Object Identity as Search Bias for Pattern Spaces

Francesca A. Lisi and **Stefano Ferilli** and **Nicola Fanizzi** [1]

Abstract. In the context of frequent pattern discovery, we present a generality relation, called θ_{OI}-*subsumption*, which is based on the assumption of Object Identity in spaces of patterns to be intended as existentially quantified conjunctive formulæ. The resulting generality order $\preceq_{\text{OI}}$ seems appropriate for organizing efficiently the space of DATALOG patterns over structured domains. Indeed we prove the existence of ideal refinement operators for $\preceq_{\text{OI}}$-ordered spaces and the monotonicity of $\preceq_{\text{OI}}$ with respect to pattern support. Features of such spaces are illustrated by means of an example of frequent pattern discovery in spatial data.

1 INTRODUCTION

The design of algorithms for frequent pattern discovery has turned out to be a popular topic in data mining [8]. The blueprint for most algorithms proposed in the literature is the levelwise method that is based on a breadth-first search in the lattice spanned by a generality order between patterns [16]. The space of patterns is searched one level at a time, starting from the most general patterns and iterating between candidate generation and candidate evaluation phases. Recently, an extension of the levelwise method to the discovery of frequent DATALOG patterns has been presented [4]. It relies on techniques of Inductive Logic Programming (ILP) in order to exploit the common background with logical languages for databases such as DATALOG [3].

The choice of the generalization model for a space of DATALOG patterns affects both its algebraic structure and, as a consequence, the definition of refinement operators to work on it. Two main approaches to the problem of generalization have been followed in ILP [13]: the logical approach and the graph matching approach. As to the former, logical implication has proven particularly hard to handle due to many negative results descending from its intrinsic complexity and non-decidability [10]. Thus, θ-*subsumption* is usually employed in ILP since it turns out to be more tractable [17]. Yet, this relationship is not fully satisfactory because of the complexity issues that anyhow it yields [9]. Then, proposed solutions regard other forms of subsumption which follow the latter approach [11].

In this paper, we propose to weaken further both θ-subsumption and implication by assuming the *object identity* bias. The resulting generality relations are called θ_{OI}-*subsumption* and *OI-implication*, respectively. The paper extends results reported in [5, 6] to the case of existentially quantified conjunctive formulæ. In particular, the generality order based on θ_{OI}-subsumption, denoted $\preceq_{\text{OI}}$, seems appropriate for organizing efficiently the space of DATALOG patterns over structured domains. Indeed, we prove the existence of ideal refinement operators for $\preceq_{\text{OI}}$-ordered spaces and the monotonicity of $\preceq_{\text{OI}}$with respect to pattern support. These properties may be of great interest to the data mining community since most research nowadays focuses on methods for discovering frequent patterns in data that are characterized by the presence of objects, properties of objects, and relations among objects. An example of such structured domains is spatial data. See [15] for details about an ILP method for mining spatial association rules.

The paper is organized as follows. Section 2 introduces the task of discovering frequent DATALOG patterns. In Section 3, the object identity bias is illustrated and applied to both θ-subsumption and implication. Section 4 is devoted to the presentation of the generality order based on θ_{OI}-subsumption. An illustrative example is commented in Section 5. Concluding remarks are given in Section 6.

[1] Dipartimento di Informatica, Università degli Studi di Bari, Via Orabona 4, I-70125 Bari, Italy, email: {lisi,ferilli,fanizzi}@di.uniba.it

2 DISCOVERING FREQUENT DATALOG PATTERNS

Let $\mathcal{A}$ be a set of DATALOG atoms. Conjunctions of atoms in $\mathcal{A}$ are called atomsets. In our framework, the language of patterns $\mathcal{L}$ is the set of well-formed atomsets generated on $\mathcal{A}$. Well-formedness encompasses properties like linkedness [12] and safety [3] which guarantees the correct evaluation of patterns.

The core mechanism in *candidate evaluation* is pattern matching. To this aim, patterns are to be intended as existentially quantified conjunctive formulæ (or *queries*) [11, 17] and data can be considered as either (intensionally) a logic theory or (extensionally) an interpretation.

