SEARCHING MULTIPLE DATABASES FOR INTERESTING COMPLEXES

JUN YAO and HUAN LIU
Department of Information Systems and Computer Science
National University of Singapore
Kent Ridge, Singapore 119260
{yaojun, liuh}@iscs.nus.sg

When many databases are available for knowledge discovery, one pressing problem is which databases have interesting knowledge. To solve this problem, users can pose a query when investigating databases and search for interesting information. In order to find interesting information and avoid unnecessary search in irrelevant databases, interestingness measures and efficient search methods are needed. In this paper, we propose a method based on a statistical measure by which some relevant databases can be detected. Knowledge, in the form of complexes, can be extracted which contains potentially interesting information. Users can identify importance and relevance of each database according to extracted knowledge. A heuristic function is applied to make the search more efficient. The extension of this method is also discussed.

1 Introduction

Large sized database, diverse relations of data and many sources from which data is collected are obstacles for users to find useful knowledge. Information among databases is not always useful for particular applications because of data redundancy and irrelevance, so efficiently inferring useful knowledge from the multiple databases is important for users. When many databases are available for knowledge discovery, one pressing issue is which databases have interesting knowledge for a particular application. To solve the problem, the process of knowledge discovery could be incorporated with a user's intentions or interests. Posing a query is one way of embedding a user's intents and interests in the process of knowledge discovery. Some methods of posing queries for knowledge discovery are proposed [1 2 3 4]. A user's interest, intuition or prior knowledge can be conveyed by means of such queries. In the case of multiple databases, problems arise: how to efficiently search for interesting knowledge in many databases by posing queries.

In this paper, we propose an approach to search for interesting knowledge in multiple databases according to a user's query. This process can be viewed as retrieval of interesting information from many databases. There are different kinds of ways to pose queries, we don't discuss how queries are made in detail, we only regard a query as a constrain which specifies a subset in all data. We focus on what interesting knowledge about this subset can be derived. A query, in our discussion, is the conditional part of a rule, the knowledge derived from multiple databases is the conclusion part of a rule, called *complexes*. These complexes should represent special or interesting knowledge in the data. A conclusion part or complex can be a conjunction of several attribute values (high order complex) as well as a single attribute value (low order complex). A large number of combinations of attribute values prohibits efficient search of high order complexes in many databases. A heuristic function is presented here to enable potentially interesting complexes to be found in relevant databases and avoid unnecessary search of high order complexes in irrelevant databases.

2 Notion of interestingness

Knowledge Discovery in Databases (KDD) targets at finding useful knowledge from large real world databases. Mining valuable knowledge is an important issue in KDD. The measurement of knowledge's value is usually determined by a user's specific application needs. *Interestingness* is usually taken as an overall measure in KDD. There is a lot of literature [5] [6] [7] [8] about the interestingness issue. The definition of interestingness mainly includes: validity, novelty, usefulness and ease of understanding [5]. These factors reflect a user's requirements on the whole.

In many situations, although discovered knowledge is highly understandable, valid and consistent with data, it is useless because it tells us nothing but common phenomena in the data. Besides the validity of knowledge, users are more concerned about what surprising variations are in data. Hence, our interestingness measure essentially is a statistical measure or so-called discrepancy measure [9], by which users can detect the deviations that indicate some outcomes in the subset covered by a query differing from those in the whole dataset. Deviation is a powerful and simple way of identifying interesting knowledge in data [7], because it indicates how the distribution of the observed events in certain subset differs from a reference level. The expected value according to the whole data (the average value) or normative value according to prior knowledge can be chosen as the reference level. We choose the expected value as our reference level. By detecting deviations in many database, users obtain complexes derived from multiple databases at the end of discovery process, each with a significance value by which users can also analyse and identify the relevance and importance of certain databases for their particular applications.

A production rule, a comprehensible representation of knowledge, is in the form as:

$$\text{IF} \quad Q \quad \text{THEN} \quad R$$

in which the right-hand side R is a *complex* rather than a class label in our discussion. Both Q and R are conjunctions of *selectors* that are in the form of: *attr* OP *value*. *attr* is the name of an attribute in a database, *value* is the possible values of an attribute *attr* and OP $\in (=, <, >, \leq, \geq)$ is an operator. We call Q the condition part and R the conclusion part.

