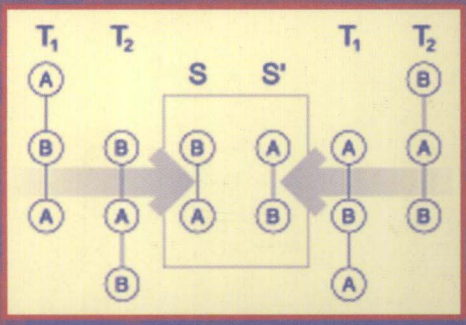
State-of-the-Art Survey

LNAI 3848

Jean-François Boulicaut
Luc De Raedt
Heikki Mannila (Eds.)

Constraint-Based Mining and Inductive Databases

**European Workshop on Inductive Databases and Constraint Based Mining
Hinterzarten, Germany, March 2004, Revised Selected Papers**



Springer

Jean-François Boulicaut Luc De Raedt
Heikki Mannila (Eds.)

Constraint-Based Mining and Inductive Databases

European Workshop on Inductive Databases and Constraint Based Mining
Hinterzarten, Germany, March 11-13, 2004
Revised Selected Papers

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Jean-François Boulicaut
INSA Lyon, LIRIS CNRS UMR 5205
69621 Villeurbanne, France
E-mail: Jean-Francois.Boulicaut@insa-lyon.fr

Luc De Raedt
Albert-Ludwigs-University, Institute for Computer Science
Georges-Köhler-Allee 79, 79110 Freiburg, Germany
E-mail: deraedt@informatik.uni-freiburg.de

Heikki Mannila
HIIT Basic Research Unit, University of Helsinki
and Helsinki University of Technology
P.O. Box 68, 00014 Helsinki, Finland
E-mail: Heikki.Mannila@cs.helsinki.fi

Library of Congress Control Number: 2005938512

CR Subject Classification (1998): I.2, H.2.8, H.2-3, D.3.3, F.1

ISSN	0302-9743
ISBN-10	3-540-31331-1 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-31331-1 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11615576 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3848

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Preface

The interconnected ideas of inductive databases and constraint-based mining are appealing and have the potential to radically change the theory and practice of data mining and knowledge discovery. Today, knowledge discovery is a very time-consuming and ad-hoc process, in which the analyst has to craft together a solution in a rather procedural manner. The ultimate goal of the inductive database framework is to develop an inductive query language, which would support the overall knowledge discovery process. Inductive queries specify constraints over patterns and models in a declarative way. Within this framework, the user then poses queries, which an inductive database management system has to answer, and knowledge discovery becomes an interactive querying process.

This book reports on the results of the European IST project cINQ (consortium on knowledge discovery by Inductive Queries) and its final workshop entitled "Inductive Databases and Constraint-Based Mining" organized in the Black Forest (Hinterzarten, Germany, March 11-14, 2004). The cINQ consortium consisted of INSA Lyon (France, coordinator: Jean-François Boulicaut), Università degli Studi di Torino (Italy, Rosa Meo and Marco Botta), the Politecnico di Milano (Italy, Pier-Luca Lanzi and Stefano Ceri), the Albert-Ludwigs-Universitaet Freiburg (Germany, Luc De Raedt), the Nokia Research Center in Helsinki (Finland, Mika Klemettinen and Heikki Mannila), and the Jozef Stefan Institute (Slovenia, Sašo Džeroski).

The workshop was attended by about 50 researchers, who presented their latest results in inductive querying and constraint-based data mining and also identified future directions. These results are presented in this book and provide a state-of-the-art overview of this newly emerging field lying at the intersection of data mining and database research. Even though we are still far away from inductive database management systems, a lot of progress has been made over the past few years, especially in constraint-based mining for local patterns (e.g., sets, sequential patterns, trees, graphs and rules), and in identifying some new primitives for data mining. Nevertheless, various important questions still remain open, such as the integration of query languages with databases and the fundamentals for inductive querying on global patterns.

The papers in this book can be categorized as follows (they are ordered in the book according to the name of the first author):

Keynote speakers: The chapter by Roberto J. Bayardo is an interesting position paper on various issues for constraint-based pattern mining. Johannes Gehrke and his co-authors provide a nice theoretical framework for optimizing constraint-based mining in difficult cases, typically when monotonicity properties are missing. Finally, Mohammed J. Zaki and his co-authors give a pragmatic view on the future of data mining software.

