

# Pattern-Based Classification: A Unifying Perspective

Björn Bringmann, Siegfried Nijssen, and Albrecht Zimmermann

Department of Computer Science  
Katholieke Universiteit Leuven  
Celestijnenlaan 200A, 3001, Leuven, Belgium

**Abstract.** The use of patterns in predictive models is a topic that has received a lot of attention in recent years. Pattern mining can help to obtain models for structured domains, such as graphs and sequences, and has been proposed as a means to obtain more accurate and more interpretable models. Despite the large amount of publications devoted to this topic, we believe however that an overview of what has been accomplished in this area is missing. This paper presents our perspective on this evolving area. We identify the principles of pattern mining that are important when mining patterns for models and provide an overview of pattern-based classification methods. We categorize these methods along the following dimensions: (1) whether they *post-process* a pre-computed set of patterns or *iteratively* execute pattern mining algorithms; (2) whether they select patterns *model-independently* or whether the pattern selection is *guided by a model*. We summarize the results that have been obtained for each of these methods.

## 1 Introduction

Important problems in data mining and machine learning are classification and pattern mining. In recent years an increasing number of publications have studied the combination of these problems. The main idea in these methods is that patterns can be used to define features or can be used as rules; classification models which make use of these features or rules may be more accurate or more simple to understand. Last, but not least, in structured domains, pattern mining can be considered a propositionalization approach which enables the use of propositional data mining and machine learning algorithms.

Despite the large amount of publications devoted to this topic, we believe however that an overview of what has been accomplished in this area is missing. It is not uncommon for publications in this area to refer to only a small portion of relevant related work, hence preventing deeper insight or a general theory from evolving. As an example, Kralj et al. pointed out that the problems of subgroup discovery, contrast set mining and emerging pattern mining are so similar that their main differences are arguably the terminology used [38]. We believe that this phenomenon is much more wide-spread. For instance, in this paper we will point out that the independently proposed areas of correlating (or correlated)

itemset mining and discriminative itemset mining are also mostly identical to the problems studied in [38], except in name.

The need to obtain a better insight in the accomplishments of this area has been observed by other authors. In particular, this has led to a tutorial at ICDM'07 by Bailey and Dong [3] (and an extensive online reference list), a tutorial at ICDM'08 by Cheng et al. [14] and a workshop at ECML PKDD [35]. In this paper, we present our perspective on this area, which differs from earlier perspectives in several key aspects.

**Pattern Type Independence:** Other overviews have stressed the fact that there are different types of data, such as graph-based, tree-based and itemset-based data. They coupled pattern selection strategies to particular pattern types, and stressed the fact that different pattern mining algorithms are needed to deal with each such data type. Even though this is true, and indeed one often needs to implement a different pattern miner to deal with a pattern type at hand, we believe that it is more important in this case to stress the conceptual similarities between these pattern mining algorithms. Doing so leads to the insight that most approaches that have been proposed for complex data types, such as graphs, can easily also be implemented in pattern mining algorithms for simpler data types, such as itemsets; this leads to a large number of additional approaches that itemset-mining based approaches could be compared with.

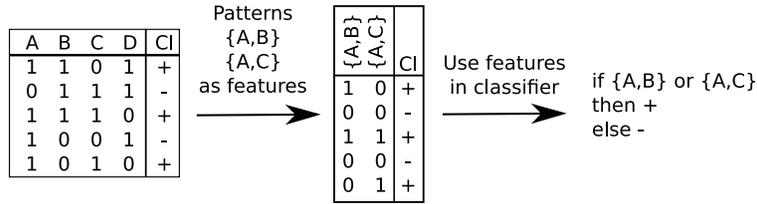
**Data Structure Independence:** In a similar way, other tutorials have stressed the fact that even for the same data type, such as itemset data, different data structures may be used to speed-up the computation of the patterns; examples are the FP-Trees [14] and ZBDDs [3]. Even though the choice for such data structures may have a significant impact on the efficiency of the computation, we believe that most pattern-based classification problems are orthogonal to the choice of such data structures: most solutions can be combined with any such data structure.

**Iterative Mining:** Initial approaches which combined pattern mining and classification models took a strict step-wise approach, in which a set of patterns is computed once and these patterns are subsequently used in models. However, in more recent years a large number of methods have been proposed which aim at integrating pattern mining, feature selection and model construction. In this paper we give a central position to such approaches.

In Section 2 we present the key components of our proposed framework. The state-of-the-art of these components is discussed in more detail in subsequent sections.

## 2 Overview

The main idea of pattern-based classification is that patterns define new features, which can be used in a classification model. A simple example is provided in the figure below.



Essentially, a pattern is a regularity that is observed in a number of examples, in our example  $\{A, B\}$  is a pattern that occurs in the first example and third example. Whether this regularity is present or not in an example can be seen as a feature of each example. A prediction can be based on this, for instance, if an example includes items  $A$  and  $B$  we may predict the example to be positive.

