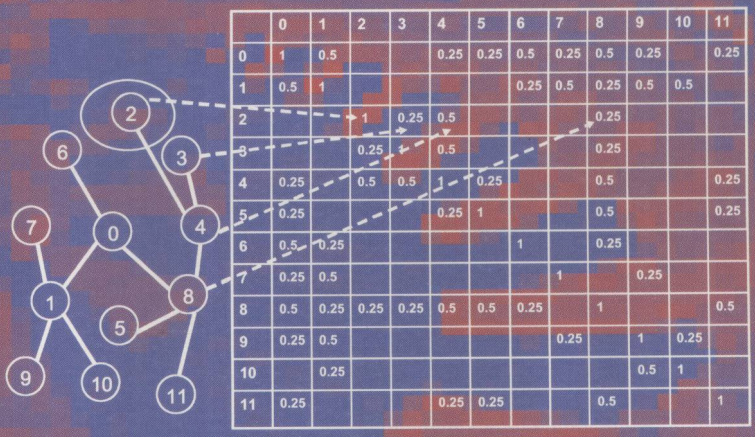
State-of-the-Art Survey

LNAI 3539

Katharina Morik
Jean-François Boulicaut
Arno Siebes (Eds.)

Local Pattern Detection

International Seminar
Dagstuhl Castle, Germany, April 2004
Revised Selected Papers



Springer

Katharina Morik Jean-François Boulicaut
Arno Siebes (Eds.)

Local Pattern Detection

International Seminar
Dagstuhl Castle, Germany, April 12-16, 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

Katharina Morik
University of Dortmund, Computer Science Department, LS VIII
44221 Dortmund, Germany
E-mail: morik@ls8.cs.uni-dortmund.de

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

Arno Siebes
Utrecht University
Department of Information and Computing Sciences
PO Box 80.089, 3508TB Utrecht, The Netherlands
E-mail: arno.siebes@cs.uu.nl

Library of Congress Control Number: 2005929338

CR Subject Classification (1998): I.2, H.2.8, F.2.2, E.5, G.3, H.3

ISSN 0302-9743
ISBN-10 3-540-26543-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-26543-6 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Boller Mediendesign
Printed on acid-free paper SPIN: 11504245 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3539

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Preface

Introduction

The dramatic increase in available computer storage capacity over the last 10 years has led to the creation of very large databases of scientific and commercial information. The need to analyze these masses of data has led to the evolution of the new field knowledge discovery in databases (KDD) at the intersection of machine learning, statistics and database technology. Being interdisciplinary by nature, the field offers the opportunity to combine the expertise of different fields into a common objective. Moreover, within each field diverse methods have been developed and justified with respect to different quality criteria. We have to investigate how these methods can contribute to solving the problem of KDD.

Traditionally, KDD was seeking to find global models for the data that explain most of the instances of the database and describe the general structure of the data. Examples are statistical time series models, cluster models, logic programs with high coverage or classification models like decision trees or linear decision functions. In practice, though, the use of these models often is very limited, because global models tend to find only the obvious patterns in the data, which domain experts already are aware of[1]. What is really of interest to the users are the local patterns that deviate from the already-known background knowledge. David Hand, who organized a workshop in 2002, proposed the new field of local patterns.

The Dagstuhl Seminar in April 2004 on Local Pattern Detection brought together experts from Europe, Japan, and the United States – 13 countries were represented. Moreover, the participants brought with them expertise in the following fields: decision trees, regression methods, bayesian models, kernel methods, inductive logic programming, deductive databases, constraint propagation, time series analysis, query optimization, outlier detection, frequent set mining, and subgroup detection. All talks were focused on the topic of local patterns in order to come to a clearer view of this new field.