- *The Hows, Whys, and Whens of Constraints in Itemset and Rule Discovery* by Roberto J. Bayardo

- *How to Quickly Find a Witness* by Daniel Kifer, Johannes Gehrke, Cristian Bucila, and Walker White
- *Generic Pattern Mining via Data Mining Template Library* by Mohammed J. Zaki, Nilanjana De, Feng Gao, Paolo Palmerini, Nagender Parimi, Jeevan Pathuri, Benjarath Phoophakdee, and Joe Urban

Foundations: Several chapters address conceptual issues related to the inductive database framework, e.g., querying primitives, condensed representations, multiple uses of frequent sets, and the optimization of sequences of inductive queries:

- *A Relational Query Primitive for Constraint-Based Pattern Mining* by Francesco Bonchi, Fosca Giannotti and Dino Pedreschi.
- *A Survey on Condensed Representations for Frequent Sets* by Toon Calders, Christophe Rigotti and Jean-François Boulicaut
- *Boolean Formulas and Frequent Sets* by Jouni K. Seppänen and Heikki Mannila
- *Computation of Mining Queries: An Algebraic Approach* by Cheikh Talibouya Diop, Arnaud Giacometti, Dominique Laurent, and Nicolas Spyratos

Optimizing inductive queries on local patterns: Several chapters concern local pattern discovery by means of constraint-based mining techniques. A variety of pattern domains are considered such as trees, graphs, subgroups, inclusion dependencies, and association rules:

- *To See the Wood for the Trees: Mining Frequent Tree Patterns* by Björn Bringmann
- *Mining Constrained Graphs: The Case of Workflow Systems* by Gianluigi Greco, Antonella Guzzo, Giuseppe Manco, Luigi Pontieri, and Domenico Saccà
- *Relevancy in Constraint-Based Subgroup Discovery* by Nada Lavrač, and Dragan Gamberger
- *Adaptive Strategies for Mining the Positive Border of Interesting Patterns: Application to Inclusion Dependencies in Databases* by Fabien De Marchi, Frédéric Flouvat, and Jean-Marc Petit
- *A Novel Incremental Approach to Association Rules Mining in Inductive Databases* by Rosa Meo, Marco Botta, Roberto Esposito, and Arianna Gallo

Optimizing inductive queries on global patterns: Less research has been devoted to constraint-based mining of global patterns or models like clusters or classifiers. Important results in this direction are presented:

- *Inductive Queries on Polynomial Equations* by Sašo Džeroski, Ljupčo Todorovski, and Peter Ljubič
- *CrossMine: Efficient Classification Across Multiple Database Relations* by Xiaoxin Yin, Jiawei Han, Jiong Yang, and Philip S. Yu
- *Inductive Querying for Discovering Subgroups and Clusters* by Albrecht Zimmermann and Luc De Raedt

Applications: It is of course important to look at concrete applications of inductive querying techniques. Three chapters report on this:

- *Remarks on the Industrial Application of Inductive Database Technologies* by Kimmo Hätönen, Mika Klemettinen, and Markus Miettinen
- *Employing Inductive Databases in Concrete Applications* by Rosa Meo, Pier Luca Lanzi, Maristella Matera, Danilo Careggio, and Roberto Esposito
- *Contribution to Gene Expression Data Analysis by Means of Set Pattern Mining* by Ruggero G. Pensa, Jérémy Besson, Céline Robardet, and Jean-François Boulicaut

The editors would like to thank the EU (FET arm of the IST programme) for supporting the cInQ project as well as the Hinterzarten workshop, the partners in the cInQ consortium, and the participants in the workshop, especially our keynote speakers: Roberto J. Bayardo, Johannes Gehrke, and Mohammed J. Zaki. We hope that the readers will enjoy reading this book as much as we enjoyed the process of producing it.

September 2005

Jean-François Boulicaut
Luc De Raedt
Heikki Mannila

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3848: J.-F. Boulicaut, L. De Raedt, H. Mannila (Eds.), Constraint-Based Mining and Inductive Databases. X, 401 pages. 2005.