The key challenges in finding pattern-based models are:

- how to find a set of patterns;
- how to combine patterns into models.

We distinguish approaches in the literature along the following dimensions:

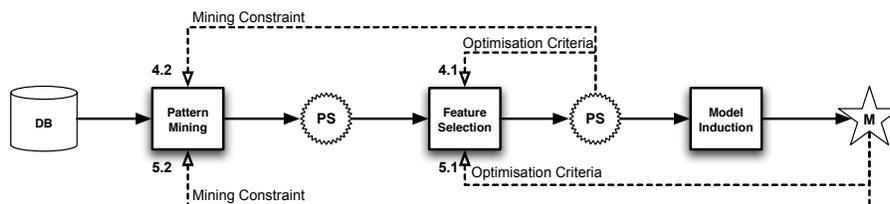
**Iterative Mining or Post-Processing:** when a set of patterns is constructed, this can be done in two ways. We can run a pattern mining algorithm once to find a large set of patterns, and post-process its result to obtain a smaller set, or we can iteratively run a pattern mining algorithm, in each round finding a very small number of patterns (often only one), taking into account previous patterns in each round.

**Model-Dependence or Model-Independence:** when we search a set of patterns, we can use two types of criteria. We can use criteria that *explicitly* take into account the type of model in which the pattern will be used, or we can use criteria which are independent of the model; typically, in such a model-independent approach the aim is to find a set of patterns which is sufficiently diverse such that a more complex model, like an SVM, can be learned on the new features.

Many approaches have been developed along each of these dimensions. We will provide an overview of these approaches in Sections 4.1, 5.1, 4.2 and 5.2. The following table clarifies how these sections correspond to these dimensions.

	MODEL-DEPENDENT	MODEL-INDEPENDENT
POST-PROCESSING	Section 4.1	Section 5.1
ITERATIVE	Section 4.2	Section 5.2

The general search strategy for all these approaches can be summarized as in Figure 1. Starting from a given set of examples the first step is to mine for patterns  $PS$  satisfying given constraints. In a following, optional, step a subset of these patterns is selected in order to optimize this set of patterns as features in a model. Eventually those patterns are used as features to induce a model  $M$  in a third step. The process allows for several ways of feedback. Each of the intermediate results can be evaluated with regard to its quality to derive



**Fig. 1.** The overall process from pattern mining over feature selection to model induction. The dashed arrows show four possibilities of steering the process resulting from the two different dimensions we identified.

constraints which can be used to guide the search/selection of further patterns or even to restart the mining or selection step with adjusted parameters. If this feedback involves explicit consultation of the induced model, we refer to the aforementioned model-dependent methods, otherwise to model-independent ones. Most approaches in the literature can be described with this model and use one of the four different types of self-steering in the process which we will discuss in the subsequent sections.

In all these approaches, an important problem is how to find one or more patterns, iteratively or not. The simplest approach is the post-processing approach which operates on frequent patterns. However, most approaches are more sophisticated and take the class attribute into account while mining patterns. Hence, an important question in all of these approaches, iterative, model-dependent or not, is how to find patterns that take a class attribute into account. We start with an overview of solutions to this problem in Section 3.

### 3 Class-Sensitive Patterns

The starting point for taking class labels into account is in all cases to compute the (possibly weighted) support of a pattern in all classes individually. One can distinguish these approaches for using the class-specific supports in constraints:

- support constraints per class, for instance, a minimum support constraint on one class combined with a maximum support constraint on another class. Such constraints can involve explicit thresholds, as for *version space patterns* [39], a minimum difference between support values for *emerging patterns*, or a maximum support of zero for an individual class as for *jumping emerging patterns* [41, 21].
- constraints on scores computed from supports, sometimes in addition to support constraint. Many alternative measures for correlation strength have been proposed, ranging from *confidence*, *lift*, *weighted relative accuracy* or *novelty*, to  $\chi^2$ , the *correlation coefficient*, *information gain*, *Fisher score* and others, including measures derived from classification models, such as in gBoost [52].

Patterns satisfying constraints on derived scores have been called *emerging patterns* [18], *subgroup descriptions* [34, 71, 27], *contrast sets* [5], *correlating patterns* [48], *discriminative patterns* [15], and *interesting rules* [6, 47]. In this case one may not be interested in finding all patterns satisfying the constraints. Instead, one may be interested in finding top- $k$  scoring patterns, or finding top- $k$  patterns per instance in the training data [67].

It was pointed out in [38] that contrast sets, emerging patterns and subgroups are compatible terms, in the sense that these terms serve the same purpose of denoting patterns that score high with respect to a scoring function that takes class labels into account. A similar observation can be made regarding correlated patterns, discriminative patterns and interesting patterns. In this paper we do not endeavour to make a choice for one of these terms; to avoid this we will call such patterns *class-sensitive patterns* for the course of this paper. Whether the community should agree on a common name, and which one this should be, is not an issue we wish to discuss here.

