Inductive Databases as Ranking

Taneli Mielikäinen

HIIT Basic Research Unit
Department of Computer Science
University of Helsinki, Finland
Taneli.Mielikainen@cs.Helsinki.FI

Abstract. Most of the research in data mining has been focused on developing novel algorithms for specific data mining tasks. However, finding the theoretical foundations of data mining has recently been recognized to be even greater concern to data mining.
One promising candidate to form a solid basis for data mining is known as inductive databases. The inductive databases are databases with a tight integration to data mining facilities. However, it is not clear what inductive databases actually are, what they should be and whether inductive databases differ notably from the usual databases with slightly broader notions of queries and data objects.
In this paper we aim to show that the viewpoint offered by inductive databases differs from the usual databases: the inductive databases can be seen as databases with ability to rank data manipulation operations. We describe how several central data mining tasks can be naturally defined by this approach and show that the proposed inductive databases framework offers conceptual benefits by clarifying and unifying the central data mining tasks. We also discuss some challenges of inductive databases based on query ranking and grading.

1 Introduction

Data mining aims to extract useful knowledge from large collections of (observational) data [1,2]. The largest effort in data mining has been devoted on developing novel methods to reach for several correctives of this goal. For example, there has been considerable amount of activity to invent techniques for pattern discovery, classification, clustering and density estimation.

However, although the general goal of data mining is not very well-defined, much less work has been devoted on finding out what that goal actually means, i.e., whether there exists the "theory of data mining" that captures the central tasks but is not too broad, essentially an empty concept. Recently this has increasingly been recognized to be one of the most important current challenges for the data mining community.

Although the main focus in data mining research has been in the development of data analysis algorithms, there has also been some work on examining possibilities to theoretical foundations of data mining [2,3]. There are two main requirements for the theory of data mining. First, it should capture most of the central data mining tasks naturally. Second, it should clarify and unify different data mining tasks. Thus, the theory of data mining should be useful in practice but still the theory should actually say something about the commonalities of the vast methods known in data mining and support the knowledge discovery process.

One promising candidate to serve as the theory of data mining is known as *inductive databases* [4,5]. The basic idea of inductive databases is to extend database technologies in such way that it enables the databases to support data mining and the knowledge discovery process. However, it is not clear what this actually means. In this paper try to clarify this by suggesting that the most essential difference of inductive databases compared to usual databases is the ability to rank or grade queries, i.e., to suggest promising queries based on the preference function determined by the data analyst.

The rest of the paper is organized as follows. In Section 2 we discuss about the foundations of data mining and the current view to inductive databases. In Section 3 we describe how inductive databases can be seen as an extension of usual databases with ranking abilities. In Section 4 we show how the central data mining tasks can be naturally expressed by the proposed inductive databases framework and discuss about some of the challenges they raise. Section 5 concludes the paper.

2 On Foundations of Data Mining and Inductive Databases

Data mining is currently a very popular research topic. However, relatively small effort has been put on finding a general frameworks for data mining.

Since data mining has immediate commonalities with statistics and machine learning, one might claim that data mining does not exist but it is just another buzzword for getting more funding to apply techniques of statistics and machine learning. However, this is not exactly true since there are also several distinctive features [3]. For example, summarizing a given data set (e.g., by means of data reduction [6] or data compression [7]) even without making any statistical assumptions is an important task in data mining but this task does not fit very well to statistics nor to machine learning. Furthermore, data mining is typically iterative and interactive, an exploratory process of data analysis. This viewpoint is not emphasized in statistics nor in machine learning. In fact, it could even cause some troubles for them.

Although the Grand Unified Theory of Data Mining would be useful to support the existence of data mining as a scientific discipline, there is even greater need for that because of practical reasons: systematizing the field of data mining would benefit practical data mining of being more than a bunch of techniques.

A very promising candidate for the theory of data mining is offered by the inductive databases [4,5]. Inductive databases are databases that contain inductive generalizations about the data, in addition to the usual data [8]. These inductive generalizations can be e.g. clusterings, classifications or patterns. Clearly, the inductive databases can naturally support exploratory data analysis in terms of ad-hoc querying to the inductive database [9]. The high-level notion of inductive databases captures all major data mining tasks, such as pattern discovery, density estimation, clustering, and prediction. However, it is not clear how these tasks should be expressed in the inductive databases framework and what the inductive databases framework should be to express these tasks in a fruitful way.