Definition 2.1 *Let* R *be a definite database. A query* Q matches R *iff* Q *is true in the least Herbrand model of* R. *The set* $answerset(Q, \text{R})$ *contains all the ground substitutions* θ *such that* $Q\theta$ *matches* R.

This matching could be checked in polynomial time [17].

In the case of association patterns, the evaluation function determines the support of a candidate pattern given a certain K in $\mathcal{A}$ that specifies what is being counted (called *key atom*). Thus we are interested in the number of different bindings for the variables occurring in K. The set $answerset(Q, \text{R})_{|K}$ contains those substitutions σ that are obtained by restriction of substitutions in $answerset(Q, \text{R})$ to the atom K.

Definition 2.2 *Let* Q *be a query with key atom* K. *The* support of the query Q w.r.t. R given K *is defined:*

$$supp(Q, \text{R}, K) = \frac{\mid answerset(Q, \text{R})_{|K} \mid}{\mid answerset(K, \text{R}) \mid}$$

Candidate patterns with support greater than a user-defined minimum threshold (*frequent patterns*) are retained. Frequent patterns are commonly not considered useful for presentation to the user as such. They can be efficiently post-processed into rules that exceed given

threshold values. In the case of association rules the measure of confidence offer a natural way of pruning weak and rare rules [1].

Evaluation functions play a relevant role also in *candidate generation*. Indeed, this phase takes advantage of generality orders that are monotonic with respect to the chosen evaluation function. The monotonicity property allows us to specify pruning criteria. This is the case of θ-subsumption. In [4] it has been proven that:

Proposition 2.1 *Given a definite database* R *and two queries* Q_1 *and* Q_2 *that contain the key atom* K*, it holds: if* $Q_1 \preceq_\theta Q_2$ *then* $supp(Q_1, \mathrm{R}, K) \leq supp(Q_2, \mathrm{R}, K)$.

Candidate generation consists of refinement steps followed by pruning. The former applies a refinement operator under θ-subsumption to patterns previously found frequent by preserving the properties of linkedness and safety. Applying such operator usually consists of adding to the pattern to be refined one or more DATALOG atoms in $\mathcal{A}$. The pruning step usually involves verifying that candidate patterns do not θ-subsumes infrequent patterns. This allows some infrequent patterns to be detected and discarded prior to evaluation.

3 THE OBJECT IDENTITY BIAS

Our proposal relies essentially on the following bias that can be considered as an extension of Reiter's *unique names* assumption [19]:
[Object Identity] *In a formula, terms denoted with different symbols must be distinct, i.e. they represent different entities of the domain.*
This assumption can be the starting point for the definition of both an equational theory for DATALOG formulas and a quasi-ordering for DATALOG spaces. As to the former option, it consists of the axioms of Clark's Equality Theory augmented with one rewriting rule that adds atoms $s \neq t$ to any $P \in \mathcal{L}$ for each pair (s, t) of distinct terms occurring in P. The resulting language is a subset of DATALOG$^{\neq}$, called DATALOG$^{\mathrm{OI}}$ [20]. For instance, the DATALOG pattern

$$P = c(X, a), r(X, Y), c(Y, b), r(X, Z), c(Y, b)$$

stands for the DATALOG$^{\mathrm{OI}}$ pattern

$$P^{\mathrm{OI}} = c(X, a), r(X, Y), c(Y, b), r(X, Z), c(Y, b), X \neq Y, \\ Y \neq Z, X \neq a, X \neq b, X \neq c, Y \neq a, Y \neq b, Y \neq c.$$

One major drawback of Object Identity as *language bias* is the cost of candidate evaluation. Indeed, it has been proved that conjunctive queries with inequalities are in general intractable [14] (except some cases mentioned later). Thus, we are more interested in Object Identity as *search bias*.

The intuition underlying OI-compliant quasi-orderings for DATALOG spaces can be illustrated by means of the patterns $P = p(X, X)$ and $Q = p(X, X), p(X, Y), p(Y, Z), p(Z, X)$. Adopting θ-subsumption as generalization model for the space $\mathcal{L}$, P and Q are equivalent. This is not so natural as it might appear, since more elements of the domain may be involved in Q than in P (indeed in our framework P is more general than Q). The expressive power is not diminished by this bias, since it is always possible to convey the same meaning of a pattern, yet it might be necessary to employ more patterns in $\mathcal{L}$, e.g. $P = p(X, Y)$ is equivalent to the set of patterns $\{P_1 = p(X, X), P_2 = p(X, Y)\}$.