In many cases threshold-based constraints are not effective enough to obtain smaller, non-redundant sets of patterns. One means to obtain smaller sets of patterns is to extend condensed representations, such as *closed*, *free* and *non-derivable* patterns [60, 7, 12], to the context of class-sensitive patterns [69, 67, 25].

Given the similarity in purpose of these patterns, it is not surprising that similar search strategies have been developed for each of them. Approaches that have been studied include post-processing frequent itemsets [2, 33, 15, 43] (for subgroups, discriminative patterns, interesting patterns, emerging patterns, in some cases with an additional support threshold), *branch-and-bound* search [69, 71, 4, 48, 67, 16, 1, 27, 52] (for subgroups, correlated patterns, contrast sets, discriminative patterns, gBoost), or variations of *iterative deepening* [11, 73, 13] (for correlated patterns, discriminative patterns). The reason that branch-and-bound searches have been proposed for class-sensitive pattern mining is that finding such patterns has been proved to be computationally hard. Proofs can be found in [48, 68].

A main difference between the papers studying class-sensitive patterns, is the choice for the scoring function. For instance, weighted relative accuracy is commonly used in subgroup discovery, while  $\chi^2$  is common in correlated pattern mining. Insight in the differences between these measures can be obtained by comparing them in ROC space [22, 51, 50]. Among others, such studies allow to compare how well the different measures can be bounded in a branch-and-bound search. Furthermore, such studies led to the insight that for some pattern domains (such as itemsets) better bounds exist than for other pattern domains [50]; however, *all* bounds introduced before [50] are pattern-domain independent, allowing for the application of existing strategies.

Despite that most bounds are pattern-domain independent, the combination of such bounds with approaches for dealing with particular pattern domains has received significant attention. Pattern domains that have been studied are item-set or attribute-value data (including [15, 48, 5, 6, 21]), sequences [11, 31], tree-

structured patterns [76, 30], and graphs (including [11, 26, 73, 52]). Approaches for class-sensitive itemset mining have been implemented using optimized data structures such as FP-trees [28, 16, 2] or binary decision diagrams (BDDs) [44]. In the remainder of this paper, we will present approaches independent of the data type or data structure for which they were proposed, as most approaches are conceptually independent of this.

An issue which has received limited attention is that of false positives. It is likely that in an exhaustive branch-and-bound search a pattern with a high score can be found, but this pattern may overfit the training data. How to control this error seems to be an open question; initial approaches suggest for instance to modify the scoring function [5, 70].

## 4 Model-Independent Pattern Selection

Mining class-sensitive patterns is usually the easy part, however. In many settings the number of patterns that is found is too large, in the sense that building classifiers on them is inefficient, overfitting is likely, and interpretability of the models may be hard. For patterns to be actually useful, there is the need to create a more compact set of effective patterns. If we do not take the subsequent model into account, the main aim of the *pattern selection* step is to reduce the redundancy of the pattern set. We can distinguish approaches which achieve this by post-processing an initial set of patterns, and approaches which iteratively search for patterns that increase the diversity of the pattern set.

### 4.1 Model-independent post-processing



**Fig. 2.** The model-independent post-processing approach: feedback is given by evaluation of the partially selected subset to steer feature selection.

Post-processing the result of pattern mining according to certain criteria has two distinct advantages: first, both for finding the initial set of patterns as for reducing this set, we can use or adapt existing well-developed pattern mining techniques, which are usually rather efficient. Second, it is in many cases possible to explicitly control the properties of the resulting pattern set or at least to give guarantees about them.

A variety of measures and constraints, and algorithms finding sets that satisfy them, have been proposed so far. In many cases, one is interested in finding a set of patterns that optimize a global criterion of diversity based on the occurrences of patterns in the data, sometimes in addition to explicit constraints. An example of a global criterion is *entropy*: if we select  $n$  patterns, we can encode every example in the data with a bit-vector of length  $n$ . This gives every bit-vector of

length  $n$  a probability in the data. The entropy of this distribution can be used as a measure of diversity. Such sets of diverse sets can be searched exhaustively [36, 37, 54]. In practice, these approaches do not scale well, and more greedy search strategies are needed. While the focus often is somewhat different, the general technique for selecting a subset of patterns by post-processing is very similar to filter approaches for *feature selection*.