As the condition part of a rule is specified by users, the task of our algorithm is to search for complexes, i.e., the conclusion parts that satisfy our criteria. In order to identify valid and novel complexes in data, we define the measure of significance for a complex as:

$$S = P(R|Q) log \frac{P(R|Q)}{P(R)} \qquad (1)$$

where P(R) is the probability of the event R occurring in a database and $P(R|Q)$ is the conditional probability of R occurring in the case of Q occurring.

The value of $P(R|Q)/P(R)$ shows the degree of the deviation between the probability of the event R occurring under the condition Q and the probability of R occurring in the whole database. If the nearer $P(R|Q)/P(R)$ is 1, i.e., the nearer S is 0, the closer the distribution of R in Q conforms to that of R in the whole database and the less novelty of the event R given Q; The larger $P(R|Q)/P(R)$ (i.e. $P(R|Q)/P(R) \gg 1$) is, the more likely the event R has a genuine correlation with Q; When $P(R|Q)/P(R) < 1$, the value of $\log P(R|Q)/P(R)$ is negative. The nearer $P(R|Q)/P(R)$ is 0, the more likely the two events R and Q are exclusive. In this paper, the complex which value $0 \leq P(R|Q)/P(R) < 1$ is discarded, we focus on the events related to Q. $P(R|Q)$ is used as the weight. Thus trivial outcomes in databases have low significance values.

$P(R|Q)$ also represents the accuracy or validity of a rule. The lower bound value (a threshold) can be determined by a user. A complex is rejected if its accuracy value is below the threshold. When the search begins, the conclusion part is an empty complex (a conjunction of zero selector). During the search, this part is specialized by intersections of the selectors. If the accuracy value of a specialized complex does not pass the lower bound, then the search on this node is pruned.

The similar significance measure is adopted in CN2 [10] called *likelihood ratio statistic* [9]. The significance values in our method and in CN2 both reflect the degree of the correlation between a complex and a class (in CN2) or a complex and the condition part of a query (in our method). The difference lies in that the significance value in our method is used to identify the novelty or unexpectness of a complex covered by a query at certain specified accuracy; the significance value in CN2 is used to evaluate the reliability of a complex classifying a class with high accuracy, i.e., high accuracy on classifying training data is not just due to chance.

3 Algorithm

Algorithm: Finding Interesting Complexes in Multiple Databases
Input: The condition part of a rule (A query)
    Databases $D_1, .., D_n$
Output: A set of Complexes with $significance$ values
define: $Complexlist_i$ is a set of complexes found in D_i
define: $Selectors_i$ is the set of all possible selectors covered by the query in D_i
Main procedure:
  For each D_i
    If D_i has the same domain as one in the query
      Then call **function**(search for complexes in D_i)
      $Complexlist_i$=a set complexes found in D_i
    Else next database
function(search for complexes in D_i)
 { For each $Selectors_i$ calculate accuracy P and significance S
  Choose the selectors that have high significance values as $star$
  (The accuracy values of selected selectors should exceed the threshold)
  (The significance values of the selectors in $star > 0$)
  (The number of the selectors in $star$ $\leq$user-defined maximum number)
 Do while any new complexes are added to $star$
  Let $newstar$ be the set $\{x \wedge y | x \in$ the new complexes in $star$, y$\in Selectors_i\}$
  If the complexes in $newstar$ can't satisfy the criteria
    Then remove the complexes out of $newstar$
  Add $newstar$ to $star$
  Remove the least significant complex from $star$ until
  The number of the complexes in $star$ $\leq$user-defined maximum number
 return $Complexlist_i = star\}$

Figure 1: The algorithm of knowledge discovery

In our discussion, the search space in one database consists of all possible combinations of the selectors in a database. Assuming a database has n attributes besides the attributes whose names are the same as those in a query and each attribute of the examples covered by a query has m possible values, the number of possible combinations of the selectors is $(m+1)^n - 1$. When the number of attributes and the number of attribute values become large, heuristic functions are needed for efficient search. Our algorithm only searches a part of the search space of a database and tries to find potentially novel complexes. When there are no more complexes found in a database, we stop the search of this database. The algorithm is presented in Figure 1, which mainly consists of three steps as follows:

1. At the beginning, we enumerate all possible selectors in the database, then choose the most promising selectors as the initial complexes to form the *star* according to our accuracy and significance criteria. The number of the complexes (N_{max}) in the star is defined by a user and the significance values of the selected complexes must be larger than zero. So the search strategy of the algorithm is actually a beam search with the width of N_{max}.