Vol. 3847: K.P. Jantke, A. Lunzer, N. Spyratos, Y. Tanaka (Eds.), Federation over the Web. X, 215 pages. 2005.

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005.

Vol. 3817: M. Faundez-Zanuy, L. Janer, A. Esposito, A. Satue-Vilar, J. Roure, V. Espinosa-Guro (Eds.), Nonlinear Analyses and Algorithms for Speech Processing. XII, 380 pages. 2005.

Vol. 3814: M. Maybury, O. Stock, W. Wahlster (Eds.), Intelligent Technologies for Interactive Entertainment. XV, 342 pages. 2005.

Vol. 3809: S. Zhang, R. Jarvis (Eds.), AI 2005: Advances in Artificial Intelligence. XXVII, 1344 pages. 2005.

Vol. 3808: C. Bento, A. Cardoso, G. Dias (Eds.), Progress in Artificial Intelligence. XVIII, 704 pages. 2005.

Vol. 3802: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part II. XLII, 1166 pages. 2005.

Vol. 3801: Y. Hao, J. Liu, Y.-P. Wang, Y.-m. Cheung, H. Yin, L. Jiao, J. Ma, Y.-C. Jiao (Eds.), Computational Intelligence and Security, Part I. XLI, 1122 pages. 2005.

Vol. 3789: A. Gelbukh, Á. de Albornoz, H. Terashima-Marín (Eds.), MICAI 2005: Advances in Artificial Intelligence. XXVI, 1198 pages. 2005.

Vol. 3782: K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick, T. Roth-Berghofer (Eds.), Professional Knowledge Management. XXIII, 739 pages. 2005.

Vol. 3735: A. Hoffmann, H. Motoda, T. Scheffer (Eds.), Discovery Science. XVI, 400 pages. 2005.

Vol. 3734: S. Jain, H.U. Simon, E. Tomita (Eds.), Algorithmic Learning Theory. XII, 490 pages. 2005.

Vol. 3721: A.M. Jorge, L. Torgo, P.B. Brazdil, R. Camacho, J. Gama (Eds.), Knowledge Discovery in Databases: PKDD 2005. XXIII, 719 pages. 2005.

Vol. 3720: J. Gama, R. Camacho, P.B. Brazdil, A.M. Jorge, L. Torgo (Eds.), Machine Learning: ECML 2005. XXIII, 769 pages. 2005.

Vol. 3717: B. Gramlich (Ed.), Frontiers of Combining Systems. X, 321 pages. 2005.

Vol. 3702: B. Beckert (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods. XIII, 343 pages. 2005.

Vol. 3698: U. Furbach (Ed.), KI 2005: Advances in Artificial Intelligence. XIII, 409 pages. 2005.

Vol. 3690: M. Pěchouček, P. Petta, L.Z. Varga (Eds.), Multi-Agent Systems and Applications IV. XVII, 667 pages. 2005.

Vol. 3684: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part IV. LXXIX, 933 pages. 2005.

Vol. 3683: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LXXX, 1397 pages. 2005.

Vol. 3682: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LXXIX, 1371 pages. 2005.

Vol. 3681: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LXXX, 1319 pages. 2005.

Vol. 3673: S. Bandini, S. Manzoni (Eds.), AI*IA 2005: Advances in Artificial Intelligence. XIV, 614 pages. 2005.

Vol. 3662: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Logic Programming and Nonmonotonic Reasoning. XIII, 454 pages. 2005.

Vol. 3661: T. Panayiotopoulos, J. Gratch, R.S. Aylett, D. Ballin, P. Olivier, T. Rist (Eds.), Intelligent Virtual Agents. XIII, 506 pages. 2005.

Vol. 3658: V. Matoušek, P. Mautner, T. Pavelka (Eds.), Text, Speech and Dialogue. XV, 460 pages. 2005.

Vol. 3651: R. Dale, K.-F. Wong, J. Su, O.Y. Kwong (Eds.), Natural Language Processing – IJCNLP 2005. XXI, 1031 pages. 2005.

Vol. 3642: D. Ślęzak, J. Yao, J.F. Peters, W. Ziarko, X. Hu (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part II. XXIII, 738 pages. 2005.

