Approximating a set of approximate inclusion dependencies

Fabien De Marchi[1] and Jean-Marc Petit[2]

[1] LIRIS, FRE CNRS 2672, Univ. Lyon 1, 69622 Villeurbanne, France
[2] LIMOS, UMR CNRS 6158, Univ. Clermont-Ferrand II, 63177 Aubière, France

Abstract. Approximating a collection of patterns is a new and active area of research in data mining. The main motivation lies in two observations : the number of mined patterns is often too large to be useful for any end-users and user-defined input parameters of many data mining algorithms are most of the time almost arbitrary defined (e.g. the frequency threshold).

In this setting, we apply the results given in the seminal paper [11] for frequent sets to the problem of approximating a set of approximate inclusion dependencies with *k inclusion dependencies*. Using the fact that inclusion dependencies are "representable as sets", we point out how approximation schemes defined in [11] for frequent patterns also apply in our context. An heuristic solution is also proposed for this particular problem. Even if the quality of this approximation with respect to the best solution cannot be precisely defined, an interaction property between IND and FD may be used to justify this heuristic.

Some interesting perspectives of this work are pointed out from results obtained so far.

1 Introduction

In the relational model, inclusion dependencies (INDs) convey many information on the data structure and data semantics, and generalize in particular foreign keys [1].

Their utility is recognized for many tasks such as semantic query optimization [3], logical and physical database design and tuning [18,2,15,5] or data integration [21]. In practice, the implementation of these technics is generally made impossible by the ignorance of satisfied INDs in the database. One can then be interested in the problem of discovering INDs satisfied in a database [10].

Some algorithms have been recently proposed for this data-mining task [4,12,6]. They output either the whole set of satisfied INDs or only the largest satisfied INDs from which the whole collection of INDs can be inferred. Nevertheless, the interest of discovering only satisfied INDs may be limited in practice: indeed, algorithms being applied on real-life databases, many of them can be qualified as "dirty" databases since constraints may not have been defined. To take into account some data inconsistencies, algorithms have been extended to the discovery of *approximate inclusion dependencies* [4,6]. The idea is to define an error measure from which approximate INDs can be

rigorously defined with respect to an user-defined threshold. The proposed algorithms find exactly all INDs having an error measure lower than the threshold.

Moreover, approximating a collection of patterns is a new and active area of research in data mining. The motivating remarks done in [11] for approximating a collection of frequent itemsets apply for approximate INDs :

1. the output depends on an user-defined threshold. In general, it is almost arbitrary chosen by a user and therefore justifies the approximation of the output.
2. a complete result can be prohibitive in certain cases, since the potential number of satisfied INDs might be very large. For instance, large set of constraints are going to be useless for experts and/or database administrators in order to analyze/maintain the database.

In this paper, we apply the results given in the seminal paper [11] for frequent sets to the problem of approximating a set of approximate inclusion dependencies in a database. Using the fact that inclusion dependencies are "representable as sets" [20,6], we point out how approximation schemes defined in [11] for frequent patterns also apply in our context. An heuristic solution is also proposed for this particular problem. Even if the quality of this approximation with respect to the best solution cannot be precisely defined, an interaction property between IND and FD may be used to justify this heuristic.

Some interesting perspectives of this work are pointed out from results obtained so far.

This work is integrated in a more general project devoted to DBA maintenance and tuning, called "DBA Companion" [5]. We have proposed methods for discovering functional dependency and keys, inclusion dependencies and foreign key and for computing small database samples preserving functional and inclusion dependencies satisfied in the database. A prototype on top of Oracle RDBMS has been developed from these techniques [16].

Paper organization. Section 2 points out some preliminaries of this work. Section 3 gives the main result of this paper on the approximation of a set of approximate INDs. Related works are discussed in section 4. We conclude in section 5.

2 Preliminaries

We assume the reader is familiar with basic concepts from relational database theory (see e.g. [19,14]).