Novelty of Local Pattern Detection

Researchers have investigated global models for a long time in statistics and machine learning. The database community has inspected the storage and retrieval of very large datasets. When statistical methods encounter the extremely large amount of records and the high dimensionality of the stored observations, exploratory methods failed. Machine learning already scales up to build up global

[1] I. Guyon, N. Matic and V. Vapnik. Discovering informative patterns and data cleaning. In *Advances in Knowledge Discovery and Data Mining* (pp. 181–204). AAAI Press/MIT Press, 1996.

models, either in the form of complete decision functions or in the form of learning all valid rules from databases. However, the classification does not deliver new, surprising insights into the data, and the valid rules reflect almost exactly the domain knowledge of the database designers. In contrast, what users expect from the exploratory analysis of databases are new insights into their data. Hence, the matter of interestingness has become a key issue. The success of Apriori or subsequently frequent set mining can be explained by it being the first step into the direction of local patterns. The correlation of more than the few features, which standard statistics could analyze, could successfully be determined by frequent set mining. Frequent set mining already outputs local patterns. Current research tasks within this set of methods include algorithmic concerns as well as the issues of interestingness measures and redundancy prevention. The collaboration of database specialists and data miners has led to the notion of inductive databases. The new approach writes measures of interest and the prevention of redundancy in terms of constraints. Also users can formulate their interests in terms of constraints. The constraints are pushed into the search process. This new approach was discussed at the seminar intensively and a view was found that covered diverse aspects of local patterns, namely their internal structure and the subjective part of interestingness as given by users.

Not all the exciting talks and contributions made their way into this book, particularly when a version of the talk was published elsewhere:

- Rosa Meo presented a language for inductive queries expressing constraints in the framework of frequent set mining.
- Bart Goethals offered a new constraint on the patterns, namely that of the database containing the minimal number of tiles, where each tile has the maximal number of '1'.
- Stefan Wrobel gave an in-depth talk on subgroup discovery, where he clearly indicated the problem of false discoveries and presented two approaches: the MIDOS algorithm, which finds subgroups according to the true deviation, and a sequential sampling algorithm, GSS, which makes subgroup discovery fast. He also tackled the redundancy problem by maximum entropy suppression effectively. Applications on spatial subgroup discovery concluded the talk.
- Arno Siebes employed a graphical view on data and patterns to express this internal structure. Moreover, aggregate functions along paths in these graphs were used to compute new features.
- Helena Ahonen-Myka gave an overview of sequence discovery with a focus on applications on text.
- Xiaohui Liu explained how to build a noise model using supervised machine learning methods and detect local patterns on this basis. Testing them against the noise model yields clean data. The approach was illustrated with two biomedical applications.
- Thorsten Joachims investigated internal structures such as parse-trees and co-reference pairing. He presented a general method for how such structures can be analyzed by SVMs. Moreover he showed how the combinatorial ex-

plosion of the number of constraints can be controlled by the upper bounds derived from statistical learning theory.

The book then covers frequent set mining in the following chapters:

- Francesco Bonchi and Fosca Giannotti show the use of constraints within the search for local patterns.
- Jean-Francois Boulicaut applies frequent set mining to gene expression data by exploiting Galois operators and mining bi-sets, which link situations and genes.
- Cline Rouveirol reports on the combination of frequent sets found in gene expression and genome alteration data.

Subgroup discovery is represented by three chapters:

- Nada Lavrac reports on successful applications of subgroup mining in medicine.
- Josef Fürnkranz presents a unifying view of diverse evaluation measures.
- Einoshin Suzuki investigates evaluation measures in order to distinguish local patterns from noise.
- Martin Scholz identifies global models with prior knowledge and local patterns with further, unexpected regularities. His subgroup discovery exploits iteratively a knowledge-based sampling method.

The statistical view is presented in the following chapters:

- Niall Adams and David Hand distinguish two stages in pattern discovery
  1. identify potential patterns (given a suitable definition);
  2. among these, identify significant (in some sense) patterns (expert or automatic).