Vol. 3641: D. Ślęzak, G. Wang, M. Szczuka, I. Düntsch, Y. Yao (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part I. XXIV, 742 pages. 2005.

Vol. 3635: J.R. Winkler, M. Niranjan, N.D. Lawrence (Eds.), Deterministic and Statistical Methods in Machine Learning. VIII, 341 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005.

Vol. 3630: M.S. Capcarrère, A.A. Freitas, P.J. Bentley, C.G. Johnson, J. Timmis (Eds.), Advances in Artificial Life. XIX, 949 pages. 2005.

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005.

Vol. 3625: S. Kramer, B. Pfahringer (Eds.), Inductive Logic Programming. XIII, 427 pages. 2005.

Vol. 3620: H. Muñoz-Ávila, F. Ricci (Eds.), Case-Based Reasoning Research and Development. XV, 654 pages. 2005.

Vol. 3614: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part II. XLI, 1314 pages. 2005.

Vol. 3613: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part I. XLI, 1334 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005.

Vol. 3601: G. Moro, S. Bergamaschi, K. Aberer (Eds.), Agents and Peer-to-Peer Computing. XII, 245 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005.

Vol. 3593: V. Mařík, R. W. Brennan, M. Pěchouček (Eds.), Holonic and Multi-Agent Systems for Manufacturing. XI, 269 pages. 2005.

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E.T. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005.

Vol. 3577: R. Falcone, S. Barber, J. Sabater-Mir, M.P. Singh (Eds.), Trusting Agents for Trusting Electronic Societies. VIII, 235 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005.

Vol. 3558: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence. XII, 470 pages. 2005.

Vol. 3554: A.K. Dey, B. Kokinov, D.B. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005.

Vol. 3550: T. Eymann, F. Klügl, W. Lamersdorf, M. Klusch, M.N. Huhns (Eds.), Multiagent System Technologies. XI, 246 pages. 2005.

Vol. 3539: K. Morik, J.-F. Boulicaut, A. Siebes (Eds.), Local Pattern Detection. XI, 233 pages. 2005.

Vol. 3538: L. Ardissono, P. Brna, A. Mitrović (Eds.), User Modeling 2005. XVI, 533 pages. 2005.

Vol. 3533: M. Ali, F. Esposito (Eds.), Innovations in Applied Artificial Intelligence. XX, 858 pages. 2005.

Vol. 3528: P.S. Szczepaniak, J. Kacprzyk, A. Niewiadomski (Eds.), Advances in Web Intelligence. XVII, 513 pages. 2005.

Vol. 3518: T.-B. Ho, D. Cheung, H. Liu (Eds.), Advances in Knowledge Discovery and Data Mining. XXI, 864 pages. 2005.

Vol. 3508: P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, M. Winikoff (Eds.), Agent-Oriented Information Systems II. X, 227 pages. 2005.

Vol. 3505: V. Gorodetsky, J. Liu, V.A. Skormin (Eds.), Autonomous Intelligent Systems: Agents and Data Mining. XIII, 303 pages. 2005.

Vol. 3501: B. Kégl, G. Lapalme (Eds.), Advances in Artificial Intelligence. XV, 458 pages. 2005.

Vol. 3492: P. Blache, E.P. Stabler, J.V. Busquets, R. Moot (Eds.), Logical Aspects of Computational Linguistics. X, 363 pages. 2005.

Vol. 3490: L. Bolc, Z. Michalewicz, T. Nishida (Eds.), Intelligent Media Technology for Communicative Intelligence. X, 259 pages. 2005.

Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Raś, S. Tsumoto (Eds.), Foundations of Intelligent Systems. XIII, 700 pages. 2005.

Vol. 3487: J.A. Leite, P. Torroni (Eds.), Computational Logic in Multi-Agent Systems. XII, 281 pages. 2005.

Vol. 3476: J.A. Leite, A. Omicini, P. Torroni, P. Yolum (Eds.), Declarative Agent Languages and Technologies II. XII, 289 pages. 2005.

Vol. 3464: S.A. Brueckner, G.D.M. Serugendo, A. Karageorgos, R. Nagpal (Eds.), Engineering Self-Organising Systems. XIII, 299 pages. 2005.

Vol. 3452: F. Baader, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XI, 562 pages. 2005.

