_m$ is called the *realized scale* for the attribute $m \in M$.

A *Conceptual Information System* consists of a many-valued context together with a set of conceptual scales.

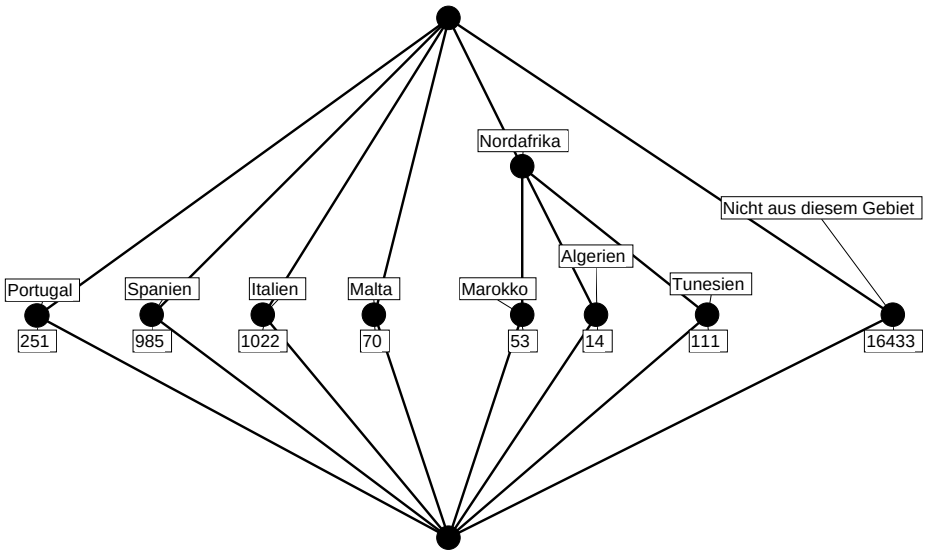

Fig. 3. Realized scale *Destination Southern Europe/Mediterranean Sea*

The set M consists of all attributes of the database scheme, while the sets M_m contain the attributes which are shown to the user by the Conceptual Information System.

Example 2. The information system INFO-80 supports planning, realization, and control of business transactions related to flight movements at Frankfurt Airport. A Conceptual Information System has been established in order to facilitate access to the data of INFO-80 ([4]).

Here we consider 18389 outbound flight movements effectuated at Frankfurt Airport during one month. For each of these flight movements 110 attributes are registered automatically and stored in a database (e. g., destination, estimated time of departure, actual time of departure, number of passengers, terminal position, ...). 77 conceptual scales have been designed (some of the 110 database attributes were of minor importance for data analysis). By combining the scales and zooming into them with other scales, the analyst can explore dependencies and irregularities.

Figure 3 shows the realized scale *Destination Southern Europe/Mediterranean Sea*. Because of the large number of flight movements, TOSCANA displays only the number of flight movements assigned to each concept. If desired, the user can drill-down to the flight number and to more detailed information. For instance, the concept labeled by 'Marokko' (Morocco) has 53 objects (i.e., flight numbers) in its extent, and the attributes 'Marokko' and 'Nordafrika' (Northern