2. Implementing combinations of the selectors is to intersect the complexes in current star with the set of all possible selectors. The new complexes generated by intersection form the *newstar*. "Good" complexes in the newstar should be selected and added to the star for replacing those not so "good" complexes compared with the new complexes. In this case, using the criteria of the step 1 to evaluate the complexes is insufficient, because intersecting the selector (or complex) of high significance value with the selector of low significance value could yields a complex of high significance. Low significance selectors usually couldn't improve the significance of a complex.

   We call the components of an intersection *predecessors* and the result of an intersection a *descendant*. If the accuracy values of predecessors and a descendant are the same, it is obvious that the significance value of the descendant is not less than those of the predecessors according to our significance definition. In order to eliminate the selectors that don't improve the significance of complexes, we define the *significance gain* to evaluate the novelty of a new complex. The complex which significance value satisfies the following inequality and its accuracy value passes the threshold is selected.

   $$S_{new} - S_{max} > S_{min}$$

   Where S_{new} means the significance of a new complex;
   S_{max}=Max(the significance values of the predecessors of a new complex);

S_{min} is the significance value of the complex (C_{min}) which has the least value in the old star.

The measure of the significance gain actually evaluates the degree of improvement of the descendant's novelty in respect to the predecessors'. If the significance gain value of a new complex is larger than the significance value of C_{min}, this means the combination of a complex with a selector is more novel than the complex C_{min}.

3. Sort the new complexes and the complexes in the star according to the values of significance and choose the high ranking complexes, the number of the complexes in the star $\leq$user-defined number N_{max}. Repeat step 2 until no more complexes can be found, the search in a database is finished. At the end of a search in a database, we get a set of rules which orders are sorted according to descending significance value. If no complexes of a database satisfy the criteria at step 1, that means the database is irrelevant to the user's interests and we stop searching this database. Hence this method searches many databases efficiently.

4 Case study

In this section, we will illustrate how the algorithm works in two small databases. These databases are about people's diet habits, we suppose they are obtained from different sources. A user can find some complexes that he is interested by the way of posing a query to retrieve these databases. For example, when a user wants to know some diet habits about Chinese people, i.e., particularity of Chinese diet habits or difference from people of other countries. A query, in the form of "If nationality=Chinese Then ?", is presented. In this experiment, we specify the threshold value as 65%, the maximum number of complexes in the star as 3.

In Table 1 the query covers six examples, all possible selectors about Chinese and the significance values (S) and accuracy values (P) are:

(1)main food=rice; S=0.737; P=100%
(2)regular eating times=3; S=0; P=100%
(3)favorite drink=tea; S=0.697; P=83.33%
(4)favorite drink=cola; S=-0.113; P=16.67%
(5)vegetarian=yes; S=-0.044; P=16.67%
(6)vegetarian=no; S=0.049; P=83.33%

According to the criteria of the algorithm ($S > 0$ and $P \geq 65\%$), selectors (1),(3),(6) are selected as the initial complexes in the star. The minimum

Table 1: Diet habit I

ID	Nationality	main food	regular eating times	drink	vegetarian
950351	Chinese	rice	3	tea	no
950301	Chinese	rice	3	tea	no
950282	Chinese	rice	3	tea	no
940112	Chinese	rice	3	tea	yes
940226	Chinese	rice	3	cola	no
950578	Chinese	rice	3	tea	no
950122	American	bread	3	cola	no
940227	American	rice	3	cola	yes
940121	American	bread	3	tea	no
950612	American	bread	3	coffee	no
940109	Indian	bread	3	tea	no
950123	Indian	rice	3	cola	no
950348	Indian	bread	3	coffee	no
940247	Indian	rice	3	coffee	no
950312	Indian	bread	3	coffee	yes

