Technische Universität Darmstadt
Knowledge Engineering Group
Hochschulstrasse 10, D-64289 Darmstadt, Germany

http://www.ke.informatik.tu-darmstadt.de

Technical Report TUD–KE–2007–06

*Bruno Crémilleux, Johannes Fürnkranz,
Arno Knobbe, Martin Scholz*

From Local Patterns to Global Models: The LeGo Approach to Data Mining

From Local Patterns to Global Models: The LeGo Approach to Data Mining

Bruno Crémilleux BRUNO.CREMILLEUX@INFO.UNICAEN.FR
GREYC - Université de Caen, Département informatique, F-14032 Caen Cedex, France
Johannes Fürnkranz JUFFI@KE.INFORMATIK.TU-DARMSTADT.DE
TU Darmstadt, Knowledge Engineering Group, Hochschulstrasse 10, D-64289 Darmstadt, Germany
Arno Knobbe A.KNOBBE@KIMINKII.COM
Utrecht University, P.O. box 80 089,NL-3508 TB Utrecht, the Netherlands
Kiminkii, P.O. box 171, NL-3990 DD, Houten, the Netherlands
Martin Scholz SCHOLZ@HP.COM
Hewlett-Packard Labs, 1501 Page Mill Road, Palo Alto, CA 94304, USA

Abstract

In this paper we present LeGo, a generic framework that utilizes existing local pattern mining techniques for global modeling in a variety of diverse data mining tasks. In the spirit of well known KDD process models, our work identifies different phases within the data mining step, each of which is formulated in terms of different formal constraints. It starts with a phase of mining patterns that are individually promising. Later phases establish the context given by the global data mining task by selecting groups of diverse and highly informative patterns, which are finally combined to one or more global models that address the overall data mining task(s). The paper discusses the connection to various learning techniques, and illustrates that our framework is broad enough to cover and leverage frequent pattern mining, subgroup discovery, pattern teams, multi-view learning, and several other popular algorithms. The Safarii learning toolbox serves as a proof-of-concept of its high potential for practical data mining applications. Finally, we point out several challenging open research questions that naturally emerge in a constraint-based local-to-global pattern mining, selection, and combination framework.

1. Introduction

Over the last decade, local pattern discovery has become a rapidly growing field (Morik et al., 2005), and a range of techniques is available for producing extensive collections of patterns. Because of the exhaustive nature of most such techniques, the pattern collections provide a fairly complete picture of the information content of the database. However, in many cases this is where the process stops. The so-called local patterns represent fragmented knowledge, and often it is not clear how the pieces of the puzzle can be combined into a global model. Because a useful global model, such as a classifier or regression model, is often the expected result of a Data Mining process, the question of how to turn large collections of patterns into global models deserves attention. In this paper, we provide an overview of what it takes to build global models from local patterns. In our view, a common ground of all the local pattern mining techniques is that they can be considered to be feature construction techniques that follow different objectives (or constraints). We will see that the redundancy of these patterns and the selection of suitable subsets of patterns are addressed in separate steps, so that each resulting feature is highly informative in the context of the global data mining problem.

We define a framework, called *from Local Patterns to Global Models* (LeGo), consisting of a number of steps. Each step can be implemented by a range of techniques from the literature, making the framework general and of value to practitioners wishing to apply their favorite algorithm in a wider context. Furthermore, it subsumes a number of existing global methods based on pattern discovery. The framework helps analyzing and improving such methods by relating it to other similar methods, and suggesting alternative options for individual steps in the process. A general framework

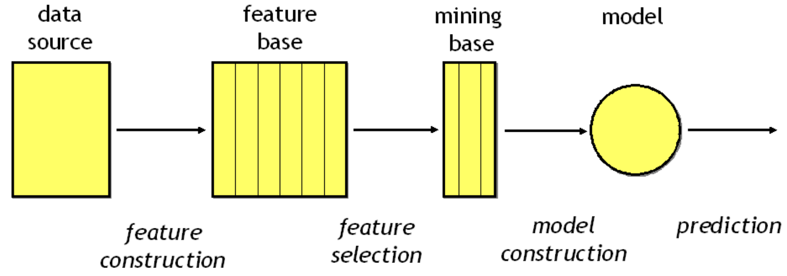


Figure 1: Conventional Data Mining Process Model

is important to understand the common grounds of different methods, to discover potential synergies, and to identify important fields of research.

This paper is organized as follows. Section 2 gives an overview of the LeGo framework, and Section 3 describes its different steps and places them in context of the current literature. Its motivation is discussed in Section 4. The Safarii learning toolbox, a running method of the LeGo framework, is detailed in Section 5. In Section 6, we discuss current works and research issues related to our framework before we conclude in Section 7.

2. The LeGo framework

We present our framework by relating it to the conventional KDD process model. Figure 1 recalls the process model, as it has been sketched in similar ways in numerous publications, going back to (Fayyad et al., 1996). As we will focus on the sequence of phases during the mining process, we have omitted the evaluation phase and the subsequent feedback loops that can go to each of the individual phases.

The process starts with a *data source* (typically a relational database) that needs to be prepared for the mining process. The first phase, known as *feature construction*, produces from the initial data source a so-called *feature-base*, by means of some, typically manual or semi-automatic, transformation process. The purpose of this transformation may be to extract specific, potentially useful information that is only represented implicitly in the data source (e.g. translating purchase-date into a weekend/weekday indicator). Alternatively, the feature construction step may be performed in order to translate the original data format into a format the learning algorithm of choice requires, such as strictly numeric or binary. Such features can be attribute-value pairs (as in classification rule learning), items (as in association rule discovery), word occurrences (as in text mining), or similar. Once the data source is transformed into a feature base, the *feature selection* phase (Guyon & Elisseeff, 2003) is responsible for selecting a subset of these features (the *mining base*). This is particularly important when large numbers of features are generated. Typical problems with large feature spaces include text mining (Forman, 2003), propositionalization approaches to relational learning (Kramer et al., 2001), and others. Finally, a *model construction* phase involves applying one of the many available inductive methods to produce a model from the mining base. In descriptive data mining, the model itself is of primary interest. In predictive data mining, the model is used for making predictions, basically treating it as a black box.

