

Exploratory Mining via Constrained Frequent Set Queries

Raymond Ng

U. of British Columbia

rng@cs.ubc.ca

Laks V.S. Lakshmanan

Concordia U.

laks@cs.concordia.ca

Jiawei Han

Simon Fraser U.

han@cs.sfu.ca

Teresa Mah

U. of British Columbia

tmah@cs.ubc.ca

Abstract

Although there have been many studies on data mining, to date there have been few research prototypes or commercial systems supporting comprehensive query-driven mining, which encourages interactive exploration of the data. Our thesis is that constraint constructs and the optimization they induce play a pivotal role in mining queries, thus substantially enhancing the usefulness and performance of the mining system. This is based on the analogy of declarative query languages like SQL and query optimization which have made relational databases so successful. To this end, our proposed demo is not yet another data mining system, but of a new paradigm in data mining – mining with constraints, as the important first step towards supporting ad-hoc mining in DBMS.

In this demo, we will show a prototype exploratory mining system that implements constraint-based mining query optimization methods proposed in [5]. We will demonstrate how a user can interact with the system for exploratory data mining and how efficiently the system may execute optimized data mining queries. The prototype system will include all the constraint pushing techniques for mining association rules outlined in [5], and will include additional capabilities for mining other kinds of rules for which the computation of constrained frequent sets forms the core first step.

1 Background and Significance to the Community

Since the introduction of association rules [1], the development of effective mechanisms for mining large databases has been the subject of numerous studies, which can be broadly divided into two groups. The first group includes studies focusing on performance and efficiency issues; while the second group includes studies that go beyond the initial notion of association rules to other kinds of mined rules. Recently it has been recognized that

the integration of data mining technologies with database management systems is of strategic importance [3]. Furthermore, it has been argued that the fundamental distinction of a data mining system from a statistical analysis program or a machine learning system should be that the former offers an ad-hoc mining query language and supports efficient processing and optimization of mining queries [4]. Sarawagi et al. [6] study the suitability of different architectures for the integration of association mining with DBMS and study the relative performance tradeoffs. Tsur et al. [8] explore the question of how techniques like the well-known Apriori algorithm can be generalized beyond their current applications to a generic paradigm called query flocks.

While these are important results toward enabling the integration of association mining and DBMS, ad-hoc mining still cannot be supported until the following fundamental problems in the present-day model of mining, first identified in [5], are addressed satisfactorily:

- **Problem 1 – Lack of User Exploration and Control:** Mining (of associations) should be an activity that allows for *exploration* on the user's part [4, 7]. However, the present day model for mining treats the mining process as an impenetrable black-box – only allowing the user to set the thresholds at the beginning, showing the user all associations satisfying the thresholds at the end, but nothing in between. What if the user sets the wrong thresholds, or simply wants to change them? What if the user wants to focus the generation of rules to a specific, small subset of candidates, based on properties of the data? Such a black-box model would be tolerable if the turnaround time of the computation were small, e.g., a few seconds. However, despite the development of many efficient algorithms association mining remains a process typically taking hours to complete. Before a new invocation of the black-box, the user is not allowed to preempt the process and needs to wait for hours. Furthermore, typically only a small fraction of the computed rules might be what the user was looking for. Thus the user often incurs a high computational cost that is disproportionate to what the user wants and gets.
- **Problem 2 – Lack of Focus:** A user may have certain broad phenomena in mind,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGMOD '99 Philadelphia PA

Copyright ACM 1999 1-58113-084-8/99/05...\$5.00

on which to focus the mining. For example, the user may want to find associations between sets of items whose types do not overlap, or associations from item sets whose total price is under \$100 to items sets whose average price is at least \$1,000 (thereby verifying whether the purchases of cheap items occur together with those of expensive ones). The interface for expressing focus offered by the present-day model is extremely impoverished, because it only allows thresholds for support and confidence to be specified.