Most of the existing work on inductive databases has been rooted on pattern discovery and the proposed inductive databases are mostly extensions of usual database query languages [4,10,11,12]. The typical model for inductive databases consists of a data component and a pattern component, and both components can be queried [4]. Restricting inductive databases to pattern discovery can be motivated by the fact that there are already a large number of practical primitives for pattern discovery available and also applicable general frameworks have been sketched [13,14].

However, the theory of data mining should capture all major data mining tasks naturally in order to become commonly accepted and supported. Although pattern discovery can be expressed quite naturally in the current models for inductive databases, expressing some other data mining tasks such as clustering in a natural way is not so easy. In the next section we propose a notion of inductive databases that can be used to naturally describe all central data mining tasks in a uniform way.

3 Inductive Databases as Ranking

A usual database consists of two components: a *data model* and *data* [15]. Also the data model consists of two components: a *data schema* which essentially restricts the collection $\mathcal{D}$ of possible data, and the collection $\mathcal{Q}$ of *data manipulation operations* such as queries and update operations. Thus, for our purposes a usual database can be considered as a triplet $(\mathcal{D}, \mathcal{Q}, D)$ where $\mathcal{D}$ is the collection of possible data, $\mathcal{Q}$ is the collection of possible data manipulation operations and $D \in \mathcal{D}$ is the actual data in the database. The collections $\mathcal{D}$ and $\mathcal{Q}$ are considered to be implicitly represented.

The central question in data mining can be expressed as "I have this data. What should I ask about it?" The results of most data mining tasks can be seen also as queries and their evaluation results. (See Section 4 for examples.) Thus, if the user has a database, the question become even more concrete: "I have this database. Which queries are relevant for this data?" Without any knowledge about the interests of the user there are no meaningful answers. Fortunately, expressing the preferences (in some precision) is usually possible even when choosing the most relevant queries directly from often enormously large collection of all queries is too difficult for the user.

This immediately gives rise to the following notion of inductive databases. An inductive database consists of the components of a usual database and a collection $\mathcal{R}$ of *rankings* of the data manipulation operations. (Most of the cases it would be sufficient to consider only rankings of queries but conceptually there is no need for that restriction.) Thus, an *inductive database* is a 4-tuple $(\mathcal{D}, \mathcal{Q}, \mathcal{R}, D)$.

A ranking is a list of data manipulation operations in $\mathcal{Q}$ where each data manipulation operation occurs at most once. The most important operation for rankings is popping the head of the list, i.e., fetching the next best data manipulation operation w.r.t. the preference function induced by the ranking.

Already this simple operation is sufficient for many data mining tasks. For example, also popping the last operation from the (possibly infinite) list would be many times useful (see Section 4.1 for an example of this in the context of frequent itemset mining) but conceptually there is no need for any new primitives since popping the last element from the list is conceptually equivalent to popping the first element from the reverse of the list, i.e., fetching the next worst operation from the ranking is conceptually equivalent to fetching the next best operations from the inverse of that ranking.

Sometimes, however, there is need for also a few other operations. For example, it would occasionally be very useful to be able to compare the usefulness of the data manipulation operations and also to compute the actual usefulness values. Fortunately, these operations are often defined as a side product of the ranking. The reason why they are not required in the proposed inductive databases framework is that computing them can be much more difficult that ranking: Comparing two arbitrary data manipulation operations can be very difficult even when fetching the next best operation is trivial. Also, meaningful usefulness values do not always even exist. For example, the ranking can be partially done by a human expert that can only express her preferences and not the actual usefulness values. This situation occurs sometimes also due to privacy issues. In practice, the rankings will be augmented with some additional information. Especially the information whether two consecutive operations in the ranking are equally good will often be of interest.

One particularly nice aspect of the proposed inductive databases framework is that the inductive database preserves the closure properties of the underlying usual database since the essential difference between the inductive and the usual databases is the ranking function.

4 Expressing Data Mining Tasks by Rankings

Although the inductive databases framework proposed in the previous section seems to be quite clean and simple, it is not yet clear whether the standard data mining tasks can be naturally expressed by it. In this section we show how the central tasks in data mining can be expressed in a valuable way by the proposed inductive databases framework. We describe how different data mining tasks can be seen as queries and query evaluations. We also highlight some challenges of defining and computing the rankings in data mining.