We now view the LeGo framework as an instance of this general process model (see Figure 2), with local patterns, rather than features, being the prime subject. We informally define *local patterns* as regularities that hold for a particular part of the data. The term local refers to the fact that it captures some aspect of the data, without providing a complete picture of the database (see Section 3.1). Local patterns do not necessarily represent exceptions in the data (Hand, 2002), but rather fragmented and incomplete knowledge, which may be fairly general. We identify the following phases:

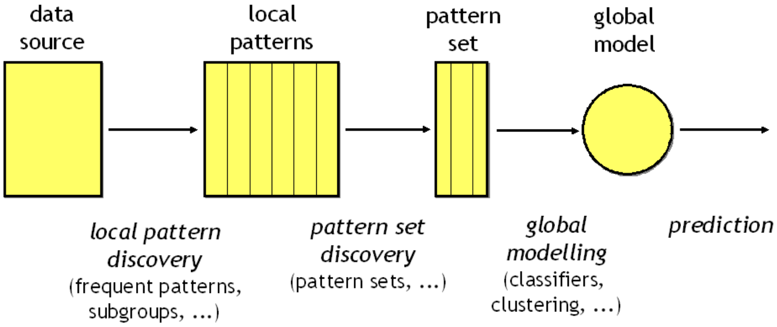


Figure 2: The LeGo framework

Local Pattern Discovery: This phase is responsible for producing a set of candidate patterns by means of an exploratory analysis of a search-space of patterns, defined by a set of inductive constraints provided by the user. As such, this phase can be seen as an automated instance of the feature construction phase in the KDD process (see Figure 2). Patterns are typically judged on qualities such as their frequency or predictive power with respect to some target concept.

Pattern Set Discovery: This phase considers the potentially large collection of patterns produced in the preceding phase, and selects from those a compact set of informative and relevant patterns that shows little redundancy. This phase is the counterpart of the feature selection phase in the KDD process (Figure 2).

Global Modeling: This phase is responsible for turning the condensed set of relevant patterns into a well-balanced global model. The Global Modeling phase either treats each local pattern as a constructed feature, and applies an existing inductive method, or applies some pattern combination strategy that is specific to the class of patterns discovered.

The prototypical instantiation of this framework is classification by association, as exemplified by the CBA rule learning algorithm (Liu et al., 1998; Liu et al., 2000). This type of algorithm typically uses a conventional association rule discovery algorithm, such as Apriori to discover a large number of patterns. From these, all patterns that have the target class in the head are selected, and only those are subsequently used for inducing a global theory. The global theory is typically a disjunction of patterns, found by a simple set-covering algorithm: patterns are sorted according to some heuristic function and the best one is repeatedly added to the disjunction. Variations in the global model may use decision lists or redundant rule sets. A variety of successor systems have been proposed that follow the same principal architecture (e.g., (Bayardo Jr., 1997; Jovanoski & Lavrač, 2001; Li et al., 2001; Yin & Han, 2003; Mutter et al., 2004)).

Note that the separation of the phases does not have to be as clear as it is in these algorithms. It is also useful to view conventional rule learning algorithms, such as those of the covering family (also known as Separate & Conquer), within this framework (Fürnkranz, 2005). In these algorithms, the Local Pattern Discovery phase focuses on finding a single best global pattern, i.e., the next rule to add to the growing theory. The examples covered by this rule are removed, and the process repeats until all examples have been covered. The purpose of this covering loop is to find a good pattern set that collectively covers all training examples. Finally, the found patterns are turned into a classifier by forming a disjunction, a decision list, or an ensemble. A similar line of research aims to discover subgroup patterns relative to prior knowledge and previously found patterns. In this case, the "covered" examples are not completely removed but their weight is adjusted appropriately by the learning scheme. Patterns correspond to soft predictions that can be averaged in predictive settings.

Thus, phases of Local Pattern Discovery, Pattern Set Discovery, and Global Modeling are tightly interleaved in these families of algorithms, which makes it harder to recognize these algorithms as instantiations of our framework. On the other hand, some parts of the framework, like the dependency of the quality criteria that are used in the Local Pattern Discovery phase on the Global Modeling task (cf. Section 6.1), become much clearer in the light of this framework.

3. LeGo Phases

In this section, we give a more detailed description of the main phases in our framework and put them into the context of the state-of-the-art in data mining.

3.1 Local Pattern Discovery

The most basic, and at the same time most popular, type of local pattern discovery is the (unsupervised) discovery of *frequent itemsets* (Goethals, 2005). Clearly, a frequent itemset is an incomplete representation of some aspect of the distribution of items, and of possible co-occurrences among the items (associations). An itemset is local because it covers only the part of the database for which its items are supported. A frequent itemset discovery algorithm typically performs an exhaustive, top-down, level-wise search for the frequent sets. In most cases, some condensed representation of the set of itemsets is returned, rather than the complete set. The discovery of frequent patterns has been generalized into more elaborate structures such as sequences, trees and graphs, for example with the aim of discovering frequent fragments in molecules (Zaki, 2001; Chi et al., 2005; Sevon et al., 2006; Nijssen & Kok, 2004). Another extension of this approach is to deal with various inductive constraints (Ng et al., 1998; De Raedt et al., 2002; Soulet & Crémilleux, 2005) and not only with the frequency constraint.

Another important example of local pattern discovery is known as *Subgroup Discovery* (sometimes referred to as *Correlated Pattern Discovery*). The main goal is to identify patterns that are *interesting*, in the sense that they are well supported and that the set of covered examples differs substantially from the overall population with respect to the distribution of its boolean (or nominal) target attribute. The result of Subgroup Discovery is a set of subsets of the data, usually characterized in the form of classification rules. The construction of a global model for predictive purposes is not the main focus of the techniques; in fact, the task can be rephrased as mining sets of local patterns in supervised settings, with the objective function—typically a rule interestingness measure—being a parameter of the task itself. A large variety of measures suitable for this task have been investigated (Klösgen, 1996; Tan et al., 2002), many of which are well-known heuristics for inductive rule learning (Fürnkranz & Flach, 2005). Important examples include the binomial test function, χ^2, or the Novelty function, which later has often been referred to as weighted relative accuracy (Lavrac et al., 1999).

These two prototypical discovery methods demonstrate an important concept: *local patterns can be interpreted as features*, in this case binary features. The set of conditions represented by the pattern (subgroup, frequent itemset, ...) either does or does not hold for a given example. Thus, any Data Mining operation that works on binary features can be employed in the subsequent phases of Pattern Set Discovery and Global Modeling. In many cases, these subsequent operations will simply ignore any information concerning the structure of the patterns, and will focus only on the resulting feature. This interpretation emphasizes the locality of patterns: each pattern helps to identify some important subset of the database that exhibits some properties that distinguish it from its complement.

3.2 Pattern Set Discovery

In the Local Pattern Discovery phase, patterns are discovered on the basis of their individual merits. In practice, this results in large sets of local patterns, with potentially high levels of redundancy among the patterns. For manual inspection of the collection of patterns, reporting more than a handful of patterns is clearly infeasible. Furthermore, when inducing global models from the set of local patterns, machine learning procedures tend to be hindered by the presence of many, often redundant, features. The goal of the Pattern Set Discovery phase therefore, is to reduce the redundancy by selecting a subset of patterns from the initial large set on the basis of their usefulness in the context of other patterns selected.