- **Problem 3 – Rigid Notion of Relationship:** The present-day model restricts the notion of associations to rules with support and confidence that exceed given thresholds. While such associations are useful, other notions of relationships may also be useful. First, there exist several significance metrics other than confidence that are equally meaningful. For example, Brin et al. argue why correlation can be more useful in many circumstances [2]. Second, there may be separate criteria for selecting candidates for the antecedent and consequent of a rule. For example, the user may want to find associations from sets of items to sets of types. Coming from different domains, the antecedent and consequent may call for different support thresholds and different conditions to be met.

To address these problems, in [5] we proposed a 2-phase architecture for exploratory mining, which follows the following general principles:

1. Open up the black-box, and establish clear breakpoints so as to allow user feedback.
2. Incorporate user feedback, not only for guidance and control of the mining process, but also for acquiring user's approval for any task involving a substantial cost.
3. Provide the user with many opportunities to express the focus.
4. Use the focus to ensure that the system does an amount of computation proportional to what the user gets. This is the first step towards ad hoc mining of rules [4, 7].
5. Allow the user to have the flexibility to choose the significance metrics and the criteria to be satisfied by the relationships to be mined.

The proposed architecture provides a rich interface for the user to express focus. The critical component is the notion of *constrained frequent set queries* (CFQs), which offer the user a means for specifying constraints, including domain, class and aggregation constraints, that must be satisfied by the antecedents and consequents of the rules to be mined. Moreover, towards the goal of doing an amount of processing commensurate with the focus specified by the user, we develop in [5] various pruning optimizations effected by the constraints. We introduced two key properties called anti-monotonicity and succinctness and showed how these properties of constraints can

be exploited to give powerful optimization of CFQs. We proposed an algorithm called CAP that incorporates these optimizations and delivers a performance that is up to 80 times faster than several algorithms based on the classical Apriori framework.

It is important to note that the user need not know anything about anti-monotonicity, succinctness, or any properties of the constraints. The constraints come from his mining objectives and his understanding of the application domain, just as for ad hoc DBMS queries. But based on the characterization of the constraints given in [5], CAP can automatically identify the best strategy for processing the constraints and deliver a level of performance that is commensurate with the extent of pruning induced by the constraints.

2 What will be Demonstrated?

Our demo will show a prototype exploratory mining system that implements the two-phase architecture outlined in [5]. In particular, Figure 1 shows the implemented architecture.

In the form of a constrained frequent set query, the user initially specifies a set of constraints $\mathcal{C}$, including the support thresholds, for the antecedent and consequent. Each constraint in $\mathcal{C}$ may be applicable to the antecedent, or the consequent, or both. The output of phase I consists of a list of pairs of candidates (S_a, S_c) , for the antecedent and consequent satisfying $\mathcal{C}$, such that both S_a and S_c have a support exceeding the thresholds initially set by the user.

In general the candidate list can be quite large, running in the order of tens of thousands of pairs. To help the user browse through the candidate list conveniently, our prototype has implemented various ways to organize the pairs in the list. One way is to order the list with respect to set inclusion and show only pairs of maximal sets. Another is to provide ranking of the sets based on their supports and the degrees to which they satisfy the given constraints, thus providing feedback to the user as to whether the constraints or the support threshold need to be adjusted. On seeing the initial list of candidates, the user can: (i) add, delete, or modify the constraints, and/or (ii) adjust the support thresholds. The user may iterate through Phase I in this manner as many times as desired.

