A Relational Approach for Discovering Frequent Patterns with Disjunctions

Corrado Loglisci, Michelangelo Ceci, and Donato Malerba

Department of Computer Science, University of Bari
Via E.Orabona 4, 70126, Bari-Italy
{loglisci,ceci,malerba}@di.uniba.it

Abstract. Traditional pattern discovery approaches permit to identify frequent patterns expressed in form of conjunctions of items and represent their frequent co-occurrences. Although such approaches have been proved to be effective in descriptive knowledge discovery tasks, they can miss interesting combinations of items which do not necessarily occur together. To avoid this limitation, we propose a method for discovering interesting patterns that consider disjunctions of items that, otherwise, would be pruned in the search. The method works in the relational data mining setting and conserves anti-monotonicity properties that permit to prune the search. Disjunctions are obtained by joining relations which can simultaneously or alternatively occur, namely relations deemed similar in the applicative domain. Experiments and comparisons prove the viability of the proposed approach.

1 Introduction

Discovery of frequent patterns in large collections of transactions or tuples has become one of the broadly investigated topics in data mining[1,4]. Patterns represent statistical regularities of co-occurrences (expressed as conjunctions) of the items present in the transactions. The most interesting patterns are those that express conjunctions which occur in at least a user-defined number of transactions. Typically, such conjunctions are obtained by the intersection of the transactions in which the items occur under the assumption that the items occur independently of each other. This poses some limitations to the mining process and, in particular, leaves unexplored two potentialities of the pattern discovery: i) discovering interesting patterns when items are not present in a sufficient number of transactions, and ii) discovering forms of relationships among items different from the classical conjunctions. Indeed, the two potentialities are not independent each other since the discovery of patterns including other relationships between items may lead to discover patterns that otherwise would be discarded.

Although traditional frequent patterns discovery approaches are based on the items which co-occur, other forms of relationships among items have been actually investigated in the literature. In particular, some works propose to accommodate a domain-dependent taxonomy over the items in the mining process in

T.B. Pedersen, M.K. Mohania, and A M. Tjoa (Eds.): DaWaK 2010, LNCS 6263, pp. 263–274, 2010.

addition to the classical conjunction. This allows to consider the generalization relationship (*is-a relationship*) among the items[13]. The presence of items at higher levels of the taxonomy in the patterns implies the presence of the items at lower levels related with the former through is-a relationships. For instance, given a complete taxonomy for which, milk *is-a* food, coffee *is-a* food, fruit juice *is-a* food, the pattern $\langle food, soap\rangle$ can be interpreted as the pattern $\langle (milk \vee fruitjuice \vee coffee), soap\rangle$ where the occurrence of $food$ implies the occurrence of at least one among $milk$, $bread$ or $fruitjuice$, that is $food = (milk \vee fruitjuice \vee coffee)$. The accommodation of a taxonomy thus allows to represent relationships among items in the form of fixed disjunctions.

Independently from the accommodation of a domain taxonomy in the mining process, the discovery of patterns with items in disjunction has been already investigated in the literature with approaches that permit to mine disjunctive association rules from transactions or tuples [10][12]. In [10], the authors provide a statistical framework based on a set operations (union and intersection) among transaction sets to identify itemsets (called contexts) that can potentially contain disjunctions. Then, these contexts are combined and explored to generate a preliminary set of disjunctive rules. Finally, the application of propositional logic techniques on this set allows to infer rules with items related by inclusive logical disjunction and exclusive logical disjunction. Differently, in [12], the authors extend traditional algorithms to mine associations among item groups formed by items in disjunction. Each group is generated by aggregating items on the basis of their conceptual distance. The items are accommodated in a weighted directed graph provided as background information, whereas the conceptual distance between two items is expressed as the weight of their relative edge. The conceptual distance is thus exploited to aggregate two rules, which present conceptually close items, into only one rule: the final rule will incorporate a group of close items in relationship of disjunction and thus it will be more frequent.

Although the first approach allows to discover disjunctive patterns without requiring background information about the items, it could join unrelated items (e.g., $\langle milk \vee jackets\rangle$) and then produce rules that are difficult to understand and which do not exploit the potentialities of the disjunction of representing the occurrence of at least one between two related events (e.g., $\langle milk \vee coffee\rangle$). An important common aspect of both approaches is that they work on tuples, namely on items represented in form of attribute-value pairs which lead to consider the disjunction only among the discrete, categorical or taxonomic values. Although simple and reasonably more effective, this representation, also known as *propositional*, can turn out to be too restrictive in applications where data are naturally complex, and moreover, trasforming such data in tuples could lead to information loss. Several studies in the literature have proved that in those cases resorting to the *relational* representation [6] permits to directly deal with the complex structure of data, to conduct a realistic investigation which distinguishes the main subjects of analysis from other subjects as well as to represent their interaction. Examples of such subjects can be found in spatial analysis where the location and the extension of spatial objects define spatial

relations, such as those topological (e.g., the region A is contained in the region B - $contained\_in(A, B)$), and spatial *properties*, such as those geometrical (e.g., shape of a region - $rectangle\_shape(A)$) [7]. Existing approaches to disjunctive patterns discovery do not consider complex data, and, in particular, they analyze neither possible interactions among them nor the sets of possible descriptive properties.