Several approaches have been proposed to reduce the number of local patterns irrespective of their subsequent use. Examples include condensed representations (Calders et al., 2005), compression of the dataset by exploiting the Minimum Description Length Principle (Siebes et al., 2006) or the constraint-based paradigm (Ng et al., 1998; De Raedt et al., 2002). Constraints provide a focus that allows to reduce the number of extracted patterns to those of a potential interest given by the user. This paradigm may be strengthened by the exploitation of (partial) domain knowledge to support knowledge discovery (Soulet et al., 2007). Unfortunately, even if these approaches enable us to reduce the number of produced patterns, the output still remains too large for an individual and global analysis performed by the end-user. The most significant patterns are lost among too much trivial, noisy and redundant information.

Recently, two approaches to Pattern Set Discovery have appeared in the literature, which explicitly represent the goal of combining and selecting patterns: constraint-based pattern set mining (De Raedt & Zimmermann, 2007), and pattern teams (Knobbe & Ho, 2006b; Knobbe & Ho, 2006a). In broad terms, these approaches are very similar. Both assume that the syntactic structure of the individual patterns is irrelevant at this stage, and that patterns can be fully characterized by a binary feature that determines for each example whether it is covered by the pattern or not. As the name suggests, *constraint-based pattern set mining* is based on the notion of constraints defined on the level of pattern sets (rather than individual patterns). These constraints can capture qualities of the set such as size or representativeness (a measure for the predictiveness of the collective). More interestingly, De Raedt and Zimmermann (2007) propose constraints on the similarity between pairs of patterns in the set, such as a minimum symmetric difference, or a maximum redundancy (defined as the amount of overlap between patterns). A defining characteristic of this approach is that *all* pattern sets that satisfy the constraints are reported. This to some degree contradicts the goal of reducing the amount of information reported to an end-user, as the amount of reported pattern sets may exceed the number of patterns discovered initially, given too lenient constraints.

In the *pattern team* approach on the other hand, only a single optimal subset of patterns is returned. Pattern sets are implicitly ranked on the basis of a quality measure, and the best-performing set (the pattern team) is reported. Typically, the quality measure promotes the utility (e.g. informativeness or predictiveness) of the set as a collective, while at the same time reducing the amount of redundancy among elements of the team. Often, selective pressure among patterns is enforced by requiring the pattern team to have a fixed size k (typically a number well below 10). Knobbe and Ho (2006b) suggest a number of quality measures, both supervised and unsupervised, that promote different qualities of pattern sets. Joint entropy (unsupervised), for example, captures the information content of patterns involved, thus promoting independence of patterns. Supervised measures such as DTM accuracy, on the other hand, select subsets that lead to accurate classifiers, for a given target concept. The reduction of redundancy in this case is achieved implicitly by limiting the size of the team. Section 5 comes back to this approach. A new measure was recently introduced in (Rückert & Kramer, 2007), which tries to achieve class-correlation and feature diversity simultaneously.

3.3 Global Modeling

Computing a global model from a set of local patterns (i.e. features) can be quite straightforward; we may basically utilize any machine learning algorithm at this point, most of which will clearly benefit from high quality features. It is well known that good features often contribute more to data mining success than the selection of a specific algorithm and the fine-tuning of its parameters. Although in LeGo, we clearly advocate this generic use of learning techniques for global modeling, up to recently many approaches employed fairly ad hoc techniques, or used methods that depend on the specific nature of the local patterns (e.g. itemsets or clusters).

Especially in the discipline of (association) rule discovery, the problem of building a unified global classification model has been approached by so-called *combination strategies*, an idea that goes back to the concept of ensembles (or Multiple Classifier Systems). If we consider each local pattern as a weak classifier, we can (arguably) construct our global model as an ensemble of patterns. We now have a range of proposed combination strategies at our disposal that effectively assign a weight to each rule or pattern (Zimmermann & De Raedt, 2004; Mutter et al., 2004). Obvious candidates are Majority Voting, which essentially assigns equal weights to each rule, and Linear Weight Voting, which gives precedence to rules that rank higher with regards to the rule evaluation measure selected (e.g. χ^2, weighted relative accuracy). Although these strategies are relevant and have been popular, they do not take into account the level of locality of patterns and possible correlations among patterns. More sophisticated combination strategies do consider these issues. An obvious example is the well-known covering approach, such as used by CBA and successors (Liu et al., 1998; Liu et al., 2000; Bayardo Jr., 1997; Jovanoski & Lavrač, 2001; Li et al., 2001; Yin & Han, 2003; Mutter et al., 2004). Further inspiration from rule induction methods is exploited by the *Double Induction* approach (Lindgren & Boström, 2003), which resolves possible conflicts between rules by inducing further rules to decide between these conflicts. This evolution of strategies naturally leads to the final approach of applying arbitrary learning methods to combine sets of patterns, assuming again that every pattern can be interpreted as a (binary) feature. The use of generic induction methods is advocated for example in the recent Correlated Pattern Mining approach (Bringmann et al., 2006) and the Safarii system (Knobbe, 2006) (see next section). Because of the nature of the learning task (binary data, high-dimensional), Support Vector Machines form an obvious and popular candidate.

In a clustering setting, several works aim at designing clustering methods based on associations and frequent patterns (Wang et al., 1999). Ecclat (Durand & Crémilleux, 2002) is based on frequent closed patterns and has the originality to enable a slight overlap between clusters. The potential clusters are the frequent closed patterns because a closed pattern gathers a maximal set of attributes shared by a set of objects, and thus allows to capture the maximum amount of similarity. Then Ecclat evaluates and selects the most interesting clusters by using an interestingness measure that forms a trade-off between two criteria, the *homogeneity* (to favor clusters having many attributes shared by many objects) and the *concentration* (to limit an excessive overlapping of objects between clusters). Co-classification is a way of conceptual clustering that provides a limited collection of bi-clusters. These bi-clusters are linked for both objects and attribute-value pairs. Pensa et al. (2005) proposes a framework for co-classification. A limitation of this framework is that a distance between the bi-sets which are at the origin of the bi-clusters has to be chosen

4. Advantages of LeGo

As building global models is clearly one of our goals, one might wonder why such a global model cannot be induced directly, as is customary in traditional inductive methods. Why spend the extra time to search for an exhaustive collection of patterns, if most of them are later discarded due to redundancy or irrelevancy?

A key motivation comes from the expected accuracy gains resulting from the more exploratory or exhaustive nature of the initial pattern discovery step. Many successful machine learning techniques implicitly include an automated pattern discovery phase. For example, the nodes in the hidden layer of a multi-layer perceptron will typically converge to several useful subconcepts, which may be important for some (but not necessarily all) of the output nodes. Similarly, kernel methods perform an implicit feature generation step.