  They notice that the former is primarily algorithmic and the latter has the potential to be statistical. They illustrate this with an application on discovering cheating students.
- Frank Höppner discusses the similarities and differences between clustering and pattern discovery. In particular he shows how interesting patterns can be found by the clever use of a hierarchical clustering algorithm.
- Stefan Rüping introduces a general framework in which local patterns being produced by different processes are identified using a hidden variable. This allows for the use of the EM algorithm to discover the local patterns directly, that is, without reference to the global data distribution. A new scaling algorithm handles the combination of classifiers. The method is illustrated using business cycle data.

Phenomena of time have always been of interest in KDD, ranging from time series analysis to episode learning. Here, three chapters are devoted to time phenomena:

- Claus Weihs focuses on the transformation of local patterns into global models illustrated with the transcription of vocal time series into sheet music.

- Katharina Morik discusses the importance of the example representation, because it determines the applicability of methods. For local pattern detection, frequency features are well suited. She shows how to characterize time-stamped data using a frequency model.
- Myra Spiliopoulou gives an overview of local patterns exhibiting temporal structures, namely changes of (learned) concepts.

Seminar Results

Based on the definition of David Hand[2]

data = background model + local patterns + random

seminar participants came up with 12 definitions of what local patterns actually are. These were intensively discussed and we finally agreed on the following:

- Local patterns cover small parts of the data space. If the learning result is considered a function, then global models are a complete function, whereas local patterns are partial.
- Local patterns deviate from the distribution of the population of which they are part. This can be done iteratively — a local pattern can be considered the overall population and deviating parts of it are then determined.
- Local patterns show some internal structure. For example, correlations of features, temporal or spatial ordering attributes, and sequences tie together instances of a local pattern.

Local patterns pose very difficult mining tasks:

- Interestingness measures differ from standard criteria for global models.
- Deviation from background knowledge (global model) requires good estimates of the global mode, where local patterns deviate from the overall distribution.
- Modeling noise (for data cleaning, distinguished from local patterns).
- Automatic feature generation and selection for local patterns (for local patterns other features are more successful than for global models; standard feature selection does not work).
- Internal structures of the patterns (correlations of several features, graphs, sequences, spatial closeness, shapes) can be expressed in several ways, e.g., TCat, constraints.
- Test theory for an extremely large space of possible hypotheses (large sets are less likely, hence global models do not encounter this problem).
- Curse of exponentiality — complexity issues.
- Redundancy of learned patterns.
- Sampling for local patterns speeds up mining and enhances quality of patterns.

[2] David Hand. Pattern Detection and Discovery. In David Hand, Niall Adams and Richard Bolton, editors, *Pattern Detection and Discovery*, Springer, 2002.

- Evaluation: benchmark missing.
- Algorithm issues.

We hope that this books reflects the issues of local pattern detection and inspires more research and applications in this exciting field.

Katharina Morik
Arno Siebes
Jean-Francois Boulicaut

Lecture Notes in Artificial Intelligence (LNAI)

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

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 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. 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. Dey, B. Kokinov, D. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 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. Mitrovic (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. Stabler, J. Busquets, R. Moot (Eds.), Logical Aspects of Computational Linguistics. X, 363 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. 3476: J. 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. 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. Faundez-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. Faltings, A. Petcu, F. Fages, F. Rossi (Eds.), Constraint Satisfaction and Constraint Logic Programming. X, 217 pages. 2005.

Vol. 3416: M. 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. 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. Dignum (Eds.), Agent Communication. X, 261 pages. 2005.

Vol. 3394: D. Kudenko, D. Kazakov, E. Alonso (Eds.), Adaptive Agents and Multi-Agent Systems II. VIII, 313 pages. 2005.

Vol. 3392: D. Seipel, M. Hanus, U. Geske, O. Bartenstein (Eds.), Applications of Declarative Programming and Knowledge Management. X, 309 pages. 2005.

Vol. 3374: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems. X, 279 pages. 2005.