Vol. 3451: M.-P. Gleizes, A. Omicini, F. Zambonelli (Eds.), Engineering Societies in the Agents World V. XIII, 349 pages. 2005.

Vol. 3446: T. Ishida, L. Gasser, H. Nakashima (Eds.), Massively Multi-Agent Systems I. XI, 349 pages. 2005.

Vol. 3445: G. Chollet, A. Esposito, M. Faúndez-Zanuy, M. Marinaro (Eds.), Nonlinear Speech Modeling and Applications. XIII, 433 pages. 2005.

Vol. 3438: H. Christiansen, P.R. Skadhauge, J. Villadsen (Eds.), Constraint Solving and Language Processing. VIII, 205 pages. 2005.

Vol. 3430: S. Tsumoto, T. Yamaguchi, M. Numao, H. Motoda (Eds.), Active Mining. XII, 349 pages. 2005.

Vol. 3419: B.V. Faltings, A. Petcu, F. Fages, F. Rossi (Eds.), Recent Advances in Constraints. X, 217 pages. 2005.

Vol. 3416: M.H. Böhlen, J. Gamper, W. Polasek, M.A. Wimmer (Eds.), E-Government: Towards Electronic Democracy. XIII, 311 pages. 2005.

Vol. 3415: P. Davidsson, B. Logan, K. Takadama (Eds.), Multi-Agent and Multi-Agent-Based Simulation. X, 265 pages. 2005.

Vol. 3413: K. Fischer, M. Florian, T. Malsch (Eds.), Socionics. X, 315 pages. 2005.

Vol. 3403: B. Ganter, R. Godin (Eds.), Formal Concept Analysis. XI, 419 pages. 2005.

Vol. 3398: D.-K. Baik (Ed.), Systems Modeling and Simulation: Theory and Applications. XIV, 733 pages. 2005.

Vol. 3397: T.G. Kim (Ed.), Artificial Intelligence and Simulation. XV, 711 pages. 2005.

Vol. 3396: R.M. van Eijk, M.-P. Huget, F.P. M. Dignum (Eds.), Agent Communication. X, 261 pages. 2005.

Table of Contents

The Hows, Whys, and Whens of Constraints in Itemset and Rule Discovery

Roberto J. Bayardo

IBM Almaden Research Center
bayardo@alum.mit.edu
http://www.almaden.ibm.com/cs/people/bayardo/

Abstract. Many researchers in our community (this author included) regularly emphasize the role constraints play in improving performance of data-mining algorithms. This emphasis has led to remarkable progress – current algorithms allow an incredibly rich and varied set of hidden patterns to be efficiently elicited from massive datasets, even under the burden of NP-hard problem definitions and disk-resident or distributed data. But this progress has come at a cost. In our single-minded drive towards maximum performance, we have often neglected and in fact hindered the important role of discovery in the knowledge discovery and data-mining (KDD) process. In this paper, I propose various strategies for applying constraints within algorithms for itemset and rule mining in order to escape this pitfall[1].

1 Introduction

Constraint-based pattern mining is the process of identifying all patterns in a given dataset that satisfy the specified constraints. There are many types of patterns we may wish to explore, depending on the data or its expected use. To name only a few, we have itemsets, sequences, episodes, substrings, rules, trees, cliques, and so on. The important aspect of constraint-based mining is not so much the specific patterns being identified, but the fact that we would like to identify *all* of them subject to the given constraints. This task of *constraint-based mining* is in contrast to *heuristic* pattern mining which attempts only to identify patterns which are likely (but not guaranteed) to be good according to certain criteria. A third task which I will touch upon only briefly, *optimization-based* pattern mining, attempts to identify only those patterns that are guaranteed to be (among the k-) best according to certain metrics.

While many may assign constraint-based mining a high face value solely from plethora of research on the topic, it is illustrative to take a step back and contemplate ***why*** it is a task worthy of our interest. Indeed, long before the "association rule" was a household name, heuristic pattern miners were proving extremely

[1] My use of the informal "I" rather than the typical "we" is to emphasize this paper is a personal position statement, along with a view of existing research in light of my position.