On the other hand, there are also widely used pre-processing techniques like principal components analysis that perform feature generation with the goal of supporting a subsequent modeling step. Further examples include text mining techniques like probabilistic latent semantic analysis (Hofmann, 1999) and latent Dirichlet allocation (Blei et al., 2003), which introduce a number of latent topics that serve as an intermediate semantic layer capturing important regularities in the input space. Each of these latent topics represents a distribution over words, and each document may be associated to several of these topics. For all these methods, the intermediate layer allows to establish a representation of each input sample in terms of more meaningful or discriminative local patterns.[1]

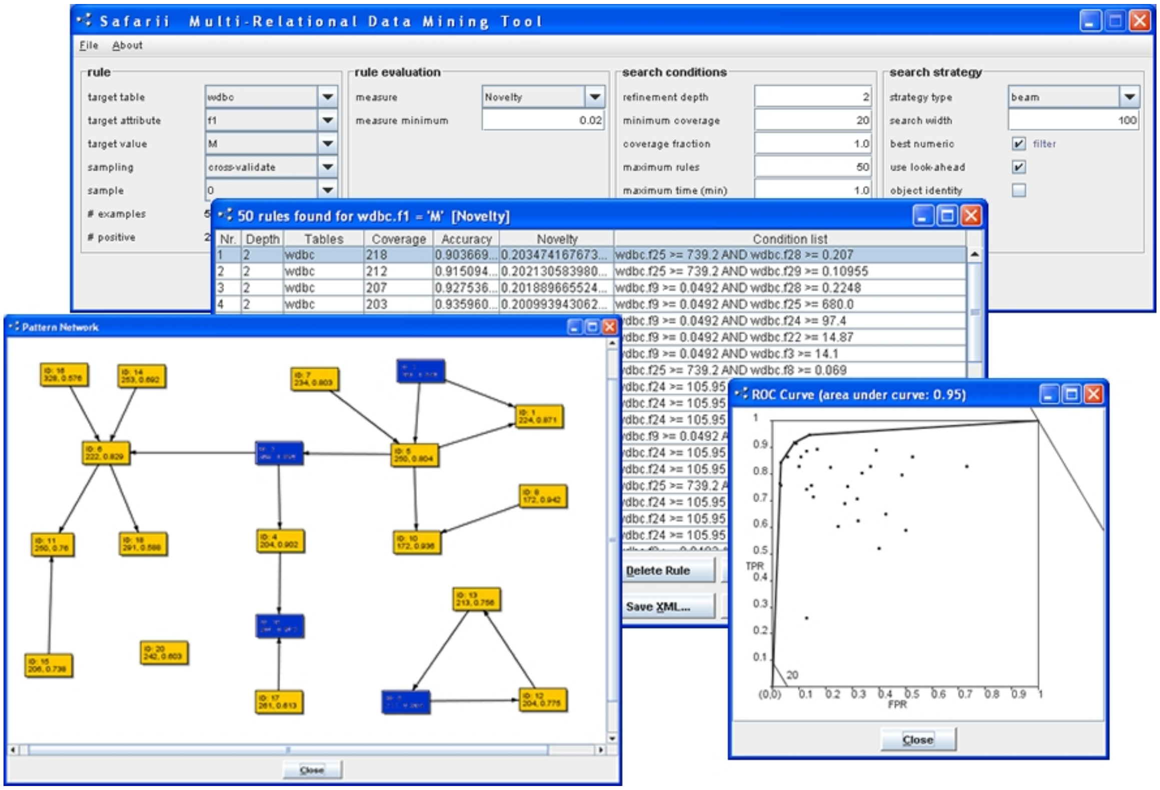
Thus, it seems to be a good idea to adopt local pattern discovery techniques as a pre-processing step for global modeling. In practice, globally useful features can usually be assumed to also perform locally well to a certain degree, which means that they can be detected by local pattern mining techniques that report a larger number of patterns. Unlike most direct learning procedures, which tend to be rather greedy, the set of discovered patterns is typically complete (within the inductive constraints). This means that in subsequent phases, two patterns of moderate quality could be combined to form a perfect model (think of the XOR-problem), whereas the greedy process might result in a suboptimal solution due to an initial greedy choice.

Another key advantage of this approach is that the local pattern discovery can, to some extent, be performed independently of subsequent global modeling steps. The found patterns can be stored as an intermediate result that could be put to use for a variety of different global modeling tasks. Depending on the concrete implementation of the LeGo framework, local patterns or entire pattern sets could be stored. One should, however, keep in mind that in some cases it could be desirable to tailor the patterns or pattern sets to the concrete global modeling task. It is an open research problem, how constraints from the global task can be propagated back to the local pattern discovery phase (cf. Section 6.1).

Storing local patterns as intermediate results is reminescent of work in inductive databases (Imielinski & Mannila, 1996; De Raedt, 2002). Their main idea is to consider the KDD process as a querying process, i.e., as sequences of queries that operate on the data as well as on the patterns and models which hold in the data. Queries which have to return patterns or models are called inductive queries. They may be considered as declarative specifications of constraints on the desired patterns or models. Similarly, the LeGo framework interprets global models as constraints on local patterns, which may serve as a filter in the pattern set discovery phase.

Finally, although we are promoting the construction of global models from local patterns, the global models themselves may not necessarily be the desired end-product. The local patterns could still be the prime subject, and global modeling could serve as a means of validating candidate pattern sets. One could argue that a pattern team that optimizes a certain classifier represents a set of patterns that are worth inspecting manually, as relevancy and lack of redundancy are mostly guaranteed. Additionally, going back from the Pattern Set Discovery phase to the Local Pattern Discovery phase, it makes sense to see what patterns were accepted into a pattern team, and how they relate to the remaining patterns. Potentially, pattern team members may be replaced by

[1] The learning approaches sketched above have in common that the intermediate layer consists of features in a continuous domain, e.g., probabilities that documents belong to specific latent topics. We reach a similar representation when interpreting e.g., rules that represent local patterns in a probabilistic framework, so that they yield continuous predictions. Continuous features that define different degrees of memberships to local patterns, as well as mutually exclusive local patterns (nominal features) emerge as natural generalizations of boolean membership features. Our framework allows for a seamless integration of all of these settings and representations.



rule

target table: wdbc
target attribute: f1
target value: M
sampling: cross-validate
sample: 0
# examples: 5
# positive: 2

rule evaluation

measure: Novelty
measure minimum: 0.02

search conditions

refinement depth: 2
minimum coverage: 20
coverage fraction: 1.0
maximum rules: 50
maximum time (min): 1.0

search strategy

strategy type: beam
search width: 100
best numeric ☑ filter
use look-ahead ☑
object identity ☐

50 rules found for wdbc.f1 = 'M' [Novelty]

Nr.	Depth	Tables	Coverage	Accuracy	Novelty	Condition list
1	2	wdbc	218	0.903669...	0.203474167673...	wdbc.f25 >= 739.2 AND wdbc.f28 >= 0.207
2	2	wdbc	212	0.915094...	0.202130583980...	wdbc.f25 >= 739.2 AND wdbc.f29 >= 0.10955
3	2	wdbc	207	0.927536...	0.201889665524...	wdbc.f9 >= 0.0492 AND wdbc.f28 >= 0.2248
4	2	wdbc	203	0.935960...	0.200993943062...	wdbc.f9 >= 0.0492 AND wdbc.f25 >= 680.0
						dbc.f9 >= 0.0492 AND wdbc.f24 >= 97.4
						dbc.f9 >= 0.0492 AND wdbc.f22 >= 14.87
						dbc.f9 >= 0.0492 AND wdbc.f3 >= 14.1
						dbc.f25 >= 739.2 AND wdbc.f8 >= 0.069
						dbc.f24 >= 105.95
						dbc.f24 >= 105.95
						dbc.f24 >= 105.95
						dbc.f9 >= 0.0492 A
						dbc.f24 >= 105.95
						dbc.f24 >= 105.95
						dbc.f25 >= 739.2 A
						dbc.f24 >= 105.95
						dbc.f24 >= 105.95
						dbc.f24 >= 105.95

Figure 3: The Safarii Multi-Relation Data Mining environment.

alternative patterns with identical binary features. It is interesting to see why such patterns are the same or similar, syntactically or semantically.