4.1 Pattern Discovery

Pattern discovery is an important subfield in data mining where the goal is to find all interesting patterns in the given data [16,17]. The interesting patterns can be e.g. local regularities in the data, or parsimonious summaries of subsets of data [18].

The most prominent examples of interesting patterns are *frequent itemsets* and *association rules* [19,20]. Let D be a bag (i.e., a multi-set) consisting of finite subsets (called *transactions*) of a set I of *items*. The bag D called a *transaction database*.

An *itemset* X is a subset of I. The frequency $\mathit{fr}\,(X, D)$ of the itemset $X \subseteq I$ in a transaction database D is the fraction of sets in D containing X, i.e., $\mathit{fr}\,(X, D) = |\{Y \in D : X \subseteq Y\}| \,/\, |D|$ where all collections of sets are interpreted as bags. The frequency $\mathit{fr}\,(X, D)$ can be seen as the empirical probability of X. The interestingness of an itemset is considered to be its frequency in the given transaction database.

An *association rule* $X \Rightarrow Y$ consists of two itemsets $X, Y \subseteq I$. The accuracy of an association rule $X \Rightarrow Y$ in D is the fraction of transactions containing Y that also contain X, i.e., $\mathit{acc}\,(X \Rightarrow Y, D) = \mathit{fr}\,(X \cup Y, D)/\mathit{fr}\,(Y, D)$. From the viewpoint of probabilities the accuracy $\mathit{acc}\,(X \Rightarrow Y, D)$ can be interpreted as an the conditional empirical probability of X given Y.

However, computing all frequent itemsets or association rules is not feasible since the number of them is exponential in the number of items in I. Instead, the practical goal has been finding all itemsets $X \subseteq I$ that are σ-frequent in D, i.e., the collection $\mathcal{F}\,(\sigma, D) = \{X \subseteq I : \mathit{fr}\,(X, D) \geq \sigma\}$, and σ-frequent δ-accurate association rules $X \Rightarrow Y$ with X and Y being σ-frequent in D and having no common items.

In practice, several minimum threshold values σ and δ have to be tried since it is difficult to find a good trade-off between the understandability of the pattern collection and its descriptive power. Instead of the minimum threshold values, one can specify the upper bound for the number of patterns to be produced but still the same problem of finding a suitable parameter value exists. Although the association rules are usually considered as the actual end product, most of the attention has been devoted for finding collection of frequent itemsets [21] since association rules can be produced from them as a simple post-processing step [19].

The proposed inductive databases framework fits perfectly for mining frequent itemsets and association rules: Itemsets and association rules can be seen as queries. The query evaluation result for an itemset $X \subseteq I$ in D corresponds to the bag $\mathit{cover}\,(X, D) = \{Y \in D : X \subseteq Y\}$ and the query evaluation result for an association rule $X \Rightarrow Y$ corresponds to two bags $\mathit{cover}\,(Y, D)$ and $\mathit{cover}\,(X \cup Y, D)$. Conceptually the association rule query can be seen to consist of first selecting the transactions of $\mathit{cover}\,(Y, D)$ from D and second $\mathit{cover}\,(X \cup Y, D)$ from $\mathit{cover}\,(Y, D)$ since $\mathit{cover}\,(X \cup Y, D) \subseteq \mathit{cover}\,(Y, D)$.

The ranking of itemsets based on their frequencies can be claimed to solve the frequent itemset mining task from the user point of view even better than finding all σ-frequent sets since the inductive database naturally supports the interactive examination of the itemsets. It also gives a slightly different viewpoint to frequent itemset mining by interpreting the frequent itemset mining as generating frequent itemsets on demand in the order of decreasing frequencies instead of computing the collection for different threshold values.

Instead of the itemsets with high frequencies (i.e., regularities in the data), one can be interested also in the very rare itemsets, i.e., the surprising itemsets. As mentioned in the previous section, conceptually this is only inverting the ranking. Also in practice it is feasible to look for infrequent itemsets in addition to the frequent ones [22].