In this paper we propose a relational data mining approach for discovering frequent patterns that include disjunctions. Patterns are represented in terms of atoms [3]. The approach allows to mine frequent patterns with disjunctions among atoms that can express relations (e.g., $contained\_in(A, B) \vee overlaps(A, B)$) or properties ($(rectangle\_shape(A) \vee square\_shape(A))$ of the analyzed data. It extends an existing logic-based method for conjunctive pattern mining [8] to the discovery of disjunctive patterns, where disjunctions are generated among *similar* relations or properties. Similarity between relations or properties is defined in the user defined background knowledge in form of conceptual distance. The approach takes advantage of the representation and reasoning techniques developed in the field of inductive logic programming (ILP). In particular, the expressive power of logic formalism is profitably used to represent relations, properties and background knowledge in the natural form of *n-ary* logic predicates. This way of using the disjunction permits to combine the occurrences of the involved relations in order to produce patterns with higher frequency, that, potentially, can be more interesting.

The paper is organized as follows. In the next section, motivation and overview of the proposed approach are presented. In Section 3, the approach is presented in detail. In Section 4 experimental results on real world data are reported. Finally, conclusions are drawn and future works are presented.

2 Motivation and Overview of the Approach

The motivation behind the usage of disjunctive forms is that the set of patterns discovered with traditional approaches strongly depends on frequency-based thresholds (e.g., support, confidence, lift) so, when these assume high values, many interesting patterns are missed: conjunctions of atoms, for which the considered statistical measure does not exceed the minimum threshold, are ignored. The introduction of the disjunctive forms would permit to include the atoms which occur simultaneously with or alternatively to other atoms with the effect of increasing the values of the considered measures associated to the patterns. For instance, by supposing that the atom $overlaps(A, B)$ may occur also when $contained\_in(A, B)$ does not occur, the pattern $\langle district(A), (contained\_in(A, B) \vee overlaps(A, B)), marketplace(B)\rangle$ might be frequent while both $\langle district(A), contained\_in(A, B), marketplace(B\rangle$ and $\langle district(A), overlaps(A, B), marketplace(B\rangle$ might not be frequent.

This advocates the starting point of our approach, which is that of considering infrequent conjunctive patterns. These patterns are re-evaluated and extended to the disjunctive form by inserting disjunctions which involve atoms already present in the patterns. Disjunctions are created among atoms which are

semantically related in the application domain. The semantic relatedness is intended as background knowledge on the atoms and permits us to numerically quantify the dissimilarity or conceptual distance between atoms. It guarantees that meaningful disjunctions are created. In this work we exploit the ILP system SPADA [8] to identify infrequent conjunctive patterns, but this does not exclude the possibility of using other methods for mining infrequent relational patterns in the initial processing step.

The proposed approach follows a three-stepped procedure. First, it extracts the infrequent conjunctive patterns which can be considered in disjunctive patterns. In particular, the patterns whose frequency is lower than the classical minimum threshold but exceeds a new ad-hoc threshold are selected. These thresholds determine therefore the set of patterns to be extended to the disjunctive form. Second, by following the main intuition proposed in [12], background knowledge is accommodated to exploit the information on the dissimilarity among the atoms in the process of generation of disjunctive patterns. Third, disjunctive patterns are produced by iteratively integrating disjunctions into the patterns by means of a pair-wise joining. The final result consists of patterns, in form of conjunctions of disjunctions of atoms, whose frequency is greater than the traditional minimum threshold. For instance, given the patterns $P_1 : \langle district(A), contained\_in(A,B), marketplace(B)\rangle$, $P_2 : \langle district(A), overlaps(A,B), marketplace(B)\rangle$ and let $contained\_in(\cdot,\cdot)$ and $overlaps(\cdot,\cdot)$ be two "similar" atoms according to the background knowledge, P_1 and P_2 can be joined in $\langle district(A), (contained\_in(A,B) \vee overlaps(A,B)), marketplace(B)\rangle$.