5. LeGo in the Safarii System

In this section, we describe some of the pattern discovery techniques implemented in the Safarii system (Knobbe, 2006), with the aim of showing an example system that implements the LeGo approach in practice. The Safarii system, developed by the third author and colleagues, is an extensive knowledge discovery environment for analyzing large data stored in relational databases. It was originally designed with a specific focus on Multi-Relational Data Mining, although that functionality has now mostly reached maturity. The majority of recent developments in the system are based on the pattern discovery and combination techniques described in this paper. These techniques are mostly independent of the specific pattern language selected, so a range of data formats can be considered, including graphical, multi-relational or simply propositional.

The system provides a range of facilities for each of the three phases described in Section 3.1 (along with other KDD-process related functions such as data pre-processing and model deployment). In each phase, the desired operations can be selected independently of the other phases. For the *Local Pattern Discovery* phase, Safarii provides a generic Subgroup Discovery algorithm similar to the one described in Section 3.1. This algorithm can be executed in a variety of ways, depending on parameters concerning the nature of the patterns discovered, the pattern language, the search strategy and so on. Additionally, the algorithm offers a range of common inductive constraints on the patterns to be discovered, such as quality, support and complexity. As the data that Safarii is designed to work with typically is complex in nature (e.g. relational, numeric or high-dimensional), and implies extremely large search spaces, exhaustive search is generally not an option. The algorithm therefore implements as its default search strategy a beam search, which is relatively efficient while not being too sensitive to the normal pitfalls of hill-climbing methods. For reasons of stability and scalability, all mining operations are expressed in terms of data mining queries that are processed inside the database. This puts the bulk of the computational burden on the RDBMS, which is optimized for such tasks, and potentially runs on a large dedicated server.

For the *Pattern Set Discovery* phase, Safarii implements a Pattern Team approach. A total of 8 quality measures are available, most of which are described in (Knobbe & Ho, 2006b). A few of these, including Joint Entropy, can be computed efficiently, and therefore offer a reasonable and quick solution. Most of the supervised measures, notably the wrapper-approaches which employ a separate classification procedure to judge pattern sets, require exhaustive search, which limits their applicability in the case of large pattern collections. Still, the supervised measures are the method of choice if predictive power is the key objective for Global Modeling. All classifiers that are available in the subsequent phase can be used in the Pattern Set Discovery phase as part of a wrapper.

Finally, in the *Global Modeling* phase, Safarii offers two classification procedures that combine patterns (which can be thought of as binary features) into predictive models. At this stage, the user has the option of applying the classifier to the original set of patterns, or the selected subset. The two learning algorithms available are Decision Table Majority (DTM) classifiers (Knobbe & Ho, 2006b) and Support Vector Machines (SVMs) using linear kernels. This allows for either high or low expressive power of the classifier, respectively. This choice is related to the extent of the initial discovery phase and the number of local patterns provided to the global modeling procedure. With extensive search, already a large part of the complexities of the dataset have been made explicit as patterns, suggesting a global model of low expressiveness (SVM). On the other hand, with shallow patterns being discovered, more extensive modeling is required (DTM) to capture possible complex interactions between patterns. As a third Global Modeling method, a Bayesian network can be induced, where the patterns form the nodes in the network. This method is typically applied to all patterns (thus skipping the intermediate phase), although the pattern team can be indicated as specific nodes in the network (see Figure 3, dark nodes). This pattern network thus conveys which are the essential patterns to be considered, and how alternative patterns relate to these and to other patterns.

6. Research Problems

In this section, we will discuss current research that is related to this framework. It is meant to both give an overview of the problems that need to be solved, as well as current work that addresses these issues.

6.1 Constraint Specification and Propagation

A key challenge is how to propagate back constraints that are imposed upon the global model into the earlier phases, as illustrated in Figure 4. Typically, we will be given constraints on the global model. The constraints can come in various different forms. For example, *optimality constraints* specify that the returned model should optimize some quality criterion, such as predictive accuracy, the area under the ROC curve, a cluster diversity measure, etc. Typically, one is only interested in the best global model, but in some scenarios it could also be of interest to return the best k models (*k-optimality*) or all models above a specified quality threshold (*quality constraints*).

In any case, the specified constraints can only be directly used in the Global Modeling phase. However, it might be advisable to optimize the preceding phases of Local Pattern Discovery and Pattern Set Discovery towards the particular global performance goal. From the same set of local patterns, different pattern sets might be selected for different goals, such as classification and clus-

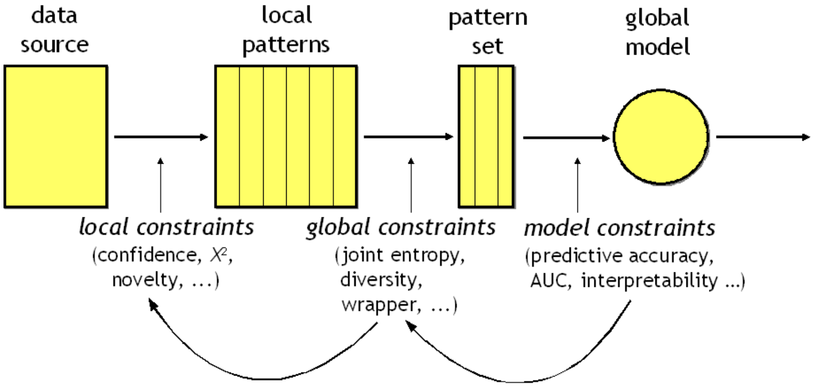

data source
local patterns
pattern set
global model
local constraints
(confidence, X^2, novelty, ...)
global constraints
(joint entropy, diversity, wrapper, ...)
model constraints
(predictive accuracy, AUC, interpretability ...)

Figure 4: Constraints have to be propagated back through the different phases

tering. Likewise, different evaluation measures for local patterns might be relevant for obtaining optimal candidates for obtaining different goals in pattern team discovery.