Clearly, mining also other kinds of interesting patterns fits immediately to this framework: the patterns are listed in the order of decreasing interestingness. Thus, it can be said that ranking suits quite well for pattern discovery.

4.2 Density Estimation

In addition to local modeling, also global modeling is important in data mining. The central global modeling tool is the estimation of probability distributions.

Based on Section 4.1 it is clear that we could present probability distributions of a given class in decreasing order of their likelihood w.r.t. the data. The ranking approach bends naturally also for other kinds of density estimation tasks. In this section we shall show how it can be used to construct refining representation of a probability distribution as a mixture (i.e., a convex combination) of simpler distributions. This kind of hierarchical modeling occurs frequently in data summarization since defining the right trade-off between understandability and accuracy of the model in advance is usually very difficult.

Let us consider the situation where the database D consists of points in $\mathbb{R}^d$ and we try to model the data by a mixture of d-dimensional Gaussians. Let

$$\mathbb{P}\left(p \mid G_1, \ldots, G_m, \alpha_1, \ldots, \alpha_m\right) = \sum_{i=1}^{m} \alpha_i \mathbb{P}\left(p \mid G_i\right)$$

be the likelihood of $p \in D$ given the Gaussians $G_1, \ldots, G_m$ with weights $\alpha_1, \ldots, \alpha_m$ such that $\sum_{i=1}^{m} \alpha_i = 1$ and $\alpha_i \geq 0$ for all $i = 1, \ldots, m$. The joint likelihood of all points in D is similarly

$$\mathbb{P}\left(D \mid G_1, \ldots, G_m, \alpha_1, \ldots, \alpha_m\right) = \prod_{p \in D} \mathbb{P}\left(p \mid G_1, \ldots, G_m, \alpha_1, \ldots, \alpha_m\right).$$

Thus, the queries are mixtures of Gaussians and the query evaluation in D results the associated likelihoods for all points in D.

The refining mixture of Gaussians for the data can be computed as follows. The first Gaussian G_1 is the one with the maximum likelihood w.r.t. the data D. The second Gaussian G_2 is the one with maximum likelihood w.r.t. the data points $p \in D$ weighted by $1 - \mathbb{P}\left(p \mid G_1, 1\right)$ and the weights α_1 and α_2 are chosen in such a way that they maximize the likelihood $\mathbb{P}\left(D \mid G_1, G_2, \alpha_1, \alpha_2\right)$.

Similarly the kth Gaussian G_k is the maximum likelihood estimate w.r.t. the data D weighted by $1 - \mathbb{P}(p \mid G_1, \ldots, G_{k-1}, \alpha_1, \ldots, \alpha_{k-1})$ and α_k is chosen and $\alpha_1, \ldots, \alpha_{k-1}$ are scaled in such way that the updated values $\alpha_1, \ldots, \alpha_k$ maximize the likelihood $\mathbb{P}(D \mid G_1, \ldots, G_k, \alpha_1, \ldots, \alpha_k)$.

This way a refining description of the data can be represented by ranking. The previous queries (i.e., the previous mixture of $k-1$ Gaussians) are exploited when computing the next best query (i.e., the mixture of k Gaussians). A similar rankings occur also when ordering patterns w.r.t. their informativeness [23].

4.3 Clustering

Maybe the most important task in data mining is clustering, i.e., grouping data into groups consisting of similar data. For clusterings, the proposed inductive databases framework seems to be especially suitable: Defining clusterings unambiguously is not usually very easy and thus several alternative suggestions, possibly with associated quality values for the clusterings, are most welcome.

Let us consider k-center clustering of points in $D \subset \mathbb{R}^d$ and let the goodness of the clustering be measured by Euclidean distance. Then the queries correspond to k points in $\mathbb{R}^d$ and the query evaluations result partitions of D into k groups. (Note that also hierarchical clustering can be described by ranking, similarly to the refining mixtures in Section 4.2.)

An important aspect in ranking clusterings is what we would really like to see is the listing of clusterings in such order that each prefix of the list contains a representative collection of clusterings. For example, in the case of k-center clustering w.r.t. Euclidean distance, the straightforward application of the distance function would typically produce quite uninformative ranking: all good clusterings would be essentially the same. Thus, after finding the best clustering, we should be able to rule out the clusterings that are too similar to it. In practice, this goal can be reached for by stochastic search techniques.