Working in the relational setting adds additional sources of complexity to the problem of joining patterns due to the *linkedness* property [9]. In fact, in the relational representation atoms in a pattern are dependent each other due to the presence of variables (differently from the items in the propositional representation [12]). In this work, patterns to be joined should differ in only one atom (if the atoms are similar) and share the remaining atoms up to a redenomination of variables. For instance, consider the patterns $P_1 : \langle district(A), contained\_in(A,B), crossed\_by(A,C), marketplace(B)\rangle, P_2 : \langle district(A), crossed\_by(A,B), overlaps(A,C), marketplace(C)\rangle$. The pattern $\langle district(A), (contained\_in(A,B) \vee overlaps(A,B)), crossed\_by(A,C), marketplace(B)\rangle$ can be extracted since B in $contained\_in(A,B)$ is involved in $marketplace(B)$ of the first pattern, as well as C in $overlaps(A,C)$ is involved in $marketplace(C)$ of the second pattern.

3 Mining Disjunctive Relational Patterns

Before formally defining the problem we face in this work, some notions are necessary. In the relational setting, when handling complex data, different roles can be played by different *sorts* of data. In our formulation complex data are distinguished into target objects of analysis (TO) and non-target objects of analysis (NTO). The former are data on which patterns are enumerated and contribute to compute the frequency of a pattern, while the latter contribute to define the former and they can be involved in a pattern. We denote the set of TO

as S and the sets of NTO by means of the sets R_k $(1 \leq k \leq M)$, where M is the number of sorts of data that are not considered to be TO. NTOs, belonging to a set R_k, can be organized hierarchically according to a user defined taxonomy. Target objects and non-target objects are represented in Datalog language [3] as ground atoms and populate the extensional part D_E of a deductive database D. A ground atom is an n-ary logic predicate symbol applied to n constants.

Some predicate symbols are introduced in order to express both properties and relationships of TO and NTO. They can be categorized into four classes: 1) *key predicate* identifies the TO in D_E (e.g., in the examples above, *district(·)*); 2) property predicates are binary predicates which define the values taken by an attribute of a TO or of an NTO; 3) structural predicates are binary predicates which relate NTO as well as TO with others NTO (e.g., in the examples above, *contained_in*$(\cdot,\cdot)$); 4) *is_a* predicate is a binary taxonomic predicate which associates NTO with a symbol contained in the user defined taxonomy.

The intensional part D_I of the deductive database D includes the definition of the domain knowledge that permits us to express the dissimilarity among atoms in the form of *Datalog* weighted edges of a graph. An example of the Datalog weighted edge is the following:

external_touch_to - (crosses - 0.88)

It states that the dissimilarity between the relationships *external_touch_to(·,·)* and *crosses(·,·)* is 0.88. More generally, it represents an undirected edge e between two vertices v_i, v_j (e.g., *external touch to*, *crosses*) with weight w_{ij} (e.g., 0.88) and it is denoted as $e(v_i, v_j, w)$. A finite sequence of undirected edges $e_1, e_2, \ldots, e_m$ which links two vertices v_i, v_j is called *path* and denoted as $\rho(v_i, v_j)$. The complete list of such undirected edges represents the background information on the dissimilarity among atoms and allows to join patterns by introducing disjunctions *(externa_touch_to(A,B)* $\vee$ *crosses(A,B))*.

Discovered patterns are conjunctions of Datalog non-ground atoms and disjunctions of non-ground atoms, which can be expressed by means of a set notation. A Datalog non-ground atom is an n-ary predicate symbol applied to n terms (either constants or variables), at least one of which is a variable. A formal definition of pattern of our interest is reported in the following:

Definition 1. *A* disjunctive pattern P *is a set of atoms and disjunctions of atoms* $p_0(t_0^1)$, $(p_1(t_1^1, t_1^2)|p_2(t_2^1, t_2^2)|\ldots), \ldots, (p_k(t_k^1, t_k^2)|\ldots|p_{k+h}(t_{k+h}^1, t_{k+h}^2))$ *where* p_0 *is the key predicate, while* p_i*,* $i = 1, \ldots, k+h$*, is either a structural predicate or a property predicate or an* is_a *predicate. Symbol " | " indicates disjunctions.*

Terms t_i^j are either constants, which correspond to values of property predicates, or variables, which identify target objects or non-target objects. Each p_i is a predicate occurring in D_E (extensionally defined predicate).