How global constraints can be propagated back into constraints on the local models is largely an open research question. Consider, again, the case of inductive rule learning, where typically the goal is to maximize predictive accuracy on unseen data. The heuristics that are used by conventional covering algorithms for local evaluation of the rules have typically been derived from first principles. In one way or another, they measure the purity of the covered examples, but the optimization of their joint coverage is entirely left to the covering loop and not addressed in the heuristic functions. The question of the importance of coverage information in local heuristics has only recently been addressed thoroughly. For example, Janssen and Fürnkranz (2006) have systematically compared three types of parameterized local rule learning heuristics which trade off precision and coverage in different ways (the m-estimate, the F-measure, and the Klösgen measures) and found that even though the heuristics have quite a different behavior, a global optimization of their parameters results in very similar measures. In further work, Janssen and Fürnkranz (2007) have obtained similar results with meta-learning a rule learning heuristic without restricting it to a certain parametric form. They also observed evidence that a proper formulation of the rule learning problem as a search problem with delayed rewards, may result in a better performance, as suggested in (Fürnkranz, 2005).

Alternatively, one may try to iteratively restart the search in order to detect local patterns that are useful in the context of previously found patterns. The pruning phase of the Ripper rule learning algorithm (Cohen, 1995) implements a technique that repeatedly deletes one rule and re-learns it in the context of all other previously found rules. Knowledge-based sampling (Scholz, 2005) is a technique that allows to mine subgroups that are interesting in the context of prior knowledge, in particular in the context of other, previously discovered patterns or given predictive models. Transforming the distribution underlying the data is a convenient way to direct the search for subgroups towards novel patterns, and to increase the diversity of mined rule sets without changing the underlying data mining algorithm. This strategy is mostly agnostic about the objective function used by the underlying pattern discovery algorithm. For the specific case of iteratively optimizing weighted relative accuracy in each step, the Subgroup Discovery task has been shown to coincide with a variant of AdaBoost (Scholz, 2006); despite explicitly mining (local) Subgroup patterns, the learner favors patterns in a greedy fashion that minimize the total example weight, a known upper-bound for error rate and the number of misranked pairs, closely related to the area under the ROC curve. Hence, this strategy bridges the gap between the mainly descriptive nature of the Subgroup Discovery task and predictive data analysis, and constitutes an example of how to back-propagate global constraints into the local pattern mining phase.

Based on the above discussion, we can identify the following problems that need to be addressed within our framework:

Specification of Constraints: What types of constraints can be used in each phase, and how can they be specified? Evaluation metrics for local and global models have been investigated in quite some depth. For the Pattern Set Discovery task, however, it is still quite unclear what types of constraints can be defined and what effects they will have.

Propagation of Constraints: How can global constraints be propagated back to local constraints? What type of local patterns must be found in order to guarantee a high performance on the global modeling task? Which local constraints optimize which global constraints?

General-Purpose Constraints: A key advantage of the modular approach could be that local patterns may be mined independently and can be re-used for several Global Modeling tasks. Are there general local constraints that give a reasonable performance on a wide variety of Global Modeling tasks?

6.2 Efficient Pattern Set Discovery

An important research question in the context of Pattern Set Discovery is concerned with computational efficiency. As an exponential number of subsets exist, exhaustive methods will only work for small pattern collections. For specific quality measures, such as joint entropy, relatively tight upper bounds can be given (Knobbe & Ho, 2006a), that can be used to discard directly the majority of candidate sets. Unfortunately, when many (largely) identical patterns abound, such pruning methods break down. As an approximate solution, one can consider greedy selection methods, reminiscent of forward feature selection methods, that exhibit computation times quadratic in the number of patterns involved. For certain classes of quality measures, such greedy search can provide near-optimal solutions (Mielikäinen & Mannila, 2003). Knobbe and Ho (2006a) argue that in the case of Joint Entropy, very good approximations can be achieved efficiently, particularly compared to the running times of exact solutions. In a recent publication, Zimmermann and Bringmann (2007) give a canonical forward selection algorithm that linearly scans an ordered list of patterns, and for each pattern decides on the added value of a new pattern given the currently selected patterns. Different measures for this added value are presented. In a comparison, they demonstrate that their greedy approach produces results similar in many respects to pattern teams using joint entropy (exact solution).

6.3 Local Pattern Discovery

A particularly important difference between local pattern discovery and global modeling is that the former is traditionally framed as a descriptive induction task, whereas the latter is predictive. Recently, several works have addressed this problem of the predictive validity of local patterns (Scheffer, 2005; Mozina et al., 2006). For example, Webb (2007) brought out-of-sample evaluation, which is standardly used in global modeling, to the local pattern discovery phase with the goal of ensuring the statistical validity of the discovered patterns. Janssen and Fürnkranz (2007) tried to model the predictive performance of individual rules by learning to predict the performance of a rule on an independent test set.

Another important research issue is efficiency and scalability. Three types of search strategies for Subgroup Discovery can be found in the literature: exhaustive, probabilistic, and heuristic search. At this point we will only list algorithms that provide reasonable guarantees. The multi-relational MIDOS algorithm (Wrobel, 1997) is an example of an exhaustive search algorithm. It applies only safe pruning, and hence reliably identifies the best subgroups in terms of its objective function, the weighted relative accuracy. A concern with exhaustive search is its time complexity, so more recent work has mainly focused on less expensive strategies. For example, the SD-Map algorithm (Atzmüller & Puppe, 2006) utilizes the FP-growth data structure (Han et al., 2000) to improve efficiency,

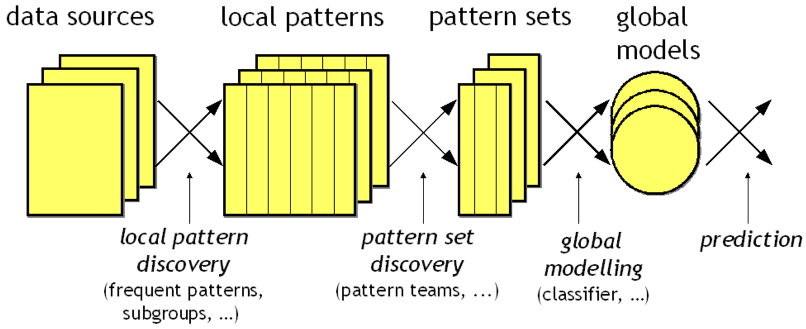


Figure 5: The Sequential Parallel Universes View

while still searching exhaustively. For most objective functions, the probabilistic search strategy of adaptive sampling helps to speed up the pattern mining process considerably, while still allowing for strong probabilistic guarantees when mining from large databases (Scheffer & Wrobel, 2002). Unlike classical Subgroup Discovery, the theoretical framework of adaptive sampling implicitly addresses the generalization performance of rules, even when optimizing criteria that are traditionally used in the context of descriptive tasks. The results are sets of probably approximately best subgroups. The Knowledge-Based Sampling technique described above uses sampling or reweighting also to filter out previously found patterns (Scholz, 2005).