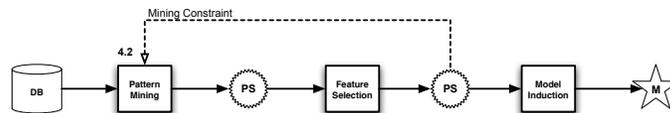
Initial proposals for measures of diversity did not provide for approximation guarantees that the pattern sets found were provably good [19]. However, more recently several pattern set criteria have been shown to be submodular, and consequently a greedy hill-climbing algorithm, which iteratively adds a highest scoring pattern to an initially empty set, achieves a result which approximates the optimum [24, 61]. Other recent approaches attempt to reduce the computational complexity of the pattern selection further, and study different measures for selecting patterns [19, 9].

An alternative to data-only approaches is also to take into account the similarity between the pattern structures, hence taking into account mutual similarities between patterns. Also here one can define optimization criteria and greedy approaches for optimizing them [29, 72].

A third set of approaches does not optimize a measure of diversity directly, but rather aims at finding a compact representation of the data; the idea is here that we wish to find a small set of patterns which allows to encode transactions as accurately as possible with as few patterns as possible. One can distinguish the MDL based approaches here [59, 63], as well as the discrete basis problem [46, 45].

Finally, machine learning-inspired sampling and verification techniques may also be used to obtain more diverse sets of patterns [10].

## 4.2 Model-independent iterative mining



**Fig. 3.** The model-independent iterative approach: the feedback derived by evaluating the pattern set directly influences the mining of individual patterns.

The alternative to feature selection lies in *feature construction*. In general, the idea is here to avoid generating patterns beforehand, but to search for patterns during the selection process. The main advantage is that we only find patterns that have meaning in the presence of other patterns. This is not necessarily the case in the setting described in the former section since the pattern mining operation itself does not take into account the relationships between patterns, and may produce many patterns which could have been pruned if the pattern search was more aware of the subsequent pattern selection.

A first strategy is to adapt post-processing algorithms. Whereas greedy post-processing algorithms iteratively search for a pattern in a pre-computed set of

patterns, this search for patterns can in some cases also be performed by a pattern mining algorithm. The main observation is that given an already selected set of patterns, some scoring functions for measuring the diversity of a new pattern set are boundable, and hence we can use similar strategies to find new patterns as in class-sensitive pattern mining [57, 61], hence avoiding having to pre-compute a set of patterns.

An alternative approach lies in using a model-dependent iterative strategy (as discussed in Section 5.2); one can ignore the model produced by these strategies afterwards and use the patterns as features in other classification models [16]. The difference between model-dependent and -independent approaches is thus sometimes not as clear-cut as our terminology suggests.

## 5 Model-Dependent Pattern Selection

While all the methods described in the preceding select patterns and sets of patterns using scoring functions, these scoring functions are not influenced by the choice of model that will be constructed from the patterns. Even though in model-independent approaches patterns may be used in SVMs, which show very good accuracy and guard against overfitting to a certain degree, the resulting models are difficult to interpret. The alternative is to use patterns directly to predict class labels, giving users the advantage of being able to examine and interpret the model. When doing this, it is often advantageous to adapt the scoring function to the model in which the patterns will be used. Also here we distinguish the existing methods into post-processing and iterative approaches.

### 5.1 Model-dependent post-processing



**Fig. 4.** The model-dependent post-processing approach: feedback is given by the model itself to steer feature selection.

These approaches are also known as methods for *associative classification*. A variety of approaches have been proposed towards building a classifier from rules and performing predictions using selected patterns.

The simplest such approaches post-process *all* patterns found in a previous phase; they rely on a conflict resolution strategy, similarly to unordered rule lists, which often means that in order to predict which class an example belongs to, a score is computed for each class from the patterns for that class. Many such scoring strategies have been proposed [19, 67, 41, 75, 56, 1, 63]. In some cases, such as [63], another approach on the border between model-dependence and -independence, model-independent pattern selection takes place before patterns are used in such a voting scheme.

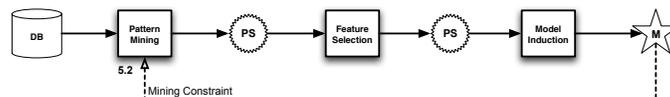
The alternative is to perform an ordered heuristic search over a set of patterns, guided by a database coverage constraint. In a sense this is a post-processing version of the sequential covering/weighted covering paradigm known in machine learning. Essentially, these approaches execute these steps:

1. they sort the patterns;
2. they select a pattern according to this sorting order;
3. they optionally remove some of the remaining unselected patterns;
4. they optionally resort remaining unselected patterns according to updated scores;
5. they recursively continue selecting a pattern.

Strategies implementing this idea have been studied in [43, 42, 76].

While these approaches construct a model greedily, [49] showed that itemsets can be post-processed to construct a decision tree optimally. In this approach, an itemset corresponds to a path from the root to a leaf in a decision tree. Itemsets are selected from a set such that the resulting tree is optimal given user-specified constraints and criteria.

## 5.2 Model-dependent iterative mining



**Fig. 5.** The model-dependent iterative approach: feedback given by the model influences which patterns are mined next.