From a syntactic viewpoint, since a substitution can be regarded as a mapping from the variables to terms of a language, we require these mappings to avoid the identification of terms:

Definition 3.1 *A substitution* σ *is an* OI-substitution *w.r.t. a set of terms* T *iff* $\forall t_1, t_2 \in T\colon\ t_1 \neq t_2$ *yields that* $t_1\sigma \neq t_2\sigma$.

In order to cope with the object identity assumption, a relationship has been derived [5] from the classic θ-subsumption. It can be regarded as a structural relation of subgraph isomorphism [11].

Definition 3.2 *Let* $P, Q \in \mathcal{L}$. P θ_{OI}-subsumes Q *iff* $P \supseteq Q\sigma$ *for some OI-substitution* σ *w.r.t.* $terms(Q)$.

Adopting the object identity bias over the interpretations, the semantics of the language of patterns $\mathcal{L}$ can be defined as follows:

Definition 3.3 *A* pre-interpretation J *of the language* $\mathcal{L}$ *on a* domain $\mathcal{D}$ *assigns each* n*-ary (*$n \geq 0$*) function symbol* f *to a mapping from* $\mathcal{D}^n$ *to* $\mathcal{D}$. *An* OI-interpretation I *based on* J *is a set of ground instances of atoms with arguments in* $\mathcal{D}$ *through* J. *Given a ground OI-substitution* γ *mapping* $vars(\mathcal{L})$ *to* $\mathcal{D}$*, an instance* $A\gamma$ *of an atom* A *is* true *in* I *iff* $A\gamma \in I$. *A negative literal* $\neg A\gamma$ *is* true *in* I *iff* $A\gamma \notin I$. *An OI-interpretation* I *is an* OI-model *for a pattern* P *iff there exists a ground OI-substitution* γ *such that all literals in* $P\gamma$ *are true in* I.

All the notions defined for standard semantics can be straightforwardly transposed to this semantics. In particular, the form of implication that is compliant with object identity [6]:

Definition 3.4 *Let* $P, Q \in \mathcal{L}$. P implies Q under object identity *(denoted* $P \models_{\mathrm{OI}} Q$*) iff all OI-models for* P *are OI-models for* Q.

This relationship is decidable for the case of clauses [7]. Observed that clauses can be obtained by negating patterns, the decidability of OI-implication between patterns descends from the case mentioned above by contradiction. OI-implication is a stronger relationship than θ_{OI}-subsumption:

Proposition 3.1 *Let* $P, Q \in \mathcal{L}$*, if* P θ_{OI}*-subsumes* Q *then* $P \models_{\mathrm{OI}} Q$.
Proof: *By definition,* P θ_{OI}*-subsumes* Q *yields* $\exists\sigma\ Q\sigma \subseteq P$. *Now let* I *be an OI-model for* P. *Then* $\forall L\gamma \in P\gamma\colon\ L\gamma \in I$ *for some ground substitution* γ. *Consider the ground substitution* $\gamma' = \sigma\gamma$. *By construction* $Q\gamma' \subseteq P\gamma$. *Then* $\forall L' \in Q\ \exists L \in P\ L\gamma = L'\gamma'$ *such that* $L'\gamma' \in I$*, i.e.* I *is an OI-model for* Q.

In clausal spaces, the proof-theory based on the notion of OI-unifier (a unifying OI-substitution) can be given. Resolution and derivation under object identity have been proven sound and a subsumption theorem also holds, thus bridging the gap between model-theory and proof-theory [6].

4 THE GENERALITY ORDER AND ITS REFINEMENT OPERATORS

The goal is to define a generality order on patterns. From an extensional point of view a pattern Q is "more general" than another pattern P when $P \models_{\mathrm{OI}} Q$. Also θ_{OI}-subsumption can be adopted to induce a generality order over pattern spaces. In various cases the two orders coincide [6].