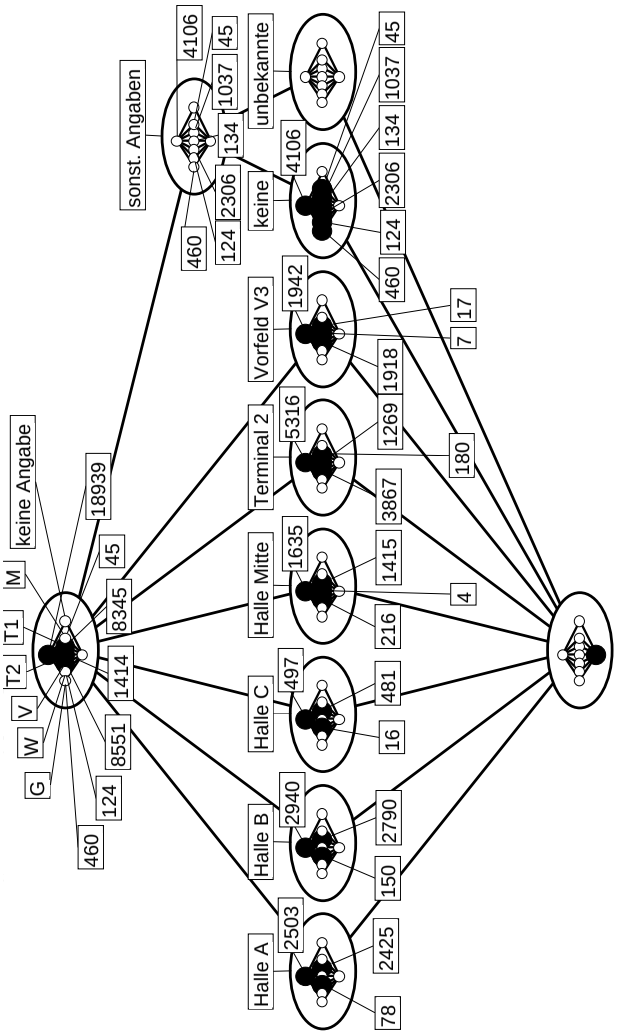

Fig. 4. Nested line diagram of the scales *Position of Baggage Conveyor* and *Position of Aircraft*

Africa) in its intent. Its upper neighbor (which is labeled by 'Nordafrika') has $53 + 14 + 111 = 178$ objects in its extent, and 'Nordafrika' in its intent.

For analyzing how objects are distributed over two different scales, one can combine the two conceptual scales. The result of combining the two conceptual scales *Position of Baggage Conveyor* and *Position of Aircraft* is displayed in the nested line diagram in Figure 4.

Each of the 17 lines of the outer scale represents seven parallel lines linking corresponding concepts of the inner scale. The concept lattice we are interested in is embedded in this *direct product*. The embedding is indicated by the bold circles. Each non-realized concept (i.e., the empty circles) indicates an implication. For example, the leftmost circle in the leftmost ellipse has the attribute 'Halle A' (hall A) and 'G' (general aviation) in its intent. These two attributes form the premise of the implication. The conclusion of the implication is given by the intent of the largest realized concept below. In this case, it is the bottom concept which has all attributes in its extent. Hence hall A as position of the baggage conveyor and the general aviation apron as position of the aircraft implies everything else. This means that the premise is not realized by any object. There are no general aviation aircraft with a baggage conveyor assigned at hall A. There are concepts in the nested line diagram which are expected to be non-realized as well. For instance, an aircraft at Terminal 1 should not have a baggage conveyor assigned at Terminal 2. But in the diagram, we see that the concept with these two attributes is realized. There are 180 flight movements having the attribute 'Terminal 2' of the outer scale and the attribute 'T1' of the inner scale.

We can detect some more unnormal combinations. There are four aircraft that docked at Terminal 2, while their assigned baggage conveyors are at Terminal 1. To seven aircraft at Terminal 2 and 17 aircraft at Terminal 1, conveyors on the 'Vorfeld V3' (Apron V3) were assigned. In all these cases, one can drill down to the original data by clicking on the number to obtain the flight movement numbers, which in turn lead to the data sets stored in the INFO-80 system.

The knowledge that the combinations are unnormal, is not coded explicitly in INFO-80, but is only implicitly present as expert knowledge. Since these combinations happen at the airport, they cannot be forbidden by database constraints. But once such conflicts are discovered, one can implement an alarm which informs the operator of the baggage transportation system about such cases.

If there are too many scales involved in the knowledge discovery process, then zooming into the outer scale reduces the size of the displayed diagram. For instance, by zooming into the concept labeled by 'Terminal 2' in Fig. 4, we obtain the diagram of the conceptual scale *Position of Aircraft*, but with the objects being restricted to those flight movements where the assigned baggage conveyor is at Terminal 2. Then one can continue the exploration by adding a new conceptual scale on the inner level. This interactive human-centered knowledge discovery process is described in detail in [8].

3 Exploring Conceptual Similarities

What are the reasons for the mismatches between the aircraft positions and the assigned baggage conveyors for certain flight movements? By adding a new conceptual scale and zooming into the concepts in Fig. 4 where these flight movements are grouped together, the analyst can examine the properties of these flight

movements with regard to the new scale. But he has still the choice between 75 scales, whereof most are probably irrelevant for the analysis. Hence we need a ranking of the scales which tells the analyst which scales are probably the most interesting.