In model-dependent, iterative mining techniques the connection to and inspiration by machine learning becomes most obvious. Hence, these approaches are best understood as adaptations of machine learning techniques. We can distinguish the following classification models.

*FOIL-like* decision list learning strategies: these are techniques that can be understood as adaptations of the FOIL rule learning technique, combined with the weighted covering metaheuristic [74].

*Decision tree* learning strategies: these methods adapt decision tree induction algorithms such as C4.5 [8, 26, 14]; they iteratively search for class-sensitive patterns that split data as well as possible according to criteria such as information gain; the search continues in parallel for the data sets resulting from the split.

*Instance-based* learning strategies: these are methods where pattern mining is delayed till a test example is given; class-sensitive patterns are searched that are relevant for the test instance [65, 64, 40].

*Boosting* strategies: these are methods in which classifications of patterns are weighted, and rules are found by iteratively reweighting examples [52, 53].

*Regression* strategies: these are methods in which predictions are based on weighted sums of patterns, and weights of patterns are found by linear regression [58]. The boosting and regression methods often include a regularization parameter which needs to be set. In [62] it was studied how to find the *regularization path*, which in this case can be seen as an ordered set of patterns, each prefix of which corresponds to a regression model for one choice of this parameter.

As pointed out, one can also choose to ignore the model constructed by any model-dependent strategy, and use the patterns as features in another type of model. The two categories show therefore different kinds of flexibility: while model-dependent results can be used both directly and as building blocks of another model, they are probably best suited to the model that was used to derive them, differing from the results of model-independent techniques.

## 6 Conclusions

In this paper we presented our perspective on the area of pattern-based classification. Key elements in our perspective are pattern type and data structure independence; instead, we propose to categorize approaches along two dimensions: whether they are model-dependent or model-independent, and whether they are iterative or non-iterative.

For almost any quality measure and mining techniques, both the pattern language and the language in which data are expressed are *not* relevant for the pattern set selection phase as long as there is a well-defined matching operator between the two. Furthermore, almost all techniques for mining class-sensitive patterns *themselves* are independent of these aspects as well, with the exception of data structures used. Such data structures, however, typically do not influence the applicability of mining techniques but only their implementation. This means that it is possible to transfer approaches freely between different representations and settings, albeit possibly at a certain cost of efficiency.

Iterative approaches have the advantage of taking the effects of already selected patterns into account by adjusting the scoring function in some way. This allows to focus on interesting areas of the pattern space, pruning subspaces that would have been explored in non-iterative mining, and visiting others that would have been ignored otherwise. The downside to this is that the space of potential solutions is far larger than in the non-iterative case, requiring the adoption of heuristic techniques and less control over, and looser guarantees for the quality of, resulting sets. Whereas early approaches were often model-dependent post-processing approaches, recent work more focuses on iterative approaches, both model-dependent and -independent.

The trade-off involved in model-dependence and -independence has been sketched in the preceding section: the agnosticism of model-independence means that resulting sets can be expected to be useful to different kinds of modeling techniques instead of being tailored towards a particular model as in model-dependent solutions. In addition, while predictive models can be used in scoring

functions, they do not exhaust the issue and therefore model-independent techniques can use measures that focus on different aspects of pattern relations and may eschew class labels completely. However, results that are produced by such approaches cannot be expected to be useful as direct predictors, making an additional more or less complex modeling step necessary which will probably reduce interpretability. Model-dependent techniques, on the other hand, usually result in models in which the relationship among particular patterns and between patterns and prediction are far more easily accessible. In addition, resulting pattern sets can still be used as input to a different modeling step but might perform worse than pattern sets produced by model-independent approaches.

A major issue in the current state-of-the-art is that so far it is not very clear to what degree the merits and drawbacks that can be derived analytically for different approaches materialize empirically. In most of the papers that proposed pattern-based classification algorithms, experiments were performed to show the benefits of the approaches. However, these comparisons were (understandably) often limited; they did not exhaustively consider all relevant comparable approaches that derive if one would take pattern-type independence into account and recognize that graph-based approaches may also be used in simpler pattern domains; also, the number of data sets in most publications is limited and usually restricted to one data type. A few recent publications have presented more exhaustive experimental comparisons; [17, 66] compared post-processing pattern based classification with kernels and traditional approaches on a large number of molecular data sets, and obtained mixed results. Similarly, [32] compared exhaustive rule discovery strategies to greedy ones on UCI data sets. These results are necessary steps into gaining a better insight in the true relative merits of the many pattern-based classification strategies.

*Acknowledgments* This work was supported by a Postdoc and a project grant from the Research Foundation—Flanders, project “Principles of Patternset Mining”.