An *inclusion dependency* (IND) over a database schema $\mathbf{R}$ is a statement of the form $R_i[X] \subseteq R_j[Y]$, where $R_i, R_j \in \mathbf{R}, X \subseteq R_i, Y \subseteq R_j$ The size (or

arity) of an IND $i = R[X] \subseteq R[Y]$, noted $|i|$ is such that $|i| = |X| = |Y|$. We call *unary inclusion dependency* an IND of size 1.

An inclusion dependency is said to be *satisfied* in a database, if all values of the left hand-side appears in the right-hand side. If I_1 and I_2 are two sets of INDs, I_1 is a *cover* of I_2 if $I_1 \models I_2$ (this notation means that each dependency in I_2 holds in any database satisfying all the dependencies in I_1) and $I_2 \models I_1$.

To evaluate approximate INDs, an error measure called g_3' has been defined in [17]:

$$g_3'(R[X] \subseteq S[Y], \mathbf{d}) = 1 - \frac{max\{|\pi_X(r')| \mid r' \subseteq r, (\mathbf{d} - \{r\}) \cup \{r'\} \models R[X] \subseteq S[Y]\}}{|\pi_X(r)|}$$

Intuitively, g_3' is the proportion of distinct values one has to remove from X to obtain a database $\mathbf{d}'$ such that $\mathbf{d}' \models R[X] \subseteq S[Y]$. An INDs i is said to be approximately satisfied if $g_3'(i)$ is lower than a given threshold.

Given $i = R[X] \subseteq S[Y]$ and $j = R[X'] \subseteq S[Y']$ two INDs over a database schema $\mathbf{R}$, we say that i *generalizes* j (or j *specializes* i), denoted by $i \preceq j$, if $X' =< A_1, ..., A_n >$, $Y' =< B_1, ..., B_n >$, and there exists a set of index $k_1 < ... < k_l \in \{1, ..., n\}$ with $l \leq n$ such that $X =< A_{k_1}, ..., A_{k_l} >$ and $Y =< B_{k_1}, ..., B_{k_l} >$ [20,4]. For example, $R[AC] \subseteq S[DF] \preceq R[ABC] \subseteq S[DEF]$, but $R[AB] \subseteq S[DF] \not\preceq R[ABC] \subseteq S[DEF]$.

Let $\mathbf{d}$ be a database over $\mathbf{R}$, the approximate satisfaction of INDs is monotone with respect to $\preceq$ since $i \preceq j \Rightarrow g_3'(i) \leq g_3'(j)$ [6].

For an precise definition of "problem representable as sets", the reader is referred to the theoretical framework defined in [20]. In [6], we detailed how an isomorphism can be defined between a set of IND under the partial order $\preceq$ and a set of unary INDs under the partial order $\subseteq$. In other words, INDs are said to be *representable as sets*.

Basically, given an IND i, the representation as sets of i is defined by $f(i) = \{j | j \preceq i, |j| = 1\}$, the function f being bijective.

A set I of INDs is *downwards closed* if $\forall i \in I$, we have $j \preceq i \Rightarrow j \in I$. A downwards closed set I can be expressed by its *positive border* or border, denoted by $\mathcal{B}d^+(I)$, which is the set of the most specialized elements in I.

3 Approximating a set of INDs

The data mining problem underlying this work can be stated as follows:

> [`All-ApproxIND` problem] Given a database d and a user-specified threshold ϵ, find the set of approximate INDs i such that $g_3^{'}(i) \leq \epsilon$.

Algorithms do exist to perform this task [4,6] and we assume that such a set of approximate INDs is available.

In this paper, we are only interested in the approximation problem of such a set, denoted by `Approx-All-ApproxIND`, and defined as follows:

[`Approx-All-ApproxIND` problem] Given a downwards closed set I of approximate INDs, find a set S of k INDs approximating I as well as possible.