Vol. 3371: M.W. Barley, N. Kasabov (Eds.), Intelligent Agents and Multi-Agent Systems. X, 329 pages. 2005.

Vol. 3369: V.R. Benjamins, P. Casanovas, J. Breuker, A. Gangemi (Eds.), Law and the Semantic Web. XII, 249 pages. 2005.

Vol. 3366: I. Rahwan, P. Moraitis, C. Reed (Eds.), Argumentation in Multi-Agent Systems. XII, 263 pages. 2005.

Vol. 3359: G. Grieser, Y. Tanaka (Eds.), Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets. XIV, 257 pages. 2005.

Vol. 3346: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 249 pages. 2005.

Vol. 3345: Y. Cai (Ed.), Ambient Intelligence for Scientific Discovery. XII, 311 pages. 2005.

Vol. 3343: C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, T. Barkowsky (Eds.), Spatial Cognition IV. XIII, 519 pages. 2005.

Vol. 3339: G.I. Webb, X. Yu (Eds.), AI 2004: Advances in Artificial Intelligence. XXII, 1272 pages. 2004.

Vol. 3336: D. Karagiannis, U. Reimer (Eds.), Practical Aspects of Knowledge Management. X, 523 pages. 2004.

Vol. 3327: Y. Shi, W. Xu, Z. Chen (Eds.), Data Mining and Knowledge Management. XIII, 263 pages. 2005.

Vol. 3315: C. Lemaître, C.A. Reyes, J.A. González (Eds.), Advances in Artificial Intelligence – IBERAMIA 2004. XX, 987 pages. 2004.

Vol. 3303: J.A. López, E. Benfenati, W. Dubitzky (Eds.), Knowledge Exploration in Life Science Informatics. X, 249 pages. 2004.

Vol. 3301: G. Kern-Isberner, W. Rödder, F. Kulmann (Eds.), Conditionals, Information, and Inference. XII, 219 pages. 2005.

Vol. 3276: D. Nardi, M. Riedmiller, C. Sammut, J. Santos-Victor (Eds.), RoboCup 2004: Robot Soccer World Cup VIII. XVIII, 678 pages. 2005.

Vol. 3275: P. Perner (Ed.), Advances in Data Mining. VIII, 173 pages. 2004.

Vol. 3265: R.E. Frederking, K.B. Taylor (Eds.), Machine Translation: From Real Users to Research. XI, 392 pages. 2004.

Vol. 3264: G. Paliouras, Y. Sakakibara (Eds.), Grammatical Inference: Algorithms and Applications. XI, 291 pages. 2004.

Vol. 3259: J. Dix, J. Leite (Eds.), Computational Logic in Multi-Agent Systems. XII, 251 pages. 2004.

Vol. 3257: E. Motta, N.R. Shadbolt, A. Stutt, N. Gibbins (Eds.), Engineering Knowledge in the Age of the Semantic Web. XVII, 517 pages. 2004.

Vol. 3249: B. Buchberger, J.A. Campbell (Eds.), Artificial Intelligence and Symbolic Computation. X, 285 pages. 2004.

Vol. 3248: K.-Y. Su, J. Tsujii, J.-H. Lee, O.Y. Kwong (Eds.), Natural Language Processing – IJCNLP 2004. XVIII, 817 pages. 2005.

Vol. 3245: E. Suzuki, S. Arikawa (Eds.), Discovery Science. XIV, 430 pages. 2004.

Vol. 3244: S. Ben-David, J. Case, A. Maruoka (Eds.), Algorithmic Learning Theory. XIV, 505 pages. 2004.

Vol. 3238: S. Biundo, T. Frühwirth, G. Palm (Eds.), KI 2004: Advances in Artificial Intelligence. XI, 467 pages. 2004.

Vol. 3230: J.L. Vicedo, P. Martínez-Barco, R. Muñoz, M. Saiz Noeda (Eds.), Advances in Natural Language Processing. XII, 488 pages. 2004.