## References

1. Bavani Arunasalam and Sanjay Chawla. Cccs: a top-down associative classifier for imbalanced class distribution. In Eliassi-Rad et al. [20], pages 517–522.
2. Martin Atzmüller and Frank Puppe. Sd-map - a fast algorithm for exhaustive subgroup discovery. In Fürnkranz et al. [23], pages 6–17.
3. James Bailey and Guozhu Dong. Contrast data mining: Methods and applications. In *ICDM Tutorials*, 2007.
4. Stephen D. Bay and Michael J. Pazzani. Detecting change in categorical data: Mining contrast sets. In *KDD*, pages 302–306, 1999.
5. Stephen D. Bay and Michael J. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, 2001.
6. Roberto J. Bayardo Jr. and Rakesh Agrawal. Mining the most interesting rules. In *KDD*, pages 145–154, 1999.

7. Jean-François Boulicaut and Baptiste Jeudy. Mining free itemsets under constraints. In Michel E. Adiba, Christine Collet, and Bipin C. Desai, editors, *IDEAS*, pages 322–329, 2001.
8. Björn Bringmann and Albrecht Zimmermann. Tree<sup>2</sup> - decision trees for tree structured data. In Alípio Jorge, Luís Torgo, Pavel Brazdil, Rui Camacho, and João Gama, editors, *PKDD*, volume 3721 of *Lecture Notes in Computer Science*, pages 46–58. Springer, 2005.
9. Björn Bringmann and Albrecht Zimmermann. The chosen few: On identifying valuable patterns. In Ramakrishnan and Zaiane [55], pages 63–72.
10. Björn Bringmann and Albrecht Zimmermann. Aggregated subset mining. In *PAKDD*, 2009.
11. Björn Bringmann, Albrecht Zimmermann, Luc De Raedt, and Siegfried Nijssen. Don't be afraid of simpler patterns. In Fürnkranz et al. [23], pages 55–66.
12. Toon Calders and Bart Goethals. Mining all non-derivable frequent itemsets. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *PKDD*, pages 74–85. Springer, 2002.
13. Loïc Cerf, Dominique Gay, Nazha Selmaoui, and Jean-François Boulicaut. A parameter-free associative classification method. In Il-Yeol Song, Johann Eder, and Tho Manh Nguyen, editors, *DaWaK*, volume 5182 of *Lecture Notes in Computer Science*, pages 293–304. Springer, 2008.
14. Hong Cheng, Jiawei Han, Xifeng Yan, and Philip S. Yu. Integration of classification and pattern mining: A discriminative and frequent pattern-based approach. In *ICDM Tutorials*, 2008.
15. Hong Cheng, Xifeng Yan, Jiawei Han, and Chih-Wei Hsu. Discriminative frequent pattern analysis for effective classification. In *ICDE*, pages 716–725, 2007.
16. Hong Cheng, Xifeng Yan, Jiawei Han, and Philip S. Yu. Direct discriminative pattern mining for effective classification. In *ICDE*, pages 169–178, 2008.
17. Mukund Deshpande, Michihiro Kuramochi, Nikil Wale, and George Karypis. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Trans. Knowl. Data Eng.*, 17(8):1036–1050, 2005.
18. Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: Discovering trends and differences. In *KDD*, pages 43–52, 1999.
19. Guozhu Dong, Xiuzhen Zhang, Limsoon Wong, and Jinyan Li. Caep: Classification by aggregating emerging patterns. In Setsuo Arikawa and Koichi Furukawa, editors, *Discovery Science*, volume 1721 of *Lecture Notes in Computer Science*, pages 30–42. Springer, 1999.
20. Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos, editors. *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*. ACM, 2006.
21. Hongjian Fan and Kotagiri Ramamohanarao. Fast discovery and the generalization of strong jumping emerging patterns for building compact and accurate classifiers. *IEEE Trans. Knowl. Data Eng.*, 18(6):721–737, 2006.
22. Johannes Fürnkranz and Peter A. Flach. ROC 'n' rule learning – towards a better understanding of covering algorithms. *Machine Learning*, 58(1):39–77, 2005.
23. Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors. *Knowledge Discovery in Databases: PKDD 2006, 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin, Germany, September 18-22, 2006, Proceedings*. Springer, 2006.