Definition 4.1 *Let* $P, Q \in \mathcal{L}$. $P \preceq_{\mathrm{OI}} Q$ *iff* P θ_{OI}-subsumes Q, $P \prec_{\mathrm{OI}} Q$ *iff* $P \preceq_{\mathrm{OI}} Q$ *and* $Q \not\preceq_{\mathrm{OI}} P$. *Finally,* $P \sim_{\mathrm{OI}} Q$ *iff* $P \preceq_{\mathrm{OI}} Q$ *and* $Q \preceq_{\mathrm{OI}} P$.

It can be easily proven that $\preceq_{\mathrm{OI}}$ is a quasi-order:

Proposition 4.1 *Let* $P, Q \in \mathcal{L}$. *If* $P \sim_{\mathrm{OI}} Q$ *then* P *and* Q *are alphabetic variants.*
Proof: *In the hypotheses above* $\exists\theta_1, \theta_2\colon\ Q\theta_1 \subseteq P$ *and* $P\theta_2 \subseteq Q$. *Observed that OI-substitutions do not identify literals, we can write:* $|Q| \leq |P|$ *and* $|P| \leq |Q|$. *Then* $|P| = |Q|$ *and the patterns must be alphabetic variants.*

Proposition 4.2 *$(\mathcal{L}, \preceq_{\text{OI}})$ is a quasi-ordered set.*
Proof:
[symmetry] *Trivial, since $\forall P \in \mathcal{L}$: P θ_{OI}-subsumes P.*
[transitivity] *Let $P_1, P_2, P_3 \in \mathcal{L}$ and $P_1 \preceq_{\text{OI}} P_2$ and $P_2 \preceq_{\text{OI}} P_3$. Then $\exists \sigma_2, \sigma_3 : P_2\sigma_2 \subseteq P_1$ and $P_3\sigma_3 \subseteq P_2$. For $\sigma = \sigma_3\sigma_2$ it holds $P_3\sigma \subseteq P_1$ and hence P_1 θ_{OI}-subsumes P_3, that is $P_1 \preceq_{\text{OI}} P_3$.*

The monotonicity of $\preceq_{\text{OI}}$ is now investigated with respect to the evaluation function. To this aim, Definition 2.1 and Proposition 2.1 are extended to our framework.

Definition 4.2 *Let* R *be a definite database. A query Q* matches R *iff Q is true in the least Herbrand OI-model of* R*. The set $answerset(Q, \text{R})$ contains all the ground OI-substitutions θ such that $Q\theta$ matches* R*.*

Proposition 4.3 *Given a definite database* R *and two queries Q_1 and Q_2 that contain the key atom K, it holds: if $Q_1 \preceq_{\text{OI}} Q_2$ then $supp(Q_1, \text{R}, K) \leq supp(Q_2, \text{R}, K)$.*
Proof: *Observe that $Q_1 \preceq_{\text{OI}} Q_2$ iff Q_1 θ_{OI}-subsumes Q_2. Then, by Proposition 3.1, $Q_1 \models_{\text{OI}} Q_2$. By Definition 4.2, if Q_1 matches* R *then Q_2 matches* R*. Let θ be a ground OI-substitution in $answerset(Q_1, \text{R})$. It holds that $\theta \in answerset(Q_2, \text{R})$. By restricting θ to K, we obtain a OI-substitution σ in $answerset(Q_1, \text{R})_{|K}$ that belongs also to $answerset(Q_2, \text{R})_{|K}$. From Definition 2.2, $supp(Q_1, \text{R}, K) \leq supp(Q_2, \text{R}, K)$ follows.*

The space resulting from the adoption of this generality order is not a lattice like for the case of θ-subsumption. Indeed, minimal generalizations and maximal specializations of patterns are not guaranteed to be unique. We need to investigate on refinement operators and their properties in this pattern space [17, 5].