Vol. 3229: J.J. Alferes, J. Leite (Eds.), Logics in Artificial Intelligence. XIV, 744 pages. 2004.

Vol. 3228: M.G. Hinchey, J.L. Rash, W.F. Truszkowski, C.A. Rouff (Eds.), Formal Approaches to Agent-Based Systems. VIII, 290 pages. 2004.

Vol. 3215: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LVII, 906 pages. 2004.

Vol. 3214: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LVIII, 1302 pages. 2004.

Vol. 3213: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LVIII, 1280 pages. 2004.

Vol. 3209: B. Berendt, A. Hotho, D. Mladenic, M. van Someren, M. Spiliopoulou, G. Stumme (Eds.), Web Mining: From Web to Semantic Web. IX, 201 pages. 2004.

Vol. 3206: P. Sojka, I. Kopecek, K. Pala (Eds.), Text, Speech and Dialogue. XIII, 667 pages. 2004.

Vol. 3202: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Knowledge Discovery in Databases: PKDD 2004. XIX, 560 pages. 2004.

Vol. 3201: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Machine Learning: ECML 2004. XVIII, 580 pages. 2004.

Vol. 3194: R. Camacho, R. King, A. Srinivasan (Eds.), Inductive Logic Programming. XI, 361 pages. 2004.

Vol. 3192: C. Bussler, D. Fensel (Eds.), Artificial Intelligence: Methodology, Systems, and Applications. XIII, 522 pages. 2004.

Vol. 3191: M. Klusch, S. Ossowski, V. Kashyap, R. Unland (Eds.), Cooperative Information Agents VIII. XI, 303 pages. 2004.

Vol. 3187: G. Lindemann, J. Denzinger, I.J. Timm, R. Unland (Eds.), Multiagent System Technologies. XIII, 341 pages. 2004.

Vol. 3176: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), Advanced Lectures on Machine Learning. IX, 241 pages. 2004.

Vol. 3171: A.L.C. Bazzan, S. Labidi (Eds.), Advances in Artificial Intelligence – SBIA 2004. XVII, 548 pages. 2004.

Vol. 3159: U. Visser, Intelligent Information Integration for the Semantic Web. XIV, 150 pages. 2004.

Vol. 3157: C. Zhang, H. W. Guesgen, W.K. Yeap (Eds.), PRICAI 2004: Trends in Artificial Intelligence. XX, 1023 pages. 2004.

Vol. 3155: P. Funk, P.A. González Calero (Eds.), Advances in Case-Based Reasoning. XIII, 822 pages. 2004.

Vol. 3139: F. Iida, R. Pfeifer, L. Steels, Y. Kuniyoshi (Eds.), Embodied Artificial Intelligence. IX, 331 pages. 2004.

Vol. 3131: V. Torra, Y. Narukawa (Eds.), Modeling Decisions for Artificial Intelligence. XI, 327 pages. 2004.

¥424.80元

Table of Contents

Pushing Constraints to Detect Local Patterns

Francesco Bonchi and Fosca Giannotti

Pisa KDD Laboratory
ISTI - CNR, Area della Ricerca di Pisa, Via Giuseppe Moruzzi, 1 - 56124 Pisa, Italy
{francesco.bonchi, fosca.giannotti}@isti.cnr.it

Abstract. The main position of this paper is that constraints can be a very useful tool in the search for local patterns. The justification for our position is twofold. On one hand, pushing constraints makes feasible the computation of frequent patterns at very low frequency levels, which is where local patterns are. On the other hand constraints can be exploited to guide the search for those patterns showing deviating, surprising characteristics. We first review the many definitions of local patterns. This review leads us to justify our position. We then provide a survey of techniques for pushing constraint into the frequent pattern computation.

1 Introduction

The collection of large electronic databases of scientific and commercial information has led to a dramatic growth of interest in methods for discovering structures in such databases. These methods often go under the general name of *data mining*. However, recently two different kinds of structures sought in data mining have been identified: *global models* and *local patterns*.

