Condensed Representations for Data Mining

Jean-Francois Boulicaut
INSA de Lyon, France

INTRODUCTION

Condensed representations have been proposed in Mannila and Toivonen (1996) as a useful concept for the optimization of typical data-mining tasks. It appears as a key concept within the inductive database framework (Boulicaut et al., 1999; de Raedt, 2002; Imielinski & Mannila, 1996), and this article introduces this research domain, its achievements in the context of frequent itemset mining (FIM) from transactional data, and its future trends.

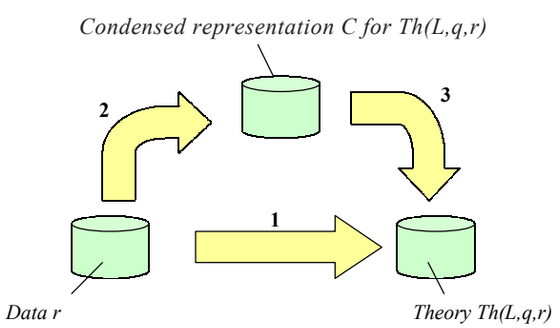
Within the inductive database framework, knowledge discovery processes are considered as querying processes. Inductive databases (IDBs) contain not only data, but also patterns. In an IDB, ordinary queries can be used to access and manipulate data, while inductive queries can be used to generate (mine), manipulate, and apply patterns. To motivate the need for condensed representations, let us start from the simple model proposed in Mannila and Toivonen (1997). Many data-mining tasks can be abstracted into the computation of a theory. Given a language L of patterns (e.g., itemsets), a database instance r (e.g., a transactional database) and a selection predicate q, which specifies whether a given pattern is interesting or not (e.g., the itemset is frequent in r), a data-mining task can be formalized as the computation of $Th(L,q,r) = \{\phi \in L \mid q(\phi,r) \text{ is true}\}$. This also can be considered as the evaluation for the inductive query q. Notice that it specifies that every pattern that satisfies q has to be computed. This completeness assumption is quite common for local pattern discovery tasks but is generally not acceptable for more complex tasks (e.g., accuracy optimization for predictive model mining). The selection predicate q can be defined in terms of a Boolean expression over some primitive constraints (e.g., a minimal frequency constraint used in conjunction with a syntactic constraint, which enforces the presence or the absence of some subpatterns). Some of the primitive constraints generally refer to the behavior of a pattern in the data by using the so-called evaluation functions (e.g., frequency).

To support the whole knowledge discovery process, it is important to support the computation of many different but correlated theories.

It is well known that a generate-and-test approach that would enumerate the sentences of L and then test the selection predicate q is generally impossible. A huge effort has been made by data-mining researchers to make an active use of the primitive constraints occurring in q to achieve a tractable evaluation of useful mining queries. It is the domain of constraint-based mining (e.g., the seminal paper) (Ng et al., 1998). In real applications, the computation of Th(L,q,r) can remain extremely expensive or even impossible, and the framework of condensed representations has been designed to cope with such a situation. The idea of ε-adequate representations was introduced in Mannila and Toivonen (1996) and Boulicaut and Bykowski (2000). Intuitively, they are alternative representations of the data that enable answering to a class of query (e.g., frequency queries for itemsets in transactional data) with a bounded precision. At a given precision ε, one can be interested in the smaller representations, which are then called concise or condensed representations. It means that a condensed representation for Th(L,q,r) is a collection $C \subset Th(L,q,r)$ such that every pattern from Th(L,q,r) can be derived efficiently from C. In the database-mining context, where r might contain a huge volume of records, we assume that efficiently means without further access to the data. The following figure illustrates that we can compute Th(L,q,r) either directly (Arrow 1) or by means of a condensed representation (Arrow 2) followed by a regeneration phase (Arrow 3).

We know several examples of condensed representations for which Phases 2 and 3 are much less expensive than Phase 1. We now introduce the background for understanding condensed representations in the well studied context of FIM.

Figure 1.