We only consider as input the whole set I of INDs answering the `All-ApproxIND` problem. Note that the positive border of I could have been considered instead of I itself [11].

We extend the propositions made in [11] in our context: First, a natural way of representing a set of INDs is to consider k INDs from which all more general INDs are generated. Before going into much details, the following notations will be needed: Let i be an IND. The downwards closed set of i, denoted by $\rho(i)$, is defined by $\rho(i) = \{j | j \preceq i\}$.

Let $I_1, \ldots I_k$ be a set of k INDs from I. An approximation of I can be represented with $S(I_1, \ldots I_k)$ (or simply S whenever $I_1, \ldots I_k$ is clear from context) defined by

$$S(I_1, \ldots I_k) = \bigcup_{i=1}^{k} \rho(I_i)$$

$S(I_1, \ldots I_k)$ is said to be a *k-spanned set of INDs.*

Second, we have to decide how to define S with respect to I. Two main alternatives exist: either elements of S have to belong to I or can be chosen outside I. For the problem of approximating a set of approximate INDs, a natural choice seems to be the first alternative, i.e. $S \subseteq I$ or more accurately, $S \subseteq \mathcal{B}d^+(I)$. This choice is justified by the "global structure" of a set I answering the `All-ApproxIND` problem, leading from an interaction property between FD and IND. This point is discussed in much details in the following subsection.

Third, in order to define the approximation made between I and S, we borrow the *coverage measure* $C(S, I)$ [11] defined as the size of the intersection between S and I. Since $S \subseteq I$, maximizing the coverage measure is equivalent to maximazing the size of S.

Describing approximatively the downwards closed set I of approximate INDs with a set S of k INDs can be defined as follows:

- the size of S is equal to k, usually much smaller than the size of $\mathcal{B}d^+(I)$,
- for some $i \in I, \exists j \in S$ such that $i \preceq j$, i.e. S is a succinct representation of I,
- for some $i \in I$, $\nexists j \in S$ such that $i \preceq j$, i.e. some information is lost,

With the assumptions made, the problem `Approx-All-ApproxIND` can now be re-formulated as follows:

Given a downwards closed set I of approximate INDs, find a k-spanned set S of INDs such that $C(S, I)$ is maximized.

Theorem 1. *`Approx-All-ApproxIND` is NP-hard.*

Proof. This problem is shown to be NP-hard for frequent itemsets [11]. Since INDs are representable as sets of unary INDs, there is a bijective function between a set of INDs and a set of set of unary INDs. The transformation is clearly polynomial. Now, if we see unary INDs as items, and set of unary INDs as itemsets, any set of INDs can be transformed into a set of itemsets. In the same way, any set of itemsets can be transformed into a set of unary INDs. The `Approx-All-ApproxIND` problem is thus equivalent to the same problem for frequent itemsets.

Let I be a downwards closed set of approximate INDs and let S^* be an optimal solution for the problem `Approx-All-ApproxIND`.

Theorem 2. *A k-spanned set S of INDs can be found in polynomial time such that*

$$C(S, I) \geq (1 - \frac{1}{e})C(S^*, I)$$

Proof. Since INDs are isomorphic to the set of unary INDs, the problem can be posed in term of set theory. Within the framework of set theory, the problem turns out to be an instance of the Max k-Cover, from which a greedy algorithm does exist and provide a $(1 - \frac{1}{e})$ approximation ratio to Max k-Cover [9]. A solution can be computed with the greedy algorithm and translated back to INDs since the function is bijective.

To sum up, despite the NP-hardness result, polynomial-time approximation algorithms can be used effectively to answer our `Approx-All-ApproxIND` problem.

3.1 A heuristic

Since spanners of a set I of INDs are chosen within the border of I, one may be tempted to select the k first IND whose arity is larger in the border of I.