One approach is to determine, in each scale, all attributes that all these flight movements have in common. The scales can then be listed according to the number of these attributes. For instance, for the 180 flight movements with aircraft positions at Terminal 1 and baggage conveyors at Terminal 2, we obtain 27 common attributes in 21 scales out of the total of 731 attributes of the 77 scales. These attributes obviously include 'Terminal 2' in the scale *Position of Baggage Conveyor* and 'T1' in the scale *Position of Aircraft*. While these attributes provide no new information to the user, there are also more interesting attributes as, e. g., 'none' in the scale *Number of mail pallets*, or 'last airport in the EU' in the scale *All Informations in [PCG]*. The computation and representation of the attributes common to all objects is discussed in more detail in [7].

The result of retrieving all attributes common to all 180 flight movements provides interesting insights in the situation (e. g., that the destination is the last airport in the EU for all these flights), but is not sufficient for explaining the mismatch. One difficulty of this approach is that we require that *all* objects must have the attributes to be returned. Attributes that do not pertain to all, but only to most of the objects are not shown to the analyst. The second drawback is that even if an attribute pertains to all objects, it may be relatively irrelevant. This is the case if almost all objects in the database have that attribute. For instance, 18811 of the 18939 outbound flight movements have the attribute 'none' in the scale *Number of Mail Pallets*. This does not mean that the information that none of the 180 aircraft has mail pallets on board is irrelevant for the analysis, but it provides an estimation of its significance.

Hence, beside the percentages of the common attributes, another criterion for a ranking of the scales is needed, which indicates the difference of the distribution of the sample set to the distribution of the total set of objects. Once we have a measure for this difference, we can rank those scales where the sample set of objects has attributes in common, thus measuring the significance of these attributes. The second possibility is to rank *all* scales according to that measure, even if they do not provide common attributes.

For the 180 flight movements with aircraft positions at Terminal 1 and baggage conveyors at Terminal 2, we will for instance discover that the scale *Destination Southern Europe/Mediterranean Sea* provides interesting insights: Among the 180 flight movements, there are 87 with destination Italy[1], hence much more than estimated from the distribution of all objects which is shown in Fig. 3. For all of the three different measures that we discuss next, this scale appears on prominent positions in the rankings.

[1] Two of the remaining 93 flight movements have destinations in Spain, two in Morocco, three in Tunisia, and 86 are out of the region Southern Europe/Mediterranean Sea.

4 Measuring Differences of Distributions over Conceptual Scales

The three measures that we discuss are based on the χ^2 distance function

$$Q(x_1, \ldots, x_n; y_1, \ldots, y_n) := \sum_{i=1}^{n} \left(\frac{x_i}{\sum_{j=1}^{n} x_j} - \frac{y_i}{\sum_{j=1}^{n} y_j} \right)^2 .$$

For each scale in the Conceptual Information System, we compute such a value (with the x_i and y_j as defined below), and rank the scales in descending order of these values.

1. The first measure, Q_1, is obtained by letting n be the number of concepts of the scale to be measured, x_i the cardinality of the extent of the ith concept in the sample set, and y_i the cardinality of the extent of the ith concept in the set G of all objects.
2. The second measure, Q_2, is defined as Q_1 with the exception that the top element of the scale is not regarded (i.e., we have that n is the number of concepts in the scale minus 1).
3. The third measure, Q_3, is again similar to Q_1, but now, the x_i and y_j are set to the cardinalities of the contingents rather than of the extents. The *contingent* of a concept $\mathfrak{c}$ is the set of all objects which belong to the extent of $\mathfrak{c}$ but not to the extent of any proper subconcept of $\mathfrak{c}$.

There are arguments for and against each of these three measures: *i*) The first measure involves the difference in each concept. This seems the most natural approach. *ii*) Because the top concept of a scale always contain all objects, the distribution of the sample set and the total set is always equal in this concept. This is the argument for studying the modification Q_2 of Q_1. *iii*) The standard in comparing distributions is to consider classifications of the objects in disjoint classes. This is the case for contingents, while extents share objects. This is the reason for introducing the third measure. However, the third measure loses information about the structure of the conceptual scale, since its measure is always equal to the corresponding nominal scale. Variations of the distribution which appear only at a very specialized level of concepts are thus considered to be more important than by the first two measures.