Some examples of disjunctive patterns are the following:

$P_1 \equiv district(A), (comes\_from(A, B)|external\_ends\_at(A, B)), shape(A, rectangle)$

$P_2 \equiv district(A), (external\_ends\_at(A, B)|runs\_along\_boundary\_and\_goes\_in(A, B)),$
$transport\_net(A, roads)$

where the variables A denote target objects, and variables B denote some non-target objects, while the predicates $district(A)$ identify the key predicate in P_1 and P_2, $shape(A, rectangle)$ and $transport\_net(A, roads)$ are property predicates and the others are structural predicates. All variables are implicitly existentially quantified.

We now can give a formal statement of the problem of discovering relational frequent patterns with disjunctions:
1. *Given:* the extensional part D_E of a deductive database D, and two thresholds $minSup \in [0; 1]$, $nSup \in [0; 1]$, the former represents a minimum frequency value while the latter represents maximum frequency value ($nSup < minSup$), *Find:* the collection I_R of the relational infrequent patterns whose support is included in $[nSup; minSup)$.
2. *Given:* the collection I_R, the intensional part D_I of a deductive database D, and two thresholds $minSup$ and $\gamma \in [0; 1]$ (γ defines the maximum dissimilarity value of atoms involved in the disjunctions), *Find:* relational disjunctive patterns whose frequency exceeds $minSup$ and whose dissimilarity of atoms involved in the disjunctions does not exceed γ.

3.1 Mining Infrequent Conjunctive Patterns

The intuition underlying the discovery of pattern with disjunctions is that of extending infrequent conjunctive patterns with disjunctive forms until the threshold $minSup$ is exceeded. Each conjunctive pattern P is associated with a statistical parameter $sup(P, D)$ (support of P on D), which is the percentage of *units of analysis* in D *covered by* P. More precisely, a unit of analysis of a target object $s \in S$ is a subset of ground atoms in D_E defined as follows:

$$D[s] = is\_a(R(s)) \cup D[s|R(s)] \cup \bigcup_{r_i \in R(s)} D[r_i|R(s)], \tag{1}$$

where $R(s)$ is the set of NTO directly or indirectly related to s, $is\_a(R(s))$ is the set of is_a atoms which define the sorts of $r_i \in R(s)$, $D[s|R(s)]$ contains properties of s and relations between s and some $r_i \in R(s)$, $D[r_i|R(s)]$ contains properties of r_i and relations between r_i and some $r_j \in R(s)$. By assigning a pattern P with an existentially quantified conjunctive formula $eqc(P)$ obtained by transforming P into a Datalog query, the units of analysis $D[s]$ are *covered by* a pattern P if $D[s] \models eqc(P)$, namely $D[s]$ logically entails $eqc(P)$).

Conjunctive patterns are mined with SPADA which however enables the discovery of relational patterns whose support exceeds $minSup$ (frequent patterns). SPADA performs a breadth-first search of the space of patterns, from the most general to the more specific ones, and prunes portions of the space which contain only infrequent patterns, which are the conjunctive patterns of our interest. The pruning strategy guarantees that all infrequent patterns are removed and, at this aim, uses a generality ordering based on the notion of θ-subsumption [11]:

Definition 2. *P_1 is more general than P_2 under θ-subsumption ($P_1 \succeq_\theta P_2$) if and only if P_1 θ-subsumes P_2, i.e. a substitution θ exists, such that $P_1\theta \subseteq P_2$.*

For instance, given $P1 \equiv district(A), crosses(A,B)$, $P2 \equiv district(A), crosses(A,B), is\_a(B, transport\_net)$, $P3 \equiv district(A), crosses(A,B), is\_a(B, transport\_net), along(A,C)$ we observe that P_1 θ-subsumes P_2 ($P_1 \succeq_\theta P_2$) and P_2 θ-subsumes P_3 ($P_2 \succeq_\theta P_3$) with substitutions $\theta_1 = \theta_2 = \oslash$. The generality order is monotonic with respect to the pattern support, so whenever $P1$ will be infrequent the patterns more specific of it (e.g., $P2, P3$) will be infrequent too.

The search is based on the level-wise method and implements a two-stepped procedure: i) generation of candidate patterns with k atoms (k-th level) by considering the frequent patterns with $k-1$ atoms (*(k-1)*-th level); ii) evaluation of the frequency with k atoms. So, the patterns whose support does not exceeds $minSup$ will be not considered for the next level: the patterns discarded (infrequent) at each level are rather considered for the generation of disjunctions. The collection I_R is thus composed of a subset of infrequent patterns, more precisely those with support greater than or equal to $nSup$ (and less than $minSup$). A detailed description on SPADA can be found in [8].