6.4 Parallel Universes

A further natural extension of the model is that in each of these phases, we can have a $m : n$ relationship, i.e., we can start with an arbitrary number of inputs, and have an arbitrary number of results. This is illustrated in Figure 5.

With this extension, many commonly used techniques that make use of multiple characterizations of the same type of objects can be incorporated into this framework. Following (Berthold et al., 2007), we use the term *parallel universes* for any type of multiple characterizations of the same objects that can be used in the learning process. For example, the data mining phase could start with multiple aligned data sources. An example for this would be multilingual text corpora, which contain multiple translations of each document. These different versions of the same document could be merged into the same set of documents, could be kept separately, or one could, e.g., consider to merge the documents of the same language group.

Consequently, we can also have multiple types of local patterns. For example, we could mine each data source separately for local patterns. However, multiple types of local patterns can also be generated from a single data source. For example, we could use multiple local pattern discovery algorithms, employ different quality criteria for the search, or generate multiple views on the data (e.g., via random sampling of the available data attributes). For subsequent use in Pattern Set Discovery or Global Modeling, the multiple local pattern sets can later be pooled together, or be kept separately as in multi-view learning (Rüping & Scheffer, 2005).

The goal of the Pattern Set Discovery phase is to select a set of patterns that are useful in the context of other patterns. Again, one may want to find a single good pattern team (Knobbe & Ho, 2006b) or multiple pattern teams (De Raedt & Zimmermann, 2007) from single or multiple sources of local patterns.

Finally, single or multiple pattern teams may be used to form a single or multiple global models. In the realm of classification, ensemble techniques for forming and using multiple models are quite commonly used and do not need to be elaborated here. Note, however, that multiple global models may also be of interest to other learning tasks, such as clustering and descriptive modeling.

7. Concluding remarks

In this paper we proposed an abstract data mining framework based on the notion of local patterns. The main characteristic of this locality is that patterns are computed with respect to a given objective function, but without taking the context of other patterns into consideration. In subsequent steps this context is established in terms of optimal pattern subset selection and the computation of pattern combinations that result in one or more well suited global models. The stepwise refinement of our proposed search strategy can well be expressed in terms of different constraints, ranging from local pattern selection strategies to the objective function we finally aim to optimize with our final global model(s). These different constraints are interleaved in a non-trivial way, but allow to guide and narrow down the search in each step so that the resulting partial optimization problems become tractable in practice.

We believe that one of the main advantages of our frameworks lies in its generality. It leverages a number of popular techniques in a natural way that are traditionally hosted in different communities. In particular, our framework allows to utilize frequent itemset mining and subgroup discovery, information-theoretic and other techniques known from ensemble methods to select orthogonal, hence highly informative sets of features, and a plethora of different model combination techniques on top of these highly valuable features. Referring to the notion of parallel universes we also illustrated connections to the relatively young field of multi-view learning.

A key challenge for future work is to understand the trade-off between exploratory search and generalization power in the LeGo framework. Starting a global modeling phase with a large number of local patterns comes, of course, also with increased computational costs. In many applications, the costs for exhaustive approaches for both the Local Pattern Discovery and Pattern Set Discovery phases may be prohibitive, while, on the other hand, too greedy approaches may loose important information. A further exploration of this trade-off seems to be a particularly promising research goal. A possible road to follow could be to propagate global constraints back to the pattern team formation and local pattern discovery phases and use them there to focus the search.

Acknowledgments

Parts of this work have been funded by the ACI ”masse de données” (French Ministry of research), Bingo project (MD 46, 2004-07), and the German Science Foundation (DFG). We are grateful to Michael Berthold, Katharina Morik, and Arno Siebes for organizing a very inspiring Dagstuhl seminar on “Local Pattern Discovery and Parallel Universes”.

References

Atzmüller, M., & Puppe, F. (2006). SD-Map: A fast algorithm for exhaustive subgroup discovery. *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-06)* (pp. 6–17). Springer-Verlag.

Bayardo Jr., R. J. (1997). Brute-force mining of high-confidence classification rules. *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97)* (pp. 123–126).

Berthold, M. R., Morik, K., & Siebes, A. (Eds.). (2007). *Parallel universes and local patterns.* Dagstuhl Seminar No. 07181.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Bringmann, B., Zimmermann, A., De Raedt, L., & Nijssen, S. (2006). Don't be afraid of simpler patterns. *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-06)* (pp. 55–66). Berlin, Germany: Springer-Verlag.

Calders, T., Rigotti, C., & Boulicaut, J.-F. (2005). A survey on condensed representations for frequent sets. *Constraint-Based Mining and Inductive Databases* (pp. 64–80). Springer-Verlag.

Chi, Y., Muntz, R. R., Nijssen, S., & Kok, J. N. (2005). Frequent subtree mining - an overview. *Fundamenta Informaticae*, *66*, 161–198.

Cohen, W. W. (1995). Fast effective rule induction. *Proceedings of the 12th International Conference on Machine Learning (ML-95)* (pp. 115–123). Lake Tahoe, CA: Morgan Kaufmann.

De Raedt, L. (2002). A perspective on inductive databases. *SIGKDD explorations*, *4*, 69–77.

De Raedt, L., Jäger, M., Lee, S. D., & Mannila, H. (2002). A theory of inductive query answering. *Proceedings of the IEEE Conference on Data Mining (ICDM'02)* (pp. 123–130). Maebashi, Japan.

De Raedt, L., & Zimmermann, A. (2007). Constraint-based pattern set mining. *Proceedings of the 7th SIAM International Conference on Data Mining (SDM-07)*. Bethesda, MD.

Durand, N., & Crémilleux, B. (2002). ECCLAT: a New Approach of Clusters Discovery in Categorical Data. *Proceedings of the 22nd International Conference on Knowledge Based Systems and Applied Artificial Intelligence (ES-02)* (pp. 177–190). Cambridge, UK.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, *17*, 37–54.

Forman, G. (2003). An extensive empirical study for feature selection in text classification. *Journal of Machine Learning Research*, *3*, 1289–1305.

Fürnkranz, J. (2005). From local to global patterns: Evaluation issues in rule learning algorithms. *Local Pattern Detection* (pp. 20–38). Springer-Verlag.

Fürnkranz, J., & Flach, P. (2005). ROC 'n' rule learning – Towards a better understanding of covering algorithms. *Machine Learning*, *58*, 39–77.

Goethals, B. (2005). Frequent set mining. In *The data mining and knowledge discovery handbook*, chapter 17, 377–397. Springer.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-00)* (pp. 1–12). ACM Press.

Hand, D. J. (2002). Pattern detection and discovery *Proceedings of the ESF exploratory workshop on pattern detection and discovery in data mining* (pp. 1–12). Springer-Verlag.

Hofmann, T. (1999). Probabilistic latent semantic analysis. *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-99)* (pp. 289–296). Stockholm, Sweden.

Imielinski, T., & Mannila, H. (1996). A database perspective on knowledge discovery. *Communications of the ACM*, *39*, 58–64.