We now point out that, in practice, a set I of (approximatively) satisfied INDs has a particular structure. This consideration leads from a well-known interaction between INDs and functional dependencies, given in Table 1.

$$\left.\begin{array}{c} R[XY] \subseteq S[UV] \\ R[XZ] \subseteq S[UW] \\ S : U \rightarrow V \end{array}\right| \Rightarrow R[XYZ] \subseteq S[UVW]$$

Table 1. An interaction between FD and IND

Intuitively, consider two INDs i and j of large arity. Let U be the set of common attributes in the right-hand sides of i and j. The more large $|U|$ is, the more U is likely to be a left hand side of functional dependencies. In other words, a new IND is likely to be satisfied from the property given in Table 1, and i and j are not anymore in the border of I.

As already discussed, this reasoning has a very strong impact on the global structure of the downwards closed set I of INDs to be approximated. It cannot be compared with the global structure of a downwards closed set of frequent sets where no similar property (of that given in Table 1) does exist. We derive two points from this consideration:

- There is no need to define S outside I. Indeed, an IND $i \notin I$ would be interesting if it covers a great number of elements of I and only a few number of elements not in I. But we argue that, from property in Table 1, i has great chances to belong to I in this case.
- We suggest the following very simple heuristic:
  Let S_k be a subset of I made up of the k largest INDs of I
  Such a set is straightforward to compute from I and offers an approximate solution of I.

Hopelessly, the solution S_k cannot be compared with other solutions obtained so far. Nevertheless, due to the property 1, we conjecture that the solution S_k is also a good approximation of I.

Remark also that this heuristic gives the best solution, i.e. $S_k = S^*$, whenever the border of I is made up of "disjoint" INDs, i.e. no attribute appears in more than one IND of $\mathcal{B}d^+(I)$. Nevertheless, this case seems to be a very constrained one in practice.

Another justification comes from experiments performed on two data mining problems: IND discovery [6] and maximal frequent item set discovery [7]. From our experimental tests, we have studied the *global structure* by doing a quantitative analysis of the positive and the negative borders of the result. In all cases, the global structure was quite different, justifying both our choice to define S from I and our heuristic.

4 Related works

In [11], authors introduce a framework for approximating a collection of frequent sets by a set of k sets. They consider two cases for choosing their approximation: either from the collection, or *outside* the collection. Moreover, they extend their results by considering the positive border of their collection of frequent sets instead of the whole collection.

Another point of view for approximating frequent sets consists in computing the *top-k most frequent sets*, as proposed in [8] for frequent closed sets.

In the setting of approximate IND discovery, [13] proposed a set of heuristics. The basic idea is to improve the pruning by considering some criteria from which interestingness of INDs can be evaluated, for example using the number of distinct values. In [17], approximation is seen with respect to SQL workloads, i.e. join path expressions reveal "interesting" INDs. These approaches are rather empirical though.

5 Conclusion and perspectives

We studied in this paper approximating techniques of a set of approximate inclusion dependencies in the following setting: what are the k INDs that best approximate the whole set of INDs ? From results given in [11] for frequent itemsets, we focus on two of them : 1) the problem of finding the best approximation spanned by k sets is NP-hard and 2) an approximation can be computed in polynomial time within a factor of $(1 - \frac{1}{e})$ with respect to the best solution. We mainly show in this paper how these results may apply for approximating a set of INDs with k INDs. These results follow from the fact that INDs are "representable as sets", i.e. the search space of IND is isomorphic to a subset lattice of some finite set.
We proposed also an heuristic exploiting the global structure of a set of approximate INDs satisfied in a database. Since no formal comparison can be made with respect to the best solution as we did before, we plan to do in future work experiments in order to evaluate the quality of this heuristic.

An interesting corollary may be also easily deduced since any problem representable as sets can indeed re-used the propositions made in [11] as for example functional dependencies or learning boolean functions.

¿From a data mining point of view, algorithms for discovering approximation of discovered patterns from the data have to be designed, instead of discovering first the set of all patterns and then applying approximation schemes.