3.2 Extending Relational Patterns with Disjunctions

The generation of disjunctive patterns is performed by creating disjunctions among similar atoms in accordance to the background knowledge: two patterns which present similar atoms are joined to form only one. The implemented algorithm (see Algorithm 1) is composed of two sub-procedures: the first one (lines 2-12) creates a graph $\mathcal{G}_\mathcal{D}$ with the patterns of I_R by exploiting the knowledge defined in D_I, while the second one (lines 13-32) joins two patterns (vertices) on the basis of the information (weight) associated to their edge.

In particular, for each pair of patterns which have the same length (namely, at the same level of the level-wise search method) it checks whether they differ in only one atom and share the remaining atoms up to a redenomination of variables (line 3). Let α and β be the two atoms differentiating P from Q (α in P, β in Q), a path ρ which links α to β (or viceversa) is searched among the weighted edges according to D_I: in the case the sum ω of the weights found in the path is lower than the maximum dissimilarity γ the vertices P and Q are inserted into $\mathcal{G}_\mathcal{D}$ and linked through an edge with weight ω (lines 4-9). Note that when there is more than one path between α and β, then the path with lowest weight is considered. Intuitively, at the end of the first sub-procedure, $\mathcal{G}_\mathcal{D}$ will contain, as vertices, the patterns which meet the condition at the line 3, and it will contain, as edges, the weights associated to the path linking the atoms differentiating the patterns.

Once we have $\mathcal{G}_\mathcal{D}$, a list $\mathcal{L}_\mathcal{D}$ is populated with the vertices and edges of $\mathcal{G}_\mathcal{D}$: an element of $\mathcal{L}_\mathcal{D}$ is a triple $\langle P, Q, \omega \rangle$ composed of a pair of vertices-patterns (P,Q) with their relative weight. Elements in $\mathcal{L}_\mathcal{D}$ are ranked in ascending order with respect to the values of ω so that the pairs of patterns with lower dissimilarity will be joined for first. This guarantees that disjunctions with very similar atoms will be preferred to the others (line 13). For each element of $\mathcal{L}_\mathcal{D}$ whose weight ω is lower than γ the two patterns P, Q are joined to generate a pattern J composed by the conjunction of the same atoms in common to the two patterns P, Q and of

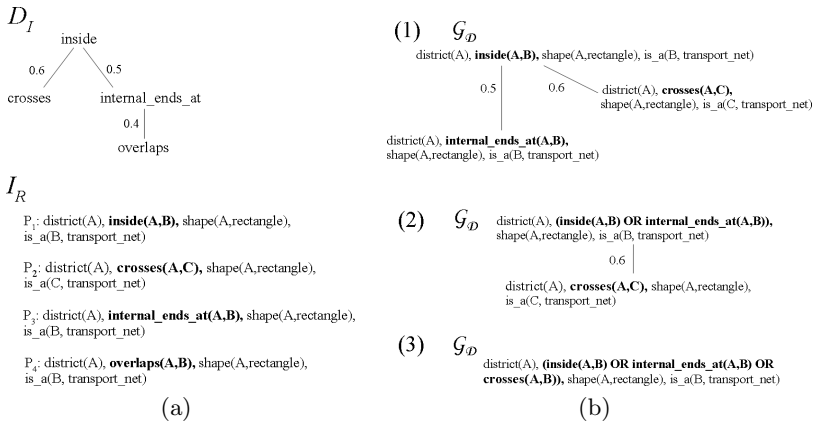

Fig. 1. Extending relational pattern with disjunctions: an example (γ=0.7)

the disjunction formed by the two different (but similar) atoms (lines 14-15). This joining procedure permits to have patterns with the same length of the original ones and which occur when at least one of original patterns occurs. Therefore, if a pattern J is obtained by joining P and Q, it covers a set of units of analysis equal to the union of those of P and Q: the support of J is determined as in line 16 and, generally, it is higher than the support of P and Q. In the case the support of J exceeds *minSup* then it can be considered statistically interesting and no further processing is necessary (lines 16-17). Otherwise, J is again considered and inserted into $\mathcal{G}_\mathcal{D}$ as follows. The edges which linked another pattern R of $\mathcal{G}_\mathcal{D}$ to P and Q are modified in order to keep the links from R to J: the weight of the edges between one pattern R and J will be set to the average value of the weights of all the edges which linked R to P and Q (lines 19-27). The modified graph $\mathcal{G}_\mathcal{D}$ contains conjunctive patterns (those of I_R) and pattern with disjunctions (those produced by joining). Thus, $\mathcal{G}_\mathcal{D}$ is re-evaluated for further joins and the algorithm proceeds iteratively (line 29-30) until no additional disjunctions can be done (namely, when $\mathcal{L}_\mathcal{D}$ is empty or the weights ω are higher than γ). At each iteration, the patterns P and Q are removed from $\mathcal{G}_\mathcal{D}$ (line 32).