The research for determining which of the measures should be preferred is still in progress. In sample applications, the different rankings have to be computed and evaluated by experts of the field. These experts have to decide which of the rankings reflects best their intuitive understanding of interestingness. We have applied all three measures to the sample set of 180 flight movements. In this paper, we discuss some observations that can be made in the first experiments:

The rankings obtained by the first two measures do not differ significantly. The ranking of Q_3 gives scales with flat hierarchies a preference compared to the other two. However, all three rankings are similar in a global sense: They provide local differences, but in the whole they consider the same scales to be

important. The scales an analyst considers to be potentially significant are all among the top twenty in each of the rankings.

In the flight movement database, there is no information coded about the semantics of the database attributes. Hence the algorithm can not distinguish whether a high score indicates a cause or a consequence of the examined situation. For instance, the scale *Area Control Group* has a high score in all three rankings. This is a consequence of the fact that by fixing the aircraft position the area control group is determined as well. If the data model provides such information (e. g., by functional dependencies), then one could use it for filtering the ranking. This feature cannot be applied to the presented system, because the flight movement database is just a flat table.

Concluding this section, we want to emphasize again that our purpose is not testing hypotheses or providing statements about dependencies. This is the task of a human expert. The analyst is free to look at any scale he wants. The system only supports him by providing suggestions where to look first. This approach combines a systematic investigation with the freedom to navigate around.

5 Outlook

As already mentioned in the last section, field studies have to be done in order to compare the different measures. Once a measure is decided to be the most suitable, this application should be implemented in the management system TOSCANA for Conceptual Information Systems. The implementation can be combined with a *highlighting of interesting concepts* as proposed in [9]. This highlighting marks those concepts of a conceptual scale which contribute overproportionally to the sum of squared differences. Further research should also include the exploitation of information provided by the conceptual data model.

References

1. A. Arnauld, P. Nicole: La logique ou l'art de penser — contenant, outre les règles communes, plusieurs observations nouvelles, propres à former le jugement. 3^o édit. reveüe & augm. P., Ch. Saveux, 1668
2. R. Cole, P. Eklund, B. Groh: Scalability in Formal Concept Analysis. *J. on Computational Intelligence* (to appear)
3. B. Ganter, R. Wille: Formale Begriffsanalyse: Mathematische Grundlagen. Springer, Heidelberg 1996
4. U. Kaufmann: *Begriffliche Analyse von Daten über Flugereignisse — Implementierung eines Erkundungs- und Analysesystems mit TOSCANA.* Diplomarbeit, TU Darmstadt, 1996
5. W. Kollewe, M. Skorsky, F. Vogt, R. Wille: TOSCANA — ein Werkzeug zur begrifflichen Analyse und Erkundung von Daten. In: R. Wille, M. Zickwolff (eds.): *Begriffliche Wissensverarbeitung — Grundfragen und Aufgaben.* B.I.–Wissenschaftsverlag, Mannheim 1994

6. G. Stumme: On-Line Analytical Processing with Conceptual Information Systems. *Proc. of the 5th Intl. Conf. of Foundations of Data Organization.* Kobe, November 12-14, 1998 (to appear)
7. G. Stumme: Dual Retrieval in Conceptual Information Systems. *Proc. of the Conf. Datenbanksysteme in Büro, Technik und Wissenschaft.* Freiburg, March 1–3, 1999 (submitted)
8. G. Stumme, R. Wille, U. Wille: Conceptual Knowledge Discovery in Databases Using Formal Concept Analysis Methods. In: J. M. Żytkow, M. Quafofou (eds.): *Principles of Data Mining and Knowledge Discovery.* Proc. of the 2nd European Symposium on PKDD '98, Lecture Notes in Artificial Intelligence **1510**, Springer, Heidelberg 1998, 450–458
9. G. Stumme, K. E. Wolff: Computing in conceptual data systems with relational structures. In: *Proc. Intl. Workshop on Data Warehousing and Data Mining*, Singapore, November 19–20, 1998, Springer, Heidelberg 1998 (to appear)
10. F. Vogt, R. Wille: TOSCANA – A graphical tool for analyzing and exploring data. In: R. Tamassia, I. G. Tollis (eds.): *Graph Drawing '94*, Lecture Notes in Computer Sciences **894**, Springer, Heidelberg 1995, 226–233
11. H. Wagner: Begriff. In: H. M. Baumgartner, C. Wild (eds.): *Handbuch philosophischer Grundbegriffe.* Kösel Verlag, München 1973, 191–209
12. R. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets.* Reidel, Dordrecht–Boston 1982, 445–470