24. Gemma C. Garriga, Hannes Heikinheimo, and Jouni K. Seppänen. Cross-mining binary and numerical attributes. In Ramakrishnan and Zaiane [55], pages 481–486.
25. Gemma C. Garriga, Petra Kralj, and Nada Lavrac. Closed sets for labeled data. In Fürnkranz et al. [23], pages 163–174.
26. Warodom Geamsakul, Takashi Matsuda, Tetsuya Yoshida, Hiroshi Motoda, and Takashi Washio. Performance evaluation of decision tree graph-based induction. In Gunter Grieser, Yuzuru Tanaka, and Akihiro Yamamoto, editors, *Discovery Science*, pages 128–140, Sapporo, Japan, October 2003. Springer.
27. Henrik Grosskreutz, Stefan Rüping, and Stefan Wrobel. Tight optimistic estimates for fast subgroup discovery. In *ECML/PKDD (1)*, pages 440–456, 2008.
28. Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.*, 8(1):53–87, 2004.
29. Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, Jérémy Besson, and Mohammed Javeed Zaki. Origami: Mining representative orthogonal graph patterns. In Ramakrishnan and Zaiane [55], pages 153–162.
30. Kosuke Hashimoto, Ichigaku Takigawa, Motoki Shiga, Minoru Kanehisa, and Hiroshi Mamitsuka. Mining significant tree patterns in carbohydrate sugar chains. *Bioinformatics*, 24(16):i167–i173, 2008.
31. Masahiro Hirao, Hiromasa Hoshino, Ayumi Shinohara, Masayuki Takeda, and Set-suo Arikawa. A practical algorithm to find the best subsequence patterns. *Theor. Comput. Sci.*, 292(2):465–479, 2003.
32. Frederik Janssen and Johannes Fürnkranz. A re-evaluation of the over-searching phenomenon in inductive rule learning. In *SDM*, pages 329–340. SIAM, 2009.
33. Branko Kavsek and Nada Lavrac. Apriori-sd: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, 20(7):543–583, 2006.
34. W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*. MIT, 1996.
35. A. Knobbe, B. Crémilleux, J. Fürnkranz, and M. Scholz. From local patterns to global models: the LeGo approach to data mining. In J. Fürnkranz and A. Knobbe, editors, *LeGo*, 2008.
36. Arno J. Knobbe and Eric K. Y. Ho. Maximally informative k-itemsets and their efficient discovery. In Eliassi-Rad et al. [20], pages 237–244.
37. Arno J. Knobbe and Eric K. Y. Ho. Pattern teams. In Fürnkranz et al. [23], pages 577–584.
38. Petra Kralj Novak, Nada Lavrač, and Geoffrey I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 10:377–403, Feb 2009. <http://www.jmlr.org/papers/volume10/kralj-novak09a/kralj-novak09a.pdf>.
39. Stefan Kramer and Luc De Raedt. Feature construction with version spaces for biochemical applications. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *ICML*, pages 258–265. Morgan Kaufmann, 2001.
40. Jinyan Li, Guozhu Dong, and Kotagiri Ramamohanarao. Instance-based classification by emerging patterns. In Djamel A. Zighed, Henryk Jan Komorowski, and Jan M. Zytkow, editors, *PKDD*, volume 1910 of *Lecture Notes in Computer Science*, pages 191–200. Springer, 2000.
41. Jinyan Li, Guozhu Dong, and Kotagiri Ramamohanarao. Making use of the most expressive jumping emerging patterns for classification. *Knowl. Inf. Syst.*, 3(2):131–145, 2001.

42. Wenmin Li, Jiawei Han, and Jian Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In Nick Cercone, Tsau Young Lin, and Xindong Wu, editors, *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 369–376, San José, California, USA, 2001. IEEE Computer Society.
43. Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In Rakesh Agrawal, Paul E. Stolorz, and Gregory Piatetsky-Shapiro, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 80–86, New York City, New York, USA, August 1998. AAAI Press.
44. Elsa Loekito and James Bailey. Fast mining of high dimensional expressive contrast patterns using zero-suppressed binary decision diagrams. In Eliassi-Rad et al. [20], pages 307–316.
45. Pauli Miettinen. The boolean column and column-row matrix decompositions. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *ECML/PKDD (1)*, volume 5211 of *Lecture Notes in Computer Science*, page 17. Springer, 2008.
46. Pauli Miettinen, Taneli Mielikäinen, Aristides Gionis, Gautam Das, and Heikki Mannila. The discrete basis problem. *IEEE Trans. Knowl. Data Eng.*, 20(10):1348–1362, 2008.
47. Yasuhiko Morimoto, Takeshi Fukuda, Hirofumi Matsuzawa, Takeshi Tokuyama, and Kunikazu Yoda. Algorithms for mining association rules for binary segmentations of huge categorical databases. In *VLDB*, pages 380–391, 1998.
48. Shinichi Morishita and Jun Sese. Traversing itemset lattice with statistical metric pruning. In *PODS*, pages 226–236, 2000.
49. Siegfried Nijssen and Elisa Fromont. Mining optimal decision trees from itemset lattices. In *Proceeding of the 13th ACM SIGKDD international conference on Knowledge discovery in data mining (KDD'07)*, pages 530–539, 2007.
50. Siegfried Nijssen, Tias Guns, and Luc De Raedt. Correlated itemset mining in roc space: a constraint programming approach. In John F. Elder IV, Françoise Fogelman-Soulié, Peter Flach, and Mohammed Zaki, editors, *KDD*, pages 647–656. ACM, 2009.
51. Siegfried Nijssen and Joost N. Kok. Multi-class correlated pattern mining. In *KDID*, pages 165–187, 2005.
52. Sebastian Nowozin, Koji Tsuda, Takeaki Uno, Taku Kudo, and Gökhan H. Bakir. Weighted substructure mining for image analysis. In *MLG*, 2007.
53. Sebastian Nowozin, Koji Tsuda, Takeaki Uno, Taku Kudo, and Gökhan H. Bakir. Weighted substructure mining for image analysis. In *CVPR*, 2007.
54. Luc De Raedt and Albrecht Zimmermann. Constraint-based pattern set mining. In *SDM*. SIAM, 2007.
55. Naren Ramakrishnan and Osmar Zaiane, editors. *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*. IEEE Computer Society, 2007.
56. Kotagiri Ramamohanarao and Hongjian Fan. Patterns based classifiers. In *World Wide Web*, pages 71–83, 2007.
57. Ulrich Rückert and Stefan Kramer. Optimizing feature sets for structured data. In Joost N. Kok, Jacek Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron, editors, *ECML*, volume 4701 of *Lecture Notes in Computer Science*, pages 716–723. Springer, 2007.
58. Hiroto Saigo, Nicole Krämer, and Koji Tsuda. Partial least squares regression for graph mining. In *KDD*, pages 578–586, 2008.