Table 2: Diet habit II

ID	Nationality	alcohol	on diet	between-meal	favorite non-veg
1122	Chinese	never	no	sometimes	pork
1078	Chinese	never	no	seldom	pork
1690	Chinese	sometimes	no	seldom	chicken
1432	Chinese	often	no	often	chicken
1364	Chinese	sometimes	no	sometimes	beef
1574	Chinese	sometimes	no	sometimes	egg
2345	American	never	no	seldom	beef
2434	American	often	no	sometimes	beef
2566	American	sometimes	no	sometimes	chicken
2567	American	sometimes	no	often	pork
3545	Indian	often	no	sometimes	fish
3256	Indian	sometimes	no	seldom	pork
3854	Indian	never	no	sometimes	chicken
3976	Indian	sometimes	no	sometimes	fish
3431	Indian	never	no	sometimes	beef

significance value of the complexes in the star is 0.049. After intersecting of (1),(3),(6) with other selectors respectively, we get the new complexes that pass the threshold value as:

(1)$\wedge$(2); S=0.737; P=100%; $S_{gain} = 0.737 - 0.737 < 0.049$;

(1)$\wedge$(3); S=1.102; P=83.33%; $S_{gain} = 1.102 - 0.737 > 0.049$;

(1)$\wedge$(6); S=0.697; P=83.33%; $S_{gain} = 0.697 - 0.737 < 0.049$;

(2)$\wedge$(3); S=0.697; P=83.33%; $S_{gain} = 0.697 - 0.697 < 0.049$;

(2)$\wedge$(6); S=0.049; P=83.33%; $S_{gain} = 0.049 - 0.049 < 0.049$;

(3)$\wedge$(6); S=0.491; P=66.67%; $S_{gain} = 0.491 - 0.697 < 0.049$;

From the complexes above, only the complex (1)$\wedge$(3) satisfies the criterion of the significance gain, then add this complex to the star and remove the complex (6) whose significance value is the least one. Because no complex satisfies the criteria after intersecting of the new complex in the star with the selectors again, the search in Table 1 stops. As the result, we obtain three rules:

1.If a person is Chinese Then his main food is rice and favorite drink is tea. S=1.102; P=83.33%

2.If a person is Chinese Then his main food is rice. S=0.737; P=100%

3.If a person is Chinese Then his favorite drink is tea. S=0.697; P=83.33%

From Table 1, the novelty of complexes means the particularity of diet habits, the first rule with the highest significance value most clearly shows the particularity of diet habit of Chinese people.

In Table 2, the selectors covered by the query and their significance and accuracy values are as follows:

drink alcohol=never; S=0; P=33.33%;

drink alcohol=sometimes; S=0.05; P=50%;

drink alcohol=often; S=0.044; P=16.67%;

on diet=no; S=0; P=100%;

take between-meal=sometimes; S=-0.131; P=50%;

take between-meal=seldom; S=0.107; P=33.33%;

take between-meal=often; S=0.054; P=16.67%;

favorite non-veg food=pork; S=0.107; P=33.33%;

favorite non-veg food=chicken; S=0.107; P=33.33%;

favorite non-veg food=beef; S=-0.113; P=16.67%;

favorite non-veg food=egg; S=0.22; P=16.67%;

In Table 2, there is only one selector's accuracy value pass the threshold (whether on diet=no), but its significance value is zero. So no complex satisfies the criteria at the first step, the search in this table stops. From the two tables, we can intuitively find that there exist some regularities of diet habits among people of different nationality in Table 1, while no particular in Table 2.

5 Related work

Several methods of knowledge discovery in multiple databases have been proposed [1 2]. In the paper [1], database techniques are used for knowledge discovery in multiple databases. When a query q (a SQL-like query in the form of: Select $attribute_i$,..; From $Database_k$,... ; Where $attribute_j = m$,...) is posed, a new table is joined according to a user's intents. The attributes in the Where clause are called *reference attributes*. The attributes in the Select clause are called *target classes*. The new table is decomposed in terms of target classes. The reference attributes are used to check whether the *decomposed table* preserves *characterizations*. A new rule discovered from the table is satisfied by all *positive examples*. If the new table is not decomposed because all the values of the target attribute are the same in the table, this means the table preserves all the reference attribute values. A rule, in the form of "If q Then *target attribute*=value", is obtained. In this method, the derived rule reflects the relation of databases.