Although listing the clusterings in a very good order is not trivial, it is clear that the rankings offer a natural way to describe the clustering tasks.

4.4 Prediction

Another data mining task with high practical relevancy is prediction of outcomes of some unknown function for a given data based on examples of outcomes for some other data points. A well-known special case of the prediction task is classification, i.e., labeling unlabeled data based on another set of labeled data (called the *training data*). The query in this case is the classifier and its evaluation on D results the classification of the data points in D which is typically a partition of D.

There are several established methods for the classification task, see e.g. [24]. Furthermore, the representations of classifiers can differ very much. This is not desirable for the inductive databases since one of the goals of the inductive databases is to have only a small number of primitives succinct for data mining.

A conceptually simple solution to this problem of different representations would be to explicitly compute labelings. However, this does not make sense in general, as the goal of classification is to *generalize* the knowledge extracted from the training data to possibly infinite collections of data [25]. Also, finding a common, understandable representation for different classifiers can be very difficult. For example, representing a hyperplane with a decision tree with splits of one variable at a time is usually very awkward (and vice versa).

Based on the above observations it might seem that inductive databases can hardly be useful in prediction tasks since there is need for possibility to represent many kinds of predictors. However, the goal of prediction does not depend on the method. For example, in the case of classification, the goal is to find classifiers that are accurate also for still unclassified (and possibly unseen) data. Thus, although the actual predictors can differ much from each other, the same ranking functions can be used for all predictors solving the same prediction task at hand. Of course, sometimes the ranking function has to be slightly modified based on e.g. the preferences of the user w.r.t. different classifiers. For example, one could prefer Bayesian methods or decision trees with small depth regardless of their estimated prediction performance.

Thus, prediction tasks can be comfortably expressed in the proposed inductive databases framework and the inductive databases framework can support also the construction of the actual inductive database since same ranking functions can be used for all different methods solving the same prediction task.

5 Conclusions

In this paper we have described how inductive databases can be seen as ranking data manipulation operations of usual databases. We have proposed an inductive databases framework based on ranking that gives clean and simple definition what the inductive databases essentially could be. Furthermore, we have shown that the framework is also very suitable for expressing the central data mining tasks naturally and it can give even some new insights to them. We believe that the concept of ranking is relevant for data mining in general due to the exploratory nature of data mining.

However, this paper is still far from concrete implementations of general purpose inductive databases. There are many important open questions relevant for the suggested concept of inductive databases ranging from technical database theory questions to understandability and relevancy questions:

- How to summarize the ranked queries? There is often a need for summarizing even the rankings. For example, recently pattern discovery research has been focused on summarizing the collection of interesting patterns under the flag of condensed representations, see e.g. [26,27] and the references therein.
- What kind of ranking languages are useful and usable? It is clear that rankings should be described by languages that are suitable for that task. Furthermore, it would be desirable if the expressive power and the computational complexity of the language used for expressing the rankings are low.

– What kind of trade-offs there are between generality, efficiency and usefulness of the ranking languages?
– Are there non-trivial trade-offs and relations between expressive powers of data definition languages, data manipulation languages and ranking languages?
– Is there a need for higher order queries? Higher order queries, e.g. queries of queries, can be expressed by the rankings. However, there might be situations where the real higher order queries would be truly beneficial.

Acknowledgments. I wish to thank Floris Geerts, Bart Goethals, Matti Kääriäinen and Heikki Mannila for exciting discussions on foundations of data mining and the nature of inductive databases.