Janssen, F., & Fürnkranz, J. (2006). On trading off consistency and coverage in inductive rule learning. *Proceedings of the LWA 2006, Lernen Wissensentdeckung Adaptivität* (pp. 306–313). Hildesheim, Germany: Gesellschaft für Informatik e. V. (GI).

Janssen, F., & Fürnkranz, J. (2007). Meta-learning a rule learning heuristic. *Proceedings of the IEEE Conference on Data Mining (ICDM-07).* Omaha, NE.

Jovanoski, V., & Lavrač, N. (2001). Classification rule learning with APRIORI-C. *Proceedings of the 10th Portuguese Conference on Artificial Intelligence (EPIA 2001)* (pp. 44–51). Porto, Portugal: Springer-Verlag.

Klösgen, W. (1996). Explora: A multipattern and multistrategy discovery assistant. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (Eds.), *Advances in knowledge discovery and data mining*, chapter 3, 249–272. Menlo Park, California: AAAI Press/The MIT Press.

Knobbe, A. J. (2006). Safarii multi-relational data mining environment. `http://www.kiminkii.com/safarii.html`.

Knobbe, A. J., & Ho, E. K. Y. (2006a). Maximally informative k-itemsets and their efficient discovery. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06)* (pp. 237–244). Philadelphia, PA.

Knobbe, A. J., & Ho, E. K. Y. (2006b). Pattern teams. *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-06)* (pp. 577–584). Berlin, Germany: Springer-Verlag.

Kramer, S., Lavrač, N., & Flach, P. (2001). Propositionalization approaches to relational data mining. In S. Džeroski and N. Lavrač (eds.) *Relational Data Mining*, 262–291. Springer-Verlag.

Lavrac, N., Flach, P., & Zupan, B. (1999). Rule evaluation measures: A unifying view. *Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP-99).* Springer-Verlag.

Li, W., Han, J., & Pei, J. (2001). CMAR: Accurate and efficient classification based on multiple class-association rules. *Proceedings of the IEEE Conference on Data Mining (ICDM-01)* (pp. 369–376).

Lindgren, T., & Boström, H. (2004). Resolving rule conflicts with double induction. *Intelligent Data Analysis*, *8*, 457–468.

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association rule mining. *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98).*

Liu, B., Ma, Y., & Wong, C.-K. (2000). Improving an exhaustive search based rule learner. *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000)* (pp. 504–509). Lyon, France.

Mielikäinen, T., & Mannila, H. (2003). The pattern ordering problem. *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-03)* (pp. 327–338). Cavtat-Dubrovnik, Croatia: Springer-Verlag.

Morik, K., Boulicaut, J.-F., & Siebes, A. (Eds.). (2005). *Local pattern detection*, vol. 3539 of *Lecture Notes in Computer Science*. Springer-Verlag.

Mozina, M., Demsar, J., Zabkar, J., & Bratko, I. (2006). Why is rule learning optimistic and how to correct it. *Proceedings of 17th European Conference on Machine Learning (ECML-06)* (pp. 330–340).

Mutter, S., Hall, M., & Frank, E. (2004). Using classification to evaluate the output of confidence-based association rule mining. *Proceedings of the Australian Joint Conference on Artificial Intelligence (AI-05)* (pp. 538–549). Cairns, Australia: Springer-Verlag.

Ng, R. T., Lakshmanan, V. S., Han, J., & Pang, A. (1998). Exploratory mining and pruning optimizations of constrained associations rules. *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-98)* (pp. 13–24). ACM Press.

Nijssen, S., & Kok, J. N. (2004). Frequent graph mining and its application to molecular databases. *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics* (pp. 4571–4577). The Hague, Netherlands.

Pensa, R., Robardet, C., & Boulicaut, J.-F. (2005). A bi-clustering framework for categorical data. *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-05)* (pp. 643–650). Porto, Portugal.

Rückert, U., & Kramer, S. (2007). Optimizing Feature Sets for Structured Data. *Proceedings of the 18th European Conference on Machine Learning (ECML-07)*, volume 4212 of *Lecture Notes in Computer Science*, pp. 716–723. Springer-Verlag.

Rüping, S., & Scheffer, T. (Eds.). (2005). *Proceedings of the ICML-05 Workshop on Learning with Multiple Views*. Bonn Germany.

Scheffer, T. (2005). Finding association rules that trade support optimally against confidence. *Intelligent Data Analysis*, *9*, 381–395.

Scheffer, T., & Wrobel, S. (2002). Finding the most interesting patterns in a database quickly by using sequential sampling. *Journal of Machine Learning Research*, *3*, 833–862.

Scholz, M. (2005). Sampling-based sequential subgroup mining. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-05)* (pp. 265–274). ACM Press.

Scholz, M. (2006). Boosting in PN spaces. *Proceedings of the 17th European Conference on Machine Learning (ECML-06)* (pp. 377–388). Springer-Verlag.

Sevon, P., Toivonen, H., & Ollikainen, V. (2006). Treedt: Tree pattern mining for gene mapping. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *3*, 174–185.

Siebes, A., Vreeken, J., & van Leeuwen, M. (2006). Item sets that compress. *Proceedings of the 6th SIAM International Conference on Data Mining (SDM-06)*, SIAM.

Soulet, A., & Crémilleux, B. (2005). An efficient framework for mining flexible constraints. *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05)* (pp. 661–671). Hanoi, Vietnam: Springer-Verlag.

Soulet, A., Klema, J., & Crémilleux, B. (2007). Efficient Mining under Rich Constraints Derived from Various Datasets. *Proceedings of the 5th International Workshop on Knowledge Discovery in Inductive Databases (KDID-06)*, vol. 4747 of *Lecture Notes in Computer Science*. Springer-Verlag.

Tan, P.-N., Kumar, V., & Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)* (pp. 32–41). Edmonton, Alberta.

Wang, K., Chu, X., & Liu, B. (1999). Clustering transactions using large items. *Proceedings of ACM CIKM International Conference on Information and Knowledge Management (CIKM-99)* (pp. 483–490). Kansas City, Missouri.

Webb, G. I. (2007). Discovering significant patterns. *Machine Learning*, *68*, 1–33.

Wrobel, S. (1997). An algorithm for multi–relational discovery of subgroups. *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD 97)* (pp. 78–87). Springer-Verlag.

Yin, X., & Han, J. (2003). CPAR: Classification based on predictive association rules. *Proceedings of the 3rd SIAM Conference on Data Mining (SDM-03)*. San Francisco, CA.

Zaki, M. J. (2001). Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, *42*, 31–60.

Zimmermann, A., & Bringmann, B. (2007). The chosen few: On identifying valuable patterns. *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM-07)*. Omaha, NE.

Zimmermann, A., & De Raedt, L. (2004). Corclass: correlated association rule mining for classification. *Proceedings of the 7th International Conference on Discovery Science (DS'04)* (pp. 60–72). Springer-Verlag.