Definition 4.3 *In a quasi-ordered set $(\mathcal{L}, \preceq)$, a* downward *(resp.* upward*)* refinement operator *is a mapping from $\mathcal{L}$ to $2^{\mathcal{L}}$ such that $\rho(P) \subseteq \{Q \in \mathcal{L} \mid Q \preceq P\}$ (resp. $\delta(P) \subseteq \{Q \in \mathcal{L} \mid P \preceq Q\}$) $\forall P \in \mathcal{L}$. Denoted with τ^* the* transitive closure *of operator τ:*

- *τ is* optimal *iff $\forall P, Q_1, Q_2 \in \mathcal{L}$ it holds $P \in \tau^*(Q_1) \cap \tau^*(Q_2)$ implies $Q_1 \in \tau^*(Q_2)$ or $Q_2 \in \tau^*(Q_1)$;*
- *τ is* locally finite *iff $\forall P \in \mathcal{L} : \tau(P)$ is finite and computable;*
- *τ is* proper *iff $\forall P \in \mathcal{L} \forall Q \in \tau(P)$: $Q \not\sim P$;*
- *ρ (resp. δ) is* complete *iff $\forall P, Q \in \mathcal{L}$ if $Q \prec P$ then $\exists Q' \in \rho^*(P)$: $Q' \sim Q$ (resp. if $P \prec Q$ then $\exists Q' \in \delta^*(P)$: $Q' \sim Q$).*

A locally finite, proper and complete operator is defined *ideal*. The ideality of refinement operators has been recognized as particularly important for the efficiency of search algorithms in spaces with *dense solutions*. Conversely, for spaces with *rare* solutions, it is possible to derive optimal operators from ideal ones, since the former are recognized to be more suitable in this case [2]. Ideal operators have been proven not to exist for clausal spaces ordered by θ-subsumption (or logical implication) [17]. The proof is based on the non-existence of complete covers for some clauses.

Definition 4.4 *In a quasi-ordered set $(\mathcal{L}, \preceq)$, Q is a* downward *(resp.* upward*)* cover *of P iff $Q \prec P$ and $\nexists Q' : Q \prec Q' \prec P$ (resp. $P \prec Q$ and $\nexists Q' : P \prec Q' \prec Q$). A* downward *(resp.* upward*)* cover set *of P, $dc(P)$ (resp. $uc(P)$), is a maximal set of non-equivalent downward (resp. upward) covers of P. $dc(P)$ (resp. $uc(P)$) is* complete *iff $\forall Q \in \mathcal{L}$, $Q \prec P$ $\exists P' \in dc(P)$: $Q \preceq P' \prec P$ (resp. $\forall Q \in \mathcal{L}$, $P \prec Q$ $\exists P' \in uc(P)$: $P \prec P' \preceq Q$).*

An optimal way to define efficient refinement operators would be mapping each pattern to its cover set. This may be infeasible because of the relationship $\preceq$ to be tested or because cover sets can be empty even when a clause has infinitely many non-equivalent proper generalizations or specializations. Indeed, it holds that:

Proposition 4.4 *[17] Given an ideal downward (resp. upward) refinement operator ρ (resp. δ) for $(\mathcal{L}, \preceq)$. $\forall P \in \mathcal{L}$: $dc(P)$ (resp. $uc(P)$) is finite and complete, and $dc(P) \subseteq \rho(P)$ (resp. $uc(P) \subseteq \delta(P)$).*

Thus, if a pattern has an incomplete or infinite cover set in a quasi-ordered space, then an ideal refinement operator cannot be defined. In the space ordered by θ-subsumption $(\mathcal{L}, \preceq_\theta)$, consider the patterns $P = p(X, X)$ and $P_n = \{p(X_i, X_j) \mid i, j \in [1, n], i \neq j\}$, for $n \geq 2$. Note that $P \prec_\theta P_{n+1} \prec_\theta P_n$ and there is no downward cover P' of P such that $P \prec_\theta P' \prec_\theta P_n$ for $n \geq 2$, i.e. $\{P_n\}_{n \geq 2}$ is an uncovered infinite descending chain. Analogously, the pattern $Q = p(X, Y), p(Y, X)$ has no finite and complete downward cover set. This holds also for function-free spaces such as DATALOG. Conversely, the existence of ideal refinement operators for clausal spaces ordered by θ_{OI}-subsumption has been proven [5]. We transpose this result to the case of patterns.