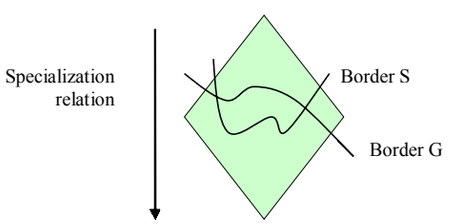


BACKGROUND

In many cases (e.g., itemsets, inclusion dependencies, sequential patterns) and for a given selection predicate or constraint, the search space L is structured by an anti-monotonic specialization relation, which provides a lattice structure. For instance, in transactional data, when L is the power set of items, and the selection predicate enforces a minimal frequency, set inclusion is such an anti-monotonic specialization relation. Anti-monotonicity means that when a sentence does not satisfy q (e.g., an itemset is not frequent), then none of its specializations can satisfy q (e.g., none of its supersets are frequent). It becomes possible to prune huge parts of the search space, which cannot contain interesting sentences. This has been studied a lot within the « learning as search » framework (Mitchell, 1982), and the generic level-wise algorithm from Mannila and Toivonen (1997) has inspired many algorithmic developments. It computes Th(L,q,r) level-wise in the lattice by considering first the most general sentences (e.g., the singleton in the FIM problem). Then, it alternates candidate evaluation (e.g., frequency counting) and candidate generation (e.g., building larger itemsets from discovered frequent itemsets) phases. The algorithm stops when it cannot generate new candidates or, in other terms, when the most specific sentences have been found (e.g., all the maximal frequent itemsets). This collection of the most specific sentences is called a *positive border* in Mannila and Toivonen (1997), and it corresponds to the S set of a Version Space in Mitchell's terminology. The a priori algorithm (Agrawal et al., 1996) is clearly the most famous instance of this level-wise algorithm.

The dual property of monotonicity is interesting, as well. A selection predicate is monotonic when its negation is anti-monotonic (i.e., when a sentence satisfies it, all its specializations satisfy it, as well). In the itemset pattern domain, the maximal frequency constraint or a syntactic constraint that enforces that a given item belongs to the itemsets are two monotonic constraints. Thanks to the duality of these definitions, a monotonic constraint gives rise to a border G, which contains the minimally general sentences w.r.t., the monotonic constraint (see the following figure).

Figure 2.



When the predicate selection is a conjunction of an anti-monotonic part and a monotonic part, the two borders define the solution set: solutions are between S and G, and (S,G) is a version space. For this conjunction case, several algorithms can be used (Bucila et al., 2002; Bonchi, 2003; de Raedt & Kramer, 2001; Jeudy & Boulicaut, 2002). When arbitrary Boolean combinations of anti-monotonic and monotonic constraints are used (e.g., disjunctions), the solution space is defined as a union of several version spaces (i.e., unions of couples of borders) (de Raedt et al., 2002).

Borders appear as a typical case of condensed representation. Assume that the collection of the maximal frequent itemsets in r is available (i.e., the S border for the minimal frequency constraint); this collection is generally several orders of magnitude smaller than the complete collection of the frequent itemsets in r, while all of them can be generated from S without any access to the data. However, in most of the applications of pattern discovery tasks, the user not only wants to get the interesting patterns, but also wants the results of some evaluation functions about these patterns. This is obvious for the FIM problem; these patterns are generally exploited in a post-processing step to derive more useful statements about the data (e.g., the popular frequent association rules that have a high enough confidence) (Agrawal et al., 1996). This can be done efficiently, if we compute not only the collection of frequent itemsets, but also their frequencies.

In fact, the semantics of an inductive query are better captured by extended theories (i.e., collections like $\{(\phi,e) \in L \otimes E \mid q(\phi,r)$ est vrai et $e=\zeta(\phi,r)\}$), where e is the result of an evaluation function ζ in r with values in E. In our FIM problem, ζ denotes the frequency (e is a number in [0,1]) of an itemset in a transactional database r. The challenge of designing condensed representations for an extended theory ThE is then to identify subsets of ThE, from which it is possible to generate ThE either exactly or with an approximation on the evaluation functions.

MAIN RESULTS

We emphasize the main results concerning condensed representations for frequent itemsets, since this is the context for which it has been studied a lot.

Condensed Representations by Borders

It makes sense to use borders as condensed representations. For FIM, specific algorithms have been designed for computing directly the S border (Bayardo, 1998). Also, the algorithm in Kramer and de Raedt (2001) computes