Traditionally, research in statistics and machine learning has investigated methods to build *global models*, i.e. high level descriptive summarizations of the general structure of the data. Examples are statistical time series models, cluster models, logic programs with high coverage or classification models like decision trees, or liner regression. The intrinsic global nature turned out to be the main drawback that these methods face in practical applications. Having a global point of view over the data, these methods rarely produce new and surprising insight: in fact, in order to be valid, they must summarize most of the data and thus they usually represent obvious knowledge that domain experts are already aware of. On the contrary, what we are seeking for is interesting and surprising knowledge which *deviates* from the already known background model.

Therefore the detection of *local patterns* [14,20] has recently emerged as a new research field with its own distinguished role within data mining. Local patterns are small configurations of data which may involve just a few points or variables, and which are of special interest because they exhibit a deviating behavior w.r.t. the underlying global model. The new field of *local pattern detection* has been proposed by Hand who organized a workshop in 2002 [15]. Such initiative gathered together researchers active in different fields (ranging from statistics to multi-relational data mining, from machine learning to inductive databases) but sharing a common interest for local patterns.

K. Morik et al. (Eds.): Local Pattern Detection, LNAI 3539, pp. 1–19, 2005.

In spring 2004, following the first successful workshop, a Dagstuhl seminar has been organized with the declared goal of finding a definition of local patterns on which most of the participants agree. The lively discussion has produced many slightly different definitions.

1.1 Frequent Pattern Discovery

Even if a rigorous definition of local pattern is still missing, many recognize the successful idea of Apriori [1,2] as a first step into the direction of local patterns. During the last decade a lot of researchers have focussed their (mainly methodological and algorithmic) investigations on the computational problem of *Frequent Pattern Discovery*, i.e. mining patterns which satisfy a user-defined minimum threshold of frequency [2,13].

The simplest form of a frequent pattern is the frequent itemset.

Definition 1 (Frequent Itemset Mining). *Let* $\mathcal{I} = \{x_1, ..., x_n\}$ *be a set of distinct literals, usually called* items, *where an item is an object with some predefined attributes (e.g., price, type, etc.). An* itemset X *is a non-empty subset of* $\mathcal{I}$. *If* $|X| = k$ *then* X *is called a* k-itemset. *A* transaction database $\mathcal{D}$ *is a bag of itemsets* $t \in 2^{\mathcal{I}}$, *usually called* transactions. *The* support *of an itemset* X *in database* $\mathcal{D}$, *denoted* $supp_{\mathcal{D}}(X)$, *is the number of transactions which are superset of* X. *Given a user-defined* minimum support σ, *an itemset* X *is called* frequent *in* $\mathcal{D}$ *if* $supp_{\mathcal{D}}(X) \geq \sigma$. *This defines the minimum frequency constraint:* $\mathcal{C}_{freq[\mathcal{D},\sigma]}(X) \Leftrightarrow supp_{\mathcal{D}}(X) \geq \sigma$. *When the dataset and the minimum support are clear from the context, we indicate the frequency constraint simply* $\mathcal{C}_{freq}$.

This computational problem is at the basis of the well known *Association Rules* mining. An association rule is an expression $X \Rightarrow Y$ where X and Y are two itemsets. The association rule is said to be *valid* if the support of the itemset $X \cup Y$ is greater than a given threshold, and if the *confidence* (or accuracy) of the rule, defined as the conditional probability $P(Y \mid X)$, is greater than a given threshold. However frequent itemsets are meaningful not only in the context of association rules mining: they can be used as basic element in many other kind of analysis, ranging from classification [18,19] to clustering [26,29].