An explanatory example is illustrated in Figure 1. Consider the background knowledge D_I on the dissimilarity among four atoms and the set I_R containing four infrequent conjunctive patterns as illustrated in Figure 1a and γ equal to 0.7. The first sub-procedure of the algorithm 1 analyzes P_1, P_2, P_3, P_4 and discovers that they differ in only one atom, while the other atoms are in common. Then, it creates the graph $\mathcal{G}_\mathcal{D}$ by collocating P_1, P_2, P_3 in three different vertices and linking them through edges whose weights are taken from the paths ρ in D_I. P_4 is not considered because the vertex *overlaps* has dissimilarity with *internal_ends_at* higher than γ (row (1) in Figure 1b). The second sub-procedure starts by ordering the weights of the edges: the first disjunction is created by joining P_1 and P_3 given that the dissimilarity value is lower than γ and the lowest (row (2) in Figure 1b). Next, the pattern so created and P_2 are checked

Algorithm 1. Extending Relational Pattern with Disjunctions

input: $I_R, D_I, \gamma, minSup$ **output:** $\mathcal{J}$ // $\mathcal{J}$ set of disjunctive patterns
for all $(P,Q) \in I_R \times I_R$, $Q \neq P$ **do**
  if $P.length = Q.length$ and $check\_atoms(P,Q)$ **then**
    $(\alpha, \beta) := atoms\_diff(P,Q)$ //α, β atoms differentiating P,Q
    if $\rho(\alpha,\beta) \neq \oslash$ **then**
      $\omega := \sum_{e(v_i,v_j,w_{ij}) \ in \ \rho(\alpha,\beta)} w_{ij}$
      if $\omega \leq \gamma$ **then**
        $addNode(P, \mathcal{G}_\mathcal{D})$; $addNode(Q, \mathcal{G}_\mathcal{D})$; $addEdge(P, Q, \omega, \mathcal{G}_\mathcal{D})$
      end if
    end if
  end if
end for
$\mathcal{L}_\mathcal{D} \leftarrow$ edges of $\mathcal{G}_\mathcal{D}$ // list of edges of $\mathcal{G}_\mathcal{D}$ ordered in ascending mode w.r.t. ω
while $\mathcal{L}_\mathcal{D} \neq \oslash$ and $\forall e(P,Q,\omega) \in \mathcal{G}_\mathcal{D}$ $\omega \leq \gamma$ **do**
  $J \leftarrow join(P,Q)$; $J.support := P.support + Q.support - (P \cap Q).support$;
  if $J.support \geq minSup$ **then**
    $\mathcal{J} := \mathcal{J} \cup \{J\}$
  else
    for all R *such that* $\exists\ e(P,R,\omega_1) \in \mathcal{G}_\mathcal{D}$ and $\exists\ e(Q,R,\omega_2) \in \mathcal{G}_\mathcal{D}$ **do**
      $addEdge(R, J, (\omega_1 + \omega_2)/2, \mathcal{G}_\mathcal{D})$
    end for
    for all R *such that* $\exists\ e(P,R,\omega_1) \in \mathcal{G}_\mathcal{D}$ and $\nexists\ e(Q,R,\omega_2) \in \mathcal{G}_\mathcal{D}$ **do**
      $addEdge(R, J, \omega_1, \mathcal{G}_\mathcal{D})$
    end for
    for all R *such that* $\exists\ e(Q,R,\omega_2) \in \mathcal{G}_\mathcal{D}$ and $\nexists\ e(P,R,\omega_1) \in \mathcal{G}_\mathcal{D}$ **do**
      $addEdge(R, J, \omega_2, \mathcal{G}_\mathcal{D})$
    end for
    $\mathcal{L}_\mathcal{D} \leftarrow$ edges of $\mathcal{G}_\mathcal{D}$
    update $\mathcal{L}_\mathcal{D}$
  end if
  $removeNode(P, \mathcal{G}_\mathcal{D})$; $removeNode(Q, \mathcal{G}_\mathcal{D})$
end while

for joining. Both have the same length and differ in only one atom. Although the first presents a disjunction and the second presents a "simple" atom, dissimilarity is lower than γ and a new disjunctive pattern is created (row (3) in Figure 1b).

4 Experiments

The described approach has been implemented as the upgrading of the system SPADA to discover relational patterns with disjunctions: the system (afterwards $jSPADA$) is now able to mine relational conjunctive patterns and disjunctive patterns as well. The experiments were performed in order to evaluate the viability of jSPADA and to compare it with SPADA from a quantitative and qualitative

standpoint[1]. In this section we present the application of both systems in spatial data mining [2] in order to discover statistical regularities in the spatial objects which can be exploited in decision making for transportation planning.