References

1. Hand, D.J., Mannila, H., Smyth, P.: Principles of Data Mining. MIT Press (2001)
2. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Academic Press (2001)
3. Mannila, H.: Theoretical frameworks for data mining. SIGKDD Explorations **1** (2000) 30–32
4. De Raedt, L.: A perspective on inductive databases. SIGKDD Explorations **4** (2003) 69–77
5. Imielinski, T., Mannila, H.: A database perspective on knowledge discovery. Communications of The ACM **39** (1996) 58–64
6. Barbará, D., DuMouchel, W., Faloutsos, C., Haas, P.J., Hellerstein, J.M., Ioannidis, Y.E., Jagadish, H.V., Johnson, T., Ng, R.T., Poosala, V., Ross, K.A., Sevcik, K.C.: The new jersey data reduction report. IEEE Data Engineering Bulletin **20** (1997) 3–45
7. Li, M., Vitányi, P.: An Introduction to Kolmogorov Complexity and Its Applications. 3rd edn. Texts in Computer Science. Springer-Verlag (1997)
8. Mannila, H.: Inductive databases and condensed representations for data mining. In Maluszynski, J., ed.: Logic Programming, Proceedngs of the 1997 International Symposium, Port Jefferson, Long Island, N.Y., October 13-16, 1997. MIT Press (1997) 21–30
9. Boulicaut, J.F., Klemettinen, M., Mannila, H.: Modeling KDD processes within the inductive database framework. In Mohania, M.K., Tjoa, A.M., eds.: Data Warehousing and Knowledge Discovery, First International Conference, DaWaK '99, Florence, Italy, August 30 - September 1, 1999, Proceedings. Volume 1676 of Lecture Notes in Artificial Intelligence. Springer (1999) 293–302
10. Giannotti, F., Manco, G.: Querying inductive databases via logic-based used-defined aggregates. In Zytkow, J.M., Rauch, J., eds.: Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD '99, Prague, Czech Republic, September 15-18, 1999, Proceedings. Volume 1704 of Lecture Notes in Artificial Intelligence. Springer (1999) 125–135
11. Imieliński, T., Virmani, A.: MSQL: A query language for database mining. Data Mining and Knowledge Discovery **3** (1999) 373–408
12. Jeudy, B., Boulicaut, J.F.: Constraint-based discovery and inductive queries: Application to association rule mining. [28] 110–124

13. De Raedt, L., Jaeger, M., Lee, S.D., Mannila, H.: A theory of inductive query answering. In Kumar, V., Tsumoto, S., eds.: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), 9-12 December 2002, Maebashi City, Japan. IEEE Computer Society (2002) 123–130
14. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery **1** (1997) 241–258
15. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
16. Hand, D.J.: Pattern detection and discovery. [28] 1–12
17. Mannila, H.: Local and global methods in data mining: Basic techniques and open problems. In Widmayer, P., Ruiz, F.T., Bueno, R.M., Hennessy, M., Eidenbenz, S., Conejo, R., eds.: Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings. Volume 2380 of Lecture Notes in Computer Science. Springer (2002) 57–68
18. Fayyad, U., Uthurusamy, R.: Evolving data mining into solutions for insights. Communications of the ACM **45** (2002) 28–31
19. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In Buneman, P., Jajodia, S., eds.: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993. ACM Press (1993) 207–216
20. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., eds.: Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press (1996) 307–328
21. Goethals, B., Zaki, M.J., eds.: Proceedings of the Workshop on Frequent Itemset Mining Implementations (FIMI-03), Melbourne Florida, USA, November 19, 2003. Volume 90 of CEUR Workshop Proceedings. (2003) `http://CEUR-WS.org/Vol-90/`.
22. Mielikäinen, T.: Intersecting data to closed sets with constraints. [21]
23. Mielikäinen, T., Mannila, H.: The pattern ordering problem. [29] 327–338
24. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics. Springer-Verlag (2001)
25. Vapnik, V.N.: The Nature of Statistic Learning Theory. Statistics for Engineering and Information Science. Springer-Verlag (2000)
26. Calders, T., Goethals, B.: Minimal k-free representations of frequent sets. [29] 71–82
27. Mielikäinen, T.: Separating structure from interestingness. In Dai, H., Srikant, R., Zhang, C., eds.: Advances in Knowledge Discovery and Data Mining, 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004, Proceedings. Volume 3056 of Lecture Notes in Artificial Intelligence. Springer (2004) 476–485
28. Hand, D.J., Adams, N.M., Bolton, R.J., eds.: Pattern Detection and Discovery, ESF Exploratory Workshop, London, UK, September 16-19, 2002, Proceedings. Volume 2447 of Lecture Notes in Computer Science. Springer (2002)
29. Lavrac, N., Gamberger, D., Blockeel, H., Todorovski, L., eds.: Knowledge Discovery in Databases: PKDD 2003, 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003, Proceedings. Volume 2838 of Lecture Notes in Artificial Intelligence. Springer (2003)