Recently the research community has turned its attention to more complex kinds of frequent patterns extracted from more structured data: sequences, trees, and graphs. All these different kinds of pattern have different peculiarities and application fields, (i.e. sequences are particular well suited for *business applications*, frequent subtrees can be mined from a set of *XML documents*, and frequent substructures from graphs can be useful, for instance, in *biological applications*, in *drug design* and in *Web-mining*), but they all share the same computational aspects: a usually very large input, an exponential search space, and a too large solution set. This situation – too many data yielding too many patterns – is harmful for two reasons. First, performance degrades: mining generally becomes inefficient or, often, simply unfeasible. Second, the identification of the fragments of interesting knowledge, blurred within a huge quantity of mostly useless patterns, is difficult. Therefore, the paradigm of *constraint-based mining* was introduced. Constraints provide focus on the interesting knowledge, thus reducing

the number of patterns extracted to those of potential interest. Additionally, they can be pushed deep inside the mining algorithm in order to achieve better performance [3,4,5,6,7,8,9,11,12,16,17,21,24,25,28].

Definition 2 (Constrained Frequent Itemset Mining). *A constraint on itemsets is a function* $C : 2^{\mathcal{I}} \rightarrow \{true, false\}$. *We say that an itemset* I *satisfies a constraint if and only if* $C(I) = true$. *We define the theory of a constraint as the set of itemsets which satisfy the constraint:* $Th(C) = \{X \in 2^{\mathcal{I}} \mid C(X)\}$. *Thus with this notation, the* frequent itemsets mining problem *requires to compute the set of all frequent itemsets* $Th(C_{freq[\mathcal{D},\sigma]})$. *In general, given a conjunction of constraints* C *the* constrained frequent itemsets mining problem *requires to compute* $Th(C_{freq}) \cap Th(C)$.

In this paper we argue that constraints can be exploited in order to guide the search for local patterns.

2 On Locality, Frequency, Deviation and Constraints

Extending a classical statistical modelling perspective with local patterns, Hand provided the following definition [14]:

$$data = background\_model + local\_patterns + random\_component$$

Based on such definition the participants of the 2004 Dagstuhl seminar discussed in order to find out what a local pattern precisely is.

At the end the participants agreed at least on the following features that a structure must exhibit in order to be considered a local pattern.

1. local patterns cover small parts of the data space;
2. local patterns deviate from the distribution of the population of which they are part;
3. local patterns show some internal structure.

In this Section we will review some of the definitions provided during the seminar by the point of view of frequent pattern mining, arriving to justify our main position: pushing constraints is an important technique to detect local patterns. Hence, let us focus on frequent patterns and association rules. We start our investigation with the obvious question:

Can association rules be considered local patterns?

The answer is not straightforward. On one hand, if we mine the complete set of valid (w.r.t. reasonable support and confidence thresholds) association rules what we obtain is a global descriptive summarization of the data, which for instance, could be used also for classification purposes [19]. Hence what we get is a global model. On the other hand, we could take in consideration a single association rule: is this a local pattern? An association rule is a simple descriptive structure which is true for a reasonably large fraction of the data. It can be understood in isolation and there is no direct attempt at a global description of the data. Thus we could conclude that an association rule is a local pattern.

Consider now an association rule with a very high support, covering a large part of the data and representing some obvious knowledge that domain experts are already aware of: is this still a local pattern?

From the above considerations two questions arise:

1. *How much local is an association rule?*
2. *When does an association rule really represents some interesting knowledge?*

The two questions correspond to the first two features (listed above) that a structure must exhibit in order to be considered a local pattern: *locality* and *deviation*. Therefore in the following we will discuss the concepts of locality and deviation when applied to association rules and frequent patterns.

Locality (and Frequency) - In [14] Hand states: "*In order to define what is meant by local it is necessary to adopt a suitable distance measure. This choice will depend on the data and the application domain: in the cases of categorical variables, it might even require exact matches. It is also necessary to pick a threshold with which the measured distance is compared.*"

When talking about frequent patterns is quite obvious to think about frequency as the measure of locality: a very frequent pattern is global (i.e. it covers a large part of the data), a not so frequent pattern is local (i.e. it covers only a small portion of the data). This agrees with the position of Morik who defined a local pattern as a description of *rare* events, which deviate from a global model and show an internal structure.