J.-F. Boulicaut et al. (Eds.): Constraint-Based Mining, LNAI 3848, pp. 1–13, 2005.

useful in machine learning circles. In fact, heuristic rule miners, which include decision tree ("divide and conquer") and covering ("separate and conquer") algorithms, remain essential components in the analyst's toolbox. I witnessed a growing interest in constraint-based mining once heuristic machine learning approaches gained reasonably widespread use in practice. The white-box nature of decision tree and other rule-based models were being used directly for end-user understanding of the data, even though they were not specifically intended for that purpose[2]. Use of these rule-based models for understanding led to questions such as the following:

- Do these rules capture and convey the "essence" of the relationship(s) in my data?
- Are there better rules (and who gets to define better)?

Note that each of these questions is open to some amount of subjective interpretation. But this is the point: the analyst is typically involved in *knowledge discovery* in which subjective and difficult to formalize notions of "goodness" are guiding the process, not simply data mining in which an algorithm follows a deterministic procedure to extract patterns that may or (more often) may not be of interest. Provided that constraints are used sensibly (and what "sensibly" means is the subject of this paper), constraint-based mining fosters discovery by providing the analyst with a broad result set capable of concretely answering a far wider set of questions than one that is heuristically determined.

A theme of this paper is that there are different phases of the knowledge discovery process in which we can exploit constraints, and the specific use of constraints should be dependent on ***when*** (in what phase) we are using them. During the mining phase, I argue that constraints should be *discovery preserving*. That is, they should filter out only those results that are *highly unlikely* to ever be of interest to the analyst. This admittedly informal notion of preserving discovery is in stark contrast to other proposals that envision query languages for constraint-based mining in which every imaginable constraint is enforced directly by the mining phase. The problem with this alternate view is simply that the analyst rarely knows the specific results of interest a priori (no pun intended). Constraints should therefore be used during the mining phase primarily for performance tractability. Discovering the precise results of interest is best left for post-processing of the mining results through interactive interfaces involving visualization, browsing, and ranking.

Recall that optimization-based pattern discovery forms an interesting middle-ground between the heuristic and constraint-based approaches: unlike heuristic approaches, it provides guarantees on result quality. Unlike constraint-based approaches, it provides these guarantees without requiring the extraction of all patterns matching the constraints, the number of which can be enormous. While these are desirable attributes, once again we are confronted with the question of what makes one rule better than the other. Optimization-based approaches allow

[2] It is therefore ironic that association rule miners are now commonly used in building general classification models, even though originally this was not their intended use!

no ambiguity on the part of the analyst since the ranking function is part of the input, if not hard-coded into the algorithm itself. Should an optimization-based approach be required (for example it is possible the pattern space is simply too large for constraints alone), I argue it is desirable for the approach to provide some ability to select and adjust the ranking criteria post-mining [6]. It is tempting to view an optimization criteria as itself just another constraint to be enforced by a constraint-based miner. Viewed as such, an optimization criteria is actually a constraint on the set of patterns rather than a constraint on the properties of the individual patterns. I believe this distinction is important enough to justify treating optimization-based approaches as separate from constraint-based ones.

As researchers, once we are convinced why something is useful, we become obsessed with ***how*** we can achieve it. And with constraint-based mining, the how part is particularly interesting due to huge computational challenges. Many constraint-based mining tasks can be proven NP-hard through reductions from problems such as constraint satisfaction, hitting set, prime implicant, and so on. Worse, the datasets involved often attain volumes beyond which standard data management strategies can efficiently cope. Then there is the issue of ensuring the results of our algorithms have statistical merit. This combination of search, data management, and statistical issues has provided ample research fodder for our community.

I cannot hope to even begin to address all interesting aspects of the hows in constraint-based mining in this short paper, but I will discuss some (often neglected) issues that I feel fit with in the context of discovery preservation. While much of what remains to be discussed applies to pattern mining in general, for concreteness sake, I focus in particular on itemsets and association rules. An itemset is simply a set of values appearing in a given dataset. An association rule is itself an itemset along with additional information specifying the division of items into antecedent and consequent subsets. The seminal work on association rule mining produced algorithms employing two distinct phases: (1) mine the frequent itemsets from the data, (2) output the rules of interest from them. While this two-phase approach was for the most part an operational detail of the mining algorithm, researchers (again, this author included) have been eager to build on only the first phase as if itemsets themselves are the output desired by the end user. I am quick to agree that itemsets are indeed *sometimes* the artifact of interest in data-mining. But that said, I believe, by and large, that the desired outcome of mining is more often *rules* since they express easy to interpret relationships between dataset elements that itemsets alone do not.