More precisely, frequent relational patterns are mined from a dataset concerning census and digital maps of Stockport, one of the ten districts in Greater Manchester, to investigate the accessibility *to* the Stepping Hill Hospital *from* the actual residence of people living within in the area served by the hospital. To define the accessibility we used the Ordnance Survey data on transport network, namely the layers of roads, railways and bus priority lines. Frequent patterns can relate five areal spatial objects or *districts* (non-target objects) which are close to the Stepping Hill Hospital with one-hundred and fifty-two districts distant from the hospital (target objects) through the transport network lines (non-target objects). D_E contains 1147 ground atoms for 152 target objects.

Property predicates represent discretized numerical census data in TO and describe the households (people) with car, more precisely these are: *no_car()*, *one_car()*, *two_cars()*, *three_more_cars()*. Structural predicates represent binary topological relations between districts and roads, railways or bus lines, and correspond to the twelve feasible relations between a region and a line according to the 9-intersection model [7]. Here, background knowledge D_I has been defined on the structural predicates and the dissimilarity values have been manually determined by applying the Sokal-Michener dissimilarity measure on the matrix representation of the twelve relations[5]. In this sense, the goal of jSPADA is of discovering disjunctive patterns defined among the twelve relations which can express information otherwise discarded by SPADA.

Experiments were performed by tuning the thresholds $minSup$, $nSup$, γ and the results are reported in Figure 2. A comparison between SPADA and jSPADA has been conducted by varying $minSup$, while, for jSPADA, the values of $nSup$ and γ are set to 0.005 and 0.6 respectively. As we see the histogram values in Figure 2a, jSPADA discovers an higher number of patterns than that of SPADA. Indeed, jSPADA returns a set which includes those frequent conjunctive (generated by SPADA) and those disjunctive generated by re-evaluating the infrequent conjunctive ones. Thus, as $minSup$ increases, the range $[nSup; minSup)$ becomes wider and, generally, more disjunctive patterns are extracted while the number of conjunctive frequent patterns decreases.

As expected, also the threshold $nSup$ has influence on the patterns discovered by jSPADA. Indeed, from the figures 2c, 2d ($minSup = 0.025$ and $\gamma = 0.6$) we note that jSPADA is highly sensitive to $nSup$ since the number of disjunctive patterns is reduced of one order of magnitude (from 20 to 0) as $nSup$ is increased by factor of two (from 0.01 to 0.02). By comparing the plots a), c) and d) we note that, by varying $minSup$, have a limited capacity in unearthing infrequent patterns (but potentially interesting) than when varying $nSup$. This confirms the viability of the approach to discover new forms of interesting patterns. Another quantitative analysis can be done with respect to the dissimilarity of the disjunctions (Figure 2b). At high values of γ disjunctions can be created also between atoms whose

[1] Data and results are accessible at http://www.di.uniba.it/∼loglisci/jSPADA/

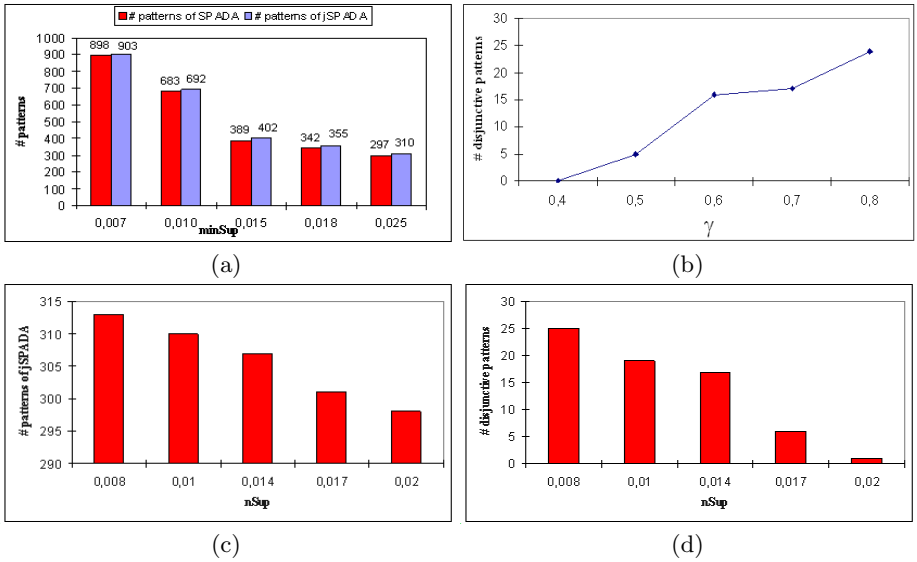

Fig. 2. Number of patterns discovered by tuning *minSup*, *nSup*, γ

similarity is small, so the patterns present disjunctions with several atoms and the final set is larger. On the contrary, lower values of γ permit of identifying disjunctions only between very similar atoms, so the disjunctions present less atoms and the final set is smaller: when γ is set to 0.4 no disjunction is created since the minimum value of similarity between atoms amounted to 0.44.