The need for mining patterns at very low support levels is confirmed also by the applications. In [22,23] association rules are extracted in a medical context. The authors state that most relevant rules with high quality (domain-dependent) metrics appear only at low frequency levels. In [10] association rules are mined with the aim of finding interesting associations between road number, weather and light conditions and serious or fatal accidents. The authors state that interesting (according to feedback from end-user) association rules were found only at very low support levels.

Therefore, a local pattern is a "*rare*" or "*not so frequent*" pattern. But at the same time it needs some support in order to distinguish by the mere random component. So a local pattern must be frequent but not too much.

These last considerations lead us to reformulate Hand's definition by the point of view of frequency:

$$data = very\_frequent\_patterns + rare\_patterns + random\_component$$

But how can reasonable support thresholds be defined? Hand states [14]: "*Some sort of compromise is needed. One way to find a suitable compromise would be to gradually expand the distance threshold defining local, so that the set of points identified as possible patterns gradually increases, stopping when the detected number of patterns seemed reasonable (a decision which depends on resources as well as on the phenomenon under investigation).*"

Deviation (and Constraints) - Extracting too many uninteresting frequent patterns, with large requirements both in terms of time and space, is an even harder problem when mining at very low support level. In fact, the pruning power of the frequency constraint decreases together with the minimum support threshold. When it becomes too low, the search space explodes and the computation becomes intractable.

This decrease in the pruning power of the frequency constraint, can be compensated by the pruning power of other constraints that the user could exploit to restrict the search for interesting patterns.

Example 1. Consider a student database at a university: rows corresponds to students, columns to courses, and a 1 entry (s, c) indicates that student s has taken course c. In other words, students are the transactions and courses taken are items in the transactions. In this context a pattern is a set of courses which satisfy some interestingness conditions. Frequent patterns are set of courses which appear together in the curricula of a number of students larger than a given minimum support threshold.

However the search for patterns can be guided by other interestingness constraints. Suppose that each course has some attributes, such has *semester*, *credits*, *prerequisites*, $difficulty\_rate$. One could be interested in finding frequent set of courses c such that $c.semester = 1$ and $sum(c.credits) \geq 50$. These constraints can be pushed inside the frequent pattern algorithm, reducing the search space and thus enabling mining at low frequency levels.

Constraints are not only useful to prune the search space, thus reducing the computation. They have also a semantic value since the language of constraints is what the user exploits in order to define which are the *interesting* patterns.

The importance of constraints in the search for local patterns is confirmed by other definitions which emerged at the Dagstuhl seminar. According to Siebes "*local patterns are described by structural requirements, virtual attributes, and conditions on attribute values.*" Similarly Boulicaut states that a local pattern is "*a sentence from a pattern language that is apriori interesting since it satisfies a given set of constraints and tells something about part of the data.*"

As stated before, one important feature of local patterns, is deviation. Constraints can be our guide in the search for deviating patterns.

Example 2. Consider again our student/courses database. Suppose the average $difficulty\_rate$ of courses to be 0.35. We can search for set of courses c, (not so) frequently taken together, and such that $avg(c.difficulty\_rate) \geq 0.95$. If we find a solution to this pattern query, this is clearly an interesting deviating pattern.

The idea of the previous example might be generalized by the following naïve definition.

Definition 3 (Deviating Pattern). *Let $\mathcal{I} = \{x_1, ..., x_n\}$ be a set of items. Let A_i be a non-negative real-valued attribute associated to each item. Given a minimum deviation factor $\delta > 1$; an itemset X is said to be deviating on attribute A_i if:*

$$avg(X.A_i) \geq \delta \cdot avg(\mathcal{I}.A_i) \quad \text{or if} \quad avg(X.A_i) \leq \frac{avg(\mathcal{I}.A_i)}{\delta}$$