Definition 4.5 *Let P be a pattern in $(\mathcal{L}, \preceq_{\text{OI}})$.*
Then, a pattern $Q \in \rho_{\text{OI}}(P)$ when one of these conditions hold:
$[d.1]$ $Q = P\theta$, where $\theta = \{X/t\}$, $X \in vars(P)$, $t \notin terms(P)$;
$[d.2]$ $Q = P \cup \{L\}$, where L is a literal, such that: $L \notin P$.
Besides, $Q \in \delta_{\text{OI}}(P)$ when one of these conditions hold:
$[u.1]$ $Q = P\sigma$, where $\sigma = \{t/X\}$, $t \in terms(P), X \notin vars(P)$;
$[u.2]$ $Q = P \setminus \{L\}$, where L is a literal, such that: $L \in P$.

Now, in order to prove the completeness of these refinement operators, some lemmas are needed:

Lemma 4.1 *Let $P, Q \in \mathcal{L}$. If there exists an OI-substitution θ such that $P\theta = Q$ then $Q \in \rho^*_{\text{OI}}(P)$ and $P \in \delta^*_{\text{OI}}(Q)$.*
Proof: *$\exists$ $P\theta = Q$. Let $n = |\theta|$, i.e. θ contains n bindings, hence we can write $\theta = \cup_{i=1}^{n} \theta_i$. Each θ_i represent a step $[d.1]$ of the definition of ρ_{OI} (renaming OI-substitutions are left out for brevity). Let $P = P_0, \ldots, P_n = Q$ be a chain such that $P_i = P_{i-1}\theta_i$, $\forall i \in [1, n]$, Thus, $P_i \in \rho_{\text{OI}}(P_{i-1})$, $\forall i \in [1, n]$, then $Q \in \rho^*_{\text{OI}}(P)$.*
$P \in \delta^*_{\text{OI}}(Q)$ using $\sigma = \cup_{i=1}^{n} \sigma_i = \cup_{i=1}^{n} \theta_i^{-1}$.

Lemma 4.2 *Let $P, Q \in \mathcal{L}$. If $P \subseteq Q$ then $Q \in \rho^*_{\text{OI}}(P)$ and $P \in \delta^*_{\text{OI}}(Q)$.*
Proof: *By induction on $n = |Q \setminus P|$.*
If $n = 0$ then $P = Q$, thus $Q \in \rho^0_{\text{OI}}(P) \subseteq \rho^*_{\text{OI}}(P)$.
For $n > 0$, let $Q \setminus P = \{L_1, \ldots, L_n\}$ and for $k \leq n$, $P_k = Q \cup \{L_1, \ldots, L_k\}$. Now let $L_{k+1} \in Q \setminus P_k$ and $P_{k+1} = P_k \cup \{L_{k+1}\}$; we prove the lemma for P_{k+1}. Since $L_{k+1} \in Q \setminus P_k$ then $L_{k+1} \notin P_k$, so it can be used to refine P_k with ρ_{OI} (case $[d.2]$), hence we obtain: $P_{k+1} \in \rho_{\text{OI}}(P_k)$. By inductive hypothesis, $P_k \in \rho^*_{\text{OI}}(P)$. Thus $P_{k+1} \in \delta^*_{\text{OI}}(P)$. Similarly, $Q = P_n \in \delta^*_{\text{OI}}(P)$.

Theorem 4.1 *In the pattern space $(\mathcal{L}, \preceq_{\text{OI}})$, the refinement operators ρ_{OI} and δ_{OI} are ideal.*
Proof: [local finiteness] *Obvious, from Definition 4.5.*
[properness] *Suppose $P \in \rho_{\text{OI}}(Q)$. Then $P \preceq_{\text{OI}} Q$. If also $Q \preceq_{\text{OI}} P$ then $P \sim_{\text{OI}} Q$. Hence, the patterns would be alphabetic variants (see Proposition 4.1), which is not possible (Definition 4.5 yields that P has a new term w.r.t. Q or it is longer than Q). Analogously for ρ_{OI}.*