Another knowledge discovery method was introduced [2]. The main idea of this research is to apply knowledge discovered from one database to enhance the process of knowledge discovery in other databases. This work includes the incorporation of the *primary key* and the *foreign keys*. The primary key is designated by database designers and uniquely identifies each record in the database. The foreign key occurs in other databases and has a domain equivalent to that of the primary key. The primary key and the foreign key serve as the links across the databases. A rule generated from the database with the primary key is in the form of "If Q Then R". Assuming the examples covered by the condition part Q in the database with the primary key are $\{E_1, E_2, ..E_i\}$. Through the links of the primary key and the foreign key, mapping these examples to another database, we get $\{E_1', E_2', ..E_i'\}$. If $\{E_1', E_2', ..E_i'\}$ satisfy *knowledge association coefficient* criterion (KAC) given by a user on a *target class*. A new rule is obtained.

For example, in a database with a foreign key, if all examples $\{E_1', E_2'...E_i'\}$ and only these examples belong to a *target class*: R′=value. By *deductive generalization*, a new rule (If Q then R′=value) is obtained. The KAC is defined as follows:

$$KAC = coverage \times specificity$$

in which *coverage* is the percentage of the examples in the target class which are covered by Q (i.e. $P(Q|R')$), and *specificity* is the percentage of the examples covered by Q and belonging to the target class R′ (i.e. $P(R'|Q)$).

At the example above, both coverage and specificity values are 100%. This method is actually equivalent to posing queries according to existing rules (i.e.

the condition part of the rule), which specifies the subset of the data in the next database by the links of the primary key and the foreign key, and finding the new conclusion parts satisfying the KAC criterion.

The earlier works have some limitations for employing them in practical applications:

1. The two criteria (KAC [2] and whether a decomposed table preserves the characterization [1]) both have bias toward the complexes that fall in a target class as far as possible, in other words they only consider the distributions of complexes in target classes rather than distributions of complexes both in target classes and in all data. Those complexes that do not satisfy the criterion perfectly but still reflect some special attribute value distributions among data may be overlooked.

2. The ability of the two methods to handle noise is not desirable, especially the first method is sensitive to data with even little noise.

3. The *target class* in above two methods is only a single attribute value, but in many situations, combinations of these attribute values often imply some useful information.

Our approach of defining the significance of a complex is capable of finding not only the single selector complexes with a high specificity value but also those potentially interesting complexes with relatively low specificity values and the complexes composed of several selectors. The calculation of a complex's significance value is based on the whole dataset, noisy data does not have large effect on calculation.

In the case study, we consider the simple situation that the databases have the same domain as that of the query. The algorithm can be extended to the situation in which databases have the links by the primary key and the foreign key. This method can make use of the relations among different databases for knowledge discovery.

A user can also pose a query in fuzzy term when the attribute values in a query are continuous values. In this case, the subset of the examples covered by a query is a fuzzy set rather than a crisp set. A user specifies the membership function of the fuzzy set, thus the membership function is utilized in the process of finding interesting complexes. This method is useful because the expression of people's thought is often in fuzzy term.

According to the probability theory for fuzzy sets [11] [12] and the theory of statistics, we modify the calculations in formula (1):

$P(R|Q) = P(Q|R) \times P(R)/P(Q),$

$P(Q) = \sum_{x \in \mathcal{D}} \mu_Q(x)p(x),$

$P(Q|R) = \sum_{x \in \mathcal{D}} \mu_Q(x)p(x|R),$

where x is an example in a database, $\mu_Q(x)$ is the membership of example x in fuzzy set Q, the universe of discourse $\mathcal{D}$ includes all examples in a database. $p(x)$ is the probability of example x occurring in the whole database, $p(x|R)$ is the conditional probability of example x occurring in the case of R occurring.

6 Discussion and Conclusion

In practical applications, there is a problem called heterogeneity of multiple databases such as homonyms (using same name for different attributes) and synonyms (using different names for the same attributes). People have provided some techniques called *schema integration* [13] to resolve this kind of schema conflicts. It is not our topic to discuss the schema integration in this paper. So besides utilization of the links of primary keys and foreign keys by which we can explore through many relational databases, we assume that databases have the uniform schema, i.e., the same attributes in different databases have the same name.