Luckily, many itemset constraints are themselves useful rule constraints, thus work in constraint-based itemset mining often has direct applications in constraint-based rule mining. There are, however, many constraints that are specific to rules such as bounds on confidence, lift, and other measures of predictive accuracy, and they have gone virtually ignored outside of result post-processing. To be fair, another reason rule-specific constraints have been ignored is that they do not fall into any of the convenient constraint classes that have been found to be easily enforceable during mining. But the fact is that many of these rule

constraints can be broken down into constituents that do fall into these classes. I will overview previous work in which properties of these constituents have been exploited for effective enforcement during mining *given appropriate structuring of the search.* That said, coming back to my original thesis, we typically would not want to enforce arbitrary rule constraints during mining to avoid hindering discovery. I therefore provide examples of rule constraints that can be regarded as discovery preserving, along with a framework for their enforcement during mining.

2 Constraints in the Discovery Process

It is well-known that knowledge discovery is a multi-phase and iterative process [11]. The data preparation and data-mining stages are often the most costly in terms of compute overhead. Thus, if possible, iteration should be restricted to subsequent phases (such as post-processing) in which it can be performed quickly. In the context of pattern mining, the role of the data-mining algorithm should be to transform the (preprocessed) dataset into a representation that allows for interactive browsing, ranking, and querying. "Interactive" means that the effects of changing a parameter, for example via a graphical control, are almost instantaneous. The following figure depicts this view.

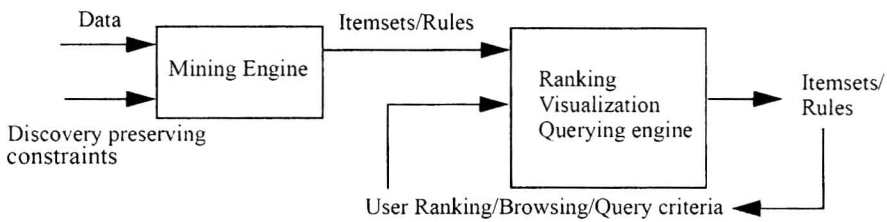


Fig. 1. Idealized View of the Mining Process

In some cases the input dataset may be sufficiently compact and the mining sufficiently trivial to allow the data-mining algorithm to be reapplied in real time to support interactivity. Mining caches can be used to further improve response [15,17], though I have doubts that cache hit rates will be significant enough for this to be of much use in practice.

More often, an intermediate representation is required to satisfy interactive response requirements. In the case of constraint-based rule and itemset mining, this intermediate representation is typically some collection of itemsets with their associated support values. For some datasets it might be possible to precompute the support of all possible itemsets and store them in an indexed database. However, most non-trivial datasets have enough items to make this impractical, as the number of itemsets increases exponentially with the number of items. A solution is to apply constraints to reduce the size of the mining result and the time required to obtain it, preferably without excluding patterns that are

of interest to the analyst. I argue that a good mining engine constraint has the following properties:

1. It is **tweakable**: post-mining, if the constraint is parameterized, the parameter should be adjustable without requiring expensive processing such as scanning or re-mining the original dataset.
2. It **provides efficiency**: applying the constraint should make the algorithm run significantly more efficiently. At this phase we are more concerned with using constraints for achieving tractability, and not necessarily in speeding up mining by a small constant.
3. It **preserves discovery**: the constraint, if it limits the sets of questions the analyst may efficiently pose during post-processing, should eliminate only those questions that are unlikely to be of value.

Properties 1 and 2 allow for the system itself to specify constraints automatically to ensure tractability of the mining run. The user is then able to efficiently adjust the constraints after the fact if necessary.