A comparison between jSPADA and SPADA can also be done from a qualitative viewpoint. jSPADA enables the discovery of patterns which enrich the information conveyed by the patterns of SPADA. For instance, the pattern discovered by SPADA

$P_1 : district(A), comes\_from(A, B), is\_a(B, road), comes\_from(A, C), is\_a(C, road)$ $[support : 12\%]$ is enriched by

P_2 discovered by jSPADA:

$P_2 : district(A), [comes\_from(A, C)|external\_ends\_at(A, C)], is\_a(C, road),$
$comes\_from(A, B), is\_a(B, rail)$ $[support : 16\%]$

which introduces the disjunctions $comes\_from(A, C)|external\_ends\_at(A, C)$ between two structural predicates. P_2 expresses the information that the road named as C can be connected to the district named as A through two possible simultaneous or alternative ways, $comes\_from(A, C)$ (C starts in A and terminates outside A) and $external\_ends\_at(A, C)$ (C starts outside A and terminates inside A). Remarkably the support of P_2 is higher than that of P_1. jSPADA permits also the discovery of completely novel patterns that SPADA neglects. One of these is the following:

$P_3 : district(A), [external\_ends\_at(A, B)|along(A, B)|comes\_from(A, B)],$
$three\_more\_cars(A, [0.033; 0.114])$ $[support : 11.1\%]$

which introduces a property predicate (i.e., the percentage of households owing more three cars included in [0.033;0.114]) and expresses in the disjunction three possible forms of accessibility to the district A by the transport line B.

5 Conclusion

In this paper we present a relational data mining approach that discovers frequent patterns that consider disjunctive forms. We advocate to the relational approach to properly deal with the complexity of real-world data. The approach enables the discovery of disjunctive patterns by re-evaluating the infrequent conjunctive patterns and extending them with disjunctions created through the exploitation of a background knowledge. We applied the algorithm to the domain of the spatial analysis and the experimental results prove the advantages of the proposed algorithm with respect to traditional algorithms of frequent pattern mining. As future work, we intend to apply jSPADA to other domains.

Acknowledgement. This work is partial fulfillment of the research objectives of the projects "DM19410 - The Molecular Biodiversity LABoratory Initiative" and "ATENEO 2008 - Scoperta di conoscenza in domini relazionali".

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI/MIT Press (1996)
2. Appice, A., Ceci, M., Lanza, A., Lisi, F.A., Malerba, D.: Discovery of spatial association rules in geo-referenced census data: A relational mining approach. Intell. Data Anal. 7(6), 541–566 (2003)
3. Ceri, S., Gottlob, G., Tanca, L.: Logic Programming and Databases. Springer, Heidelberg (1990)
4. Dehaspe, L., Toivonen, H.: Discovery of frequent datalog patterns. Data Min. Knowl. Discov. 3(1), 7–36 (1999)
5. Diday, E., Esposito, F.: An introduction to symbolic data analysis and the sodas software. Intell. Data Anal. 7(6), 583–601 (2003)
6. Dzeroski, S., Lavrac, N.: Relational Data Mining. Springer, Heidelberg (2001)
7. Egenhofer, M.J., Franzosa, R.D.: Point set topological relations. International Journal of Geographical Information Systems 5, 161–174 (1991)
8. Lisi, F.A., Malerba, D.: Inducing multi-level association rules from multiple relations. Machine Learning 55(2), 175–210 (2004)
9. Lloyd, J.W.: Foundations of Logic Programming, 2nd edn. Springer, Heidelberg (1987)
10. Nanavati, A.A., Chitrapura, K.P., Joshi, S., Krishnapuram, R.: Mining generalised disjunctive association rules. In: CIKM, pp. 482–489. ACM Press, New York (2001)
11. Plotkin, G.D.: A note on inductive generalization. Machine Intelligence 5, 153–163 (1970)
12. Roddick, J.F., Fule, P.: Semgram - integrating semantic graphs into association rule mining. In: Proc. of AusDM, vol. 70, pp. 129–137 (2007)
13. Srikant, R., Agrawal, R.: Mining generalized association rules. In: VLDB, pp. 407–419. Morgan Kaufmann, San Francisco (1995)