A weakness of our method is that it is based on the assumption that data is sufficient to show particularity of the observed event distribution in a subset with respect to the whole set. For example, the condition part of a user's query only covers a few examples in one database and discovered complexes have high values of significance and accuracy, while in another database, the query covers many examples but discovered complexes have relatively low values of significance and accuracy. In the first case, a complex's high accuracy on the training data does not mean high accuracy on future data from a statistical viewpoint, because the calculation of accuracy of a complex is based on a few examples.

There are two alternatives to deal with this problem. The first one is to use estimating accuracy instead of the accuracy in the training dataset as the validity criterion. We can use the *far more pessimistic estimate* method adopted in C4.5 [14] to calculate the estimating accuracy of a complex. Another one is to add another weight P(R) to the formula (1):

$$S = P(R|Q) \times P(R) \times log \frac{P(R|Q)}{P(Q)}$$

$$= \quad P(R \wedge Q) \times log \frac{P(R|Q)}{P(Q)}$$

Thus, the weight before log becomes the joint probability of R and Q. If the number of examples in a database that satisfy Q is too small, the significance of the complexes found in this database are also relatively small. The results of the case study of section 4 is the same after applying the modified significance measure.

Till now selecting the most appropriate parameter mainly depends on the practical application and a user's experience. In the case study of section 4, if we choose smaller threshold value, we can still get the same order of result rules, because complexes with the higher accuracy are reinforced according to definition of the significance. The order of rules sorted by descending significance value is not very sensitive to the choice of parameter, although in some cases adjusting these values may lead to some different results. Another problem is the overlapping of complexes' information. Although information of some complexes is overlapping, we also keep them in the result list because sometimes their accuracy values are different. It's also our future work to derive more concise knowledge which is easy for users to analyse.

Acknowledgements

We gratefully thank Dr Hongjun Lu and Manoranjan Dash for their valuable comments on earlier drafts of this paper.

Reference

1. J. P. Yoon and L. Kerschberg. A framework for knowledge discovery and evolution in databases. In *IEEE Transactions on Knowledge and Data Engineering 5(6)*, pages 973–979, December 1993.
2. J. S. Ribeiro, K. A. Kaufman, and L. Kerschberg. Knowledge discovery from multiple databases. In *Proceedings of KDD-95*, pages 240–245, 1995.
3. B. Liu, W. Hsu, H. Y. Lee, and L. F. Mun. Tuple-level analysis for identification of interesting patterns. In *Technical Report of Dept. ISCS. National University of Singapore*, 1996.
4. W. M. Shen, K. L. Ong, B. Mitbander, and C. Zaniolo. Metaqueires for data ming. In *Advances in Knowledge Discovery and Data Mining*, pages 376–398. AAAI Press / The MIT Press, 1996.

5. U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI Press / The MIT Press, 1996.
6. A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 275–281, 1995.
7. G. Piatetsky-Shapiro C. J. Matheus and D.McNeill. Selecting and reporting what is interesting. In *Advances in Knowledge Discovery and Data Mining*, pages 495–514. AAAI Press / The MIT Press, 1996.
8. B. Liu and W. Hsu. Post-analysis of learned rules. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence AAAI-96*, pages 828–834. AAAI press/ The MIT press, August 1996.
9. J. G. Kalbfleisch. *Probability and Statstical Inference.* Springer-Verlag New York Inc., 1979.
10. P. Clark and T. Niblett. The CN2 induction algorithm. In *Machine Learning 3*, pages 261–283, 1989.
11. A. Kandel. *Fuzzy Mathematical Techniques with Applications.* Addison Wesley Publishing Company, 1986.
12. H. Bandemer and S. Gottwald. *Fuzzy sets, Fuzzy Logic, Fuzzy method with Applications.* John Wiley Sons Ltd, 1993.
13. M. Lenzerin C. Batini and S. B. Navathe. A comparitive analysis of methodologies for database schema intergration. In *ACM Computing Surveys 18(4)*, pages 323–364, 1986.
14. J. R. Quinlan. *C4.5: Program for machine learning.* Morgan Kaufmann, 1992.