Property 3 implies that the system has a low probability of excluding patterns that may have otherwise been found interesting by the user. Property 3 is clearly the most subjective. Indeed, any pattern elimination can probably be rationalized as eliminating something useful for *some* purpose. However there are some constraints that do satisfy these properties in most settings. One example is a very low setting of minimum support. (1) Minimum support can be easily adjusted upwards post-mining without going back to the original dataset. One only needs to filter (or ignore) those itemsets whose supports falls below the modified limit. (2) Minimum support has been proven to provide significant boosts in efficiency during mining, even at relatively low settings. (3) Minimum support, provided it can be set low enough, preserves discovery since results with extremely low support are unlikely to be statistically valid.

Is minimum support enough? I feel it is safe to say that for "market-basket" and other sparse datasets, the answer is wholeheartedly yes. In fact, minimum support as exploited by the earliest of association rule miners (such as Apriori) is often entirely sufficient. In figure 2, I reprint with permission two graphs from a recent workshop on frequent itemset mining implementations (FIMI-03 [12]) in which participants submitted implementations for apples-to-apples comparison on a variety of datasets. For the sparse datasets, Borgelt's Apriori implementation outperformed most of the newer algorithms. Only for the very lowest support settings on the bmspos dataset was it outperformed by any significant amount. The point is that for any significantly complex mining task, the transformation and mining phases will be applied offline. Whether an algorithm requires one versus two hours to complete is not a major concern if iteration is relegated to post-processing.

Dense datasets tell a different story. Most tabular datasets with more than a handful of columns are sufficiently dense to render minimum support pruning woefully inadequate. In the FIMI-03 experiments, minimum support was the only constraint considered, and the minimum support levels attainable by any

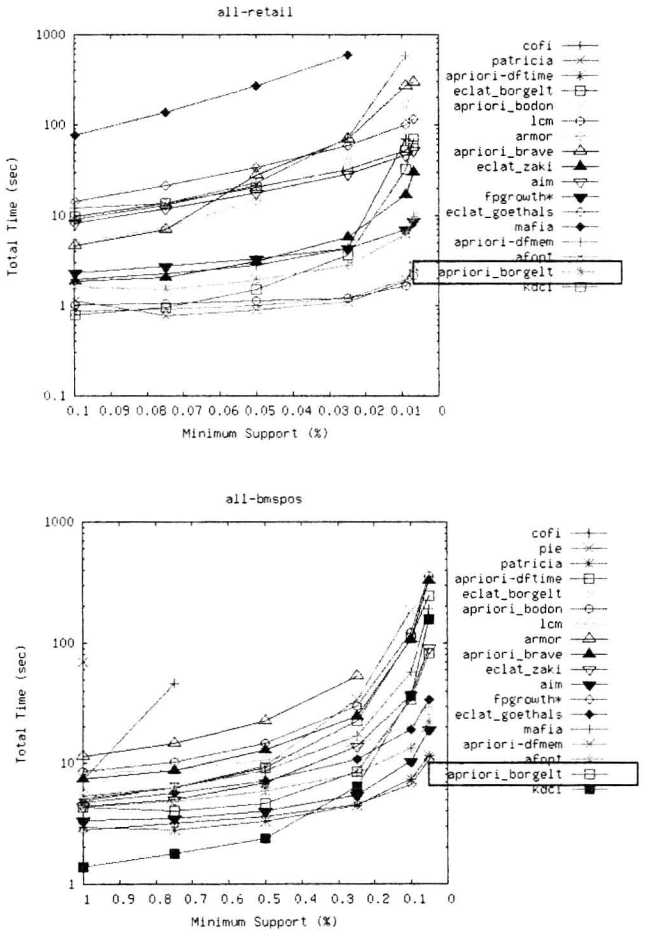


Fig. 2. Performance of the FIMI-03 implementations on sparse datasets

algorithm on the densest datasets were nowhere near what would be necessary to find any reasonably predictive rules [6]. We must therefore ask, what other constraints might we employ? Another good constraint is that the mining artifacts, whether itemsets or rules, be in a sense non-redundant. In the rule mining context, I noted in [8] that when an itemset has support equivalent to that of one of its subsets, it is redundant in the sense that it leads only to rules that are equivalent to existing rules in predictive accuracy and the population covered. It is a simple matter to prune such itemsets in order to avoid excessive counting due to equivalent supports. This idea is the basis of what is now commonly known as freeness and closure [13,19,25] in the context of itemset mining, and also what I called "antecedent maximality" in the context of rule mining [6]. Closure, while