Once satisfied with the current candidate list, the user can instruct the system to proceed to Phase II, wherein the user has the opportunity to specify: (i) the significance metric, (ii) a threshold for the metric specified above, and (iii) further conditions to be imposed on the antecedent and consequent. For instance, if the user wishes to operate in the classical association mining setting, the user would choose confidence as the significance metric, give a confidence threshold, and require that $(S_a \cup S_c)$ be frequent. Alternatively, the user may wish to form rules with the correlation of the antecedent and the consequent exceeding a given threshold. In this case, the user selects correlation as the significance metric and specifies the minimum correlation coefficient. Essentially, the effort spent in Phase II is geared towards the computation of what

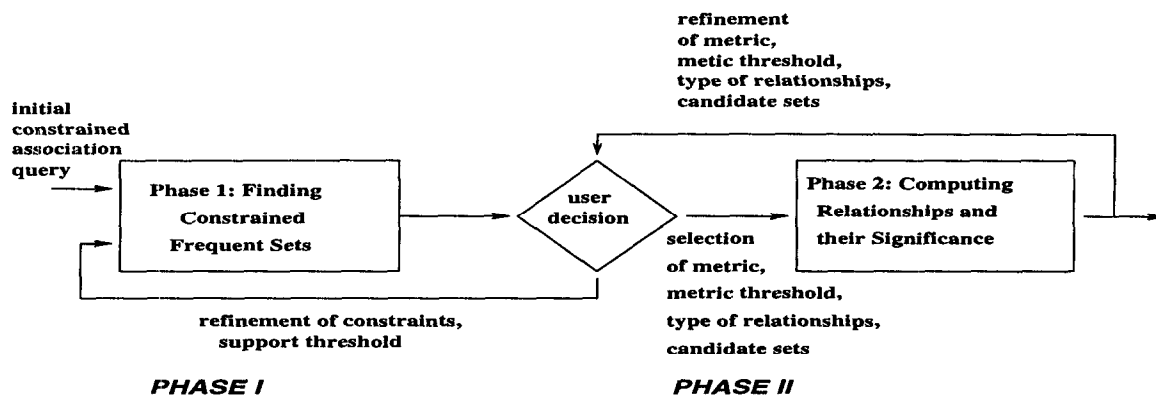


Figure 1: An Architecture for Exploratory Mining with CFQs

the user really wants. Even if Phase II involves costly computation (e.g., computation of correlations), the user has the final say in authorizing such costly operations.

Finally, the output of Phase II consists of all relationships that satisfy the conditions given at the beginning of Phase II. Upon examining this output, the user has the opportunity to make further changes to any parameters set before. Depending on which parameters are reset, this may yet trigger Phase I and Phase II, or just Phase II, computation.

A key feature of the proposed architecture is that it is downward compatible. This means that if a user wants only classical associations and the classical mode of interaction, the user can simply set all the appropriate parameters at the beginning, and need not be prompted at the breakpoints for feedback. Of course, we stress that the real power of the architecture stems from its provision for human-centered exploration for rule mining, and its implementation of the five principles suggested in the previous section.

The architecture *per se* does not address performance issues. Nonetheless, most of the pruning optimizations developed in [5] have been implemented. Thus, as part of the demonstration, we can observe how effective the developed optimizations can be.

In sum, the demo will show a state-of-the-art interactive mining system with constrained frequent sets. We believe that this system forms the important first step towards supporting ad-hoc mining of association and other related kinds of rules, and the eventual integration of mining technologies with DBMS.

References

- [1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data*, pages 207–216, Washington, D.C., May 1993.
- [2] S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. In *Proc. 1997 ACM-SIGMOD*, pp 265–276.
- [3] S. Chaudhuri. Data mining and database systems: Where is the intersection? *Bulletin of the Technical Committee on Data Engineering*, 21:4–8, March 1998.
- [4] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. *Communications of ACM*, 39:58–64, 1996.
- [5] R. Ng, L. V. S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained associations rules. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data*, pages 13–24, Seattle, Washington, June 1998.
- [6] S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data*, pages 343–354, Seattle, Washington, June 1998.
- [7] A. Silberschatz and S. Zdonik. Database systems – breaking out of the box. *ACM SIGMOD Record*, 26:36–50, 1997.
- [8] D. Tsur, J. D. Ullman, S. Abitboul, C. Clifton, R. Motwani, and S. Nestorov. Query flocks: A generalization of association-rule mining. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data*, pages 1–12, Seattle, Washington, June 1998.