59. Arno Siebes, Jilles Vreeken, and Matthijs van Leeuwen. Item sets that compress. In Joydeep Ghosh, Diane Lambert, David B. Skillicorn, and Jaideep Srivastava, editors, *SDM*. SIAM, 2006.
60. Rafik Taouil, Nicolas Pasquier, Yves Bastide, and Lotfi Lakhal. Mining bases for association rules using closed sets. In *ICDE*, page 307, 2000.
61. Marisa Thoma, Hong Cheng, Arthur Gretton, Jiawei Han, Hans-Peter Kriegel, Alexander J. Smola, Le Song, Philip S. Yu, Xifeng Yan, and Karsten M. Borgwardt. Near-optimal supervised feature selection among frequent subgraphs. In *SDM*, 2009.
62. Koji Tsuda. Entire regularization paths for graph data. In Zoubin Ghahramani, editor, *ICML*, volume 227 of *ACM International Conference Proceeding Series*, pages 919–926. ACM, 2007.
63. Matthijs van Leeuwen, Jilles Vreeken, and Arno Siebes. Compression picks item sets that matter. In Fürnkranz et al. [23], pages 585–592.
64. Adriano Veloso, Wagner Meira Jr., Marcos André Gonçalves, and Mohammed Javeed Zaki. Multi-label lazy associative classification. In Joost N. Kok, Jacek Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron, editors, *PKDD*, volume 4702 of *Lecture Notes in Computer Science*, pages 605–612. Springer, 2007.
65. Adriano Veloso, Wagner Meira Jr., and Mohammed J. Zaki. Lazy associative classification. In *ICDM*, pages 645–654. IEEE Computer Society, 2006.
66. Nikil Wale, Ian A. Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowl. Inf. Syst.*, 14(3):347–375, 2008.
67. Jianyong Wang and George Karypis. Harmony: Efficiently mining the best rules for classification. In *SDM*, 2005.
68. Lusheng Wang, Hao Zhao, Guozhu Dong, and Jianping Li. On the complexity of finding emerging patterns. *Theor. Comput. Sci.*, 335(1):15–27, 2005.
69. Geoffrey I. Webb. Opus: An efficient admissible algorithm for unordered search. *J. Artif. Intell. Res. (JAIR)*, 3:431–465, 1995.
70. Geoffrey I. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.
71. S. Wrobel. An algorithm for multi-relational discovery of subgroups. In J. Komorowski and J. Zytkow, editors, *Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD '97)*, pages 78 – 87. SV, 1997.
72. Dong Xin, Hong Cheng, Xifeng Yan, and Jiawei Han. Extracting redundancy-aware top-k patterns. In Eliassi-Rad et al. [20], pages 444–453.
73. Xifeng Yan, Hong Cheng, Jiawei Han, and Philip S. Yu. Mining significant graph patterns by leap search. In *SIGMOD Conference*, pages 433–444, 2008.
74. Xiaoxin Yin and Jiawei Han. Cpar: Classification based on predictive association rules. In Daniel Barbará and Chandrika Kamath, editors, *SDM*. SIAM, 2003.
75. Mohammed Javeed Zaki and Charu C. Aggarwal. XRules: an effective structural classifier for XML data. In Lise Getoor, Ted E. Senator, Pedro Domingos, and Christos Faloutsos, editors, *KDD*, pages 316–325, Washington, DC, USA, August 2003. ACM.
76. Albrecht Zimmermann and Björn Bringmann. CTC - correlating tree patterns for classification. In *ICDM*, pages 833–836, 2005.