[completeness] *Let $P, Q \in \mathcal{L}$ such that $P \preceq_{\text{OI}} Q$. Then $\exists \theta Q\theta \subseteq P$. Let $Q' = Q\theta$. For Lemma 4.1, $Q' \in \rho^*_{\text{OI}}(Q)$. Since $Q' \subseteq P$, by applying Lemma 4.2, it holds that $P \in \rho^*_{\text{OI}}(Q')$. Thus $P \in \rho^*_{\text{OI}}(Q)$, then ρ_{OI} is complete (equivalent patterns are not considered since they are alphabetic variants, by Proposition 4.1). A similar proof demonstrates the completeness of δ_{OI}.*

Though both refinement operators are ideal, the downward operator ρ_{OI} is of greater help in the context of frequent pattern discovery. Indeed, by Proposition 4.3, it drives the search towards patterns with decreasing support and enables the early detection of infrequent patterns.

5 AN ILLUSTRATIVE EXAMPLE

As aforementioned, $\preceq_{\text{OI}}$-ordered pattern spaces are suitable for dealing with structured domains. In the following, features of such spaces are illustrated by means of an example of frequent pattern discovery in spatial data. From now on, patterns stand for association patterns. Thus the chosen evaluation function is $supp(Q, \text{R}, K)$.

The discovery of spatial patterns is a descriptive mining task that aims at the detection of associations between reference objects and task-relevant objects, the former being the main subject of the description while the latter being spatial objects that are relevant for the task at hand and spatially related to the former. For instance, we may be interested in describing a given area by finding associations among large towns (*reference objects*) and spatial objects in the road network and hydrography layers (*task-relevant objects*). Some kind of taxonomic knowledge on task-relevant geographic layers may also be taken into account to get descriptions at different concept levels (*multiple-level patterns*). Also in spatial data mining, patterns can be presented in the form of rules. We search for association rules with large support and high confidence (*strong association rules*). Formally, the problem $\mathcal{P}$ is the following:

Given

- a spatial database
- a set of reference objects S
- some task-relevant geographic layers R_k, $1 \leq k \leq m$, together with spatial hierarchies defined on them,
- two thresholds for each level l in the spatial hierarchies, $minsup[l]$ and $minconf[l]$

Find strong multiple-level spatial association rules.

An instance $\imath(\mathcal{P})$ is the aforementioned discovery of associations between large towns (S) and spatial objects taken from the layers of road network (R_1) and hydrography (R_2).

An ILP method for spatial association rule mining, called SPADA, has been presented in [15]. It benefits from the available background knowledge (such as spatial hierarchies, spatial constraints and rules for spatial qualitative reasoning), systematically explores the hierarchical structure of task-relevant geographic layers and deals with numerical aspatial properties of spatial objects. Here, we refer to this work by focusing our attention on frequent pattern discovery and assuming $\imath(\mathcal{P})$ as problem instance. Search follows the principles reported in Section 2.

Data and patterns are represented in DATALOG. From the viewpoint of syntax, the language of patterns $\mathcal{L}$ can be generated by the following grammar rule in Backus-Naur format:

$$\langle S \rangle \{\langle attr(S) \rangle\}_{0\ldots n} \{\langle rel(S, R_k) \rangle \langle R_k \rangle \{\langle attr(R_k) \rangle\}_{0\ldots n}\}_{1\ldots n}$$

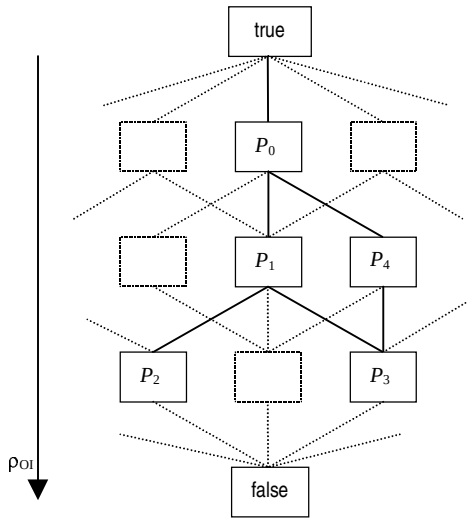

Figure 1. Fragment of $\preceq_{\text{OI}}$-ordered pattern space.

which emphasizes the property of linkedness that patterns must satisfy. Generally speaking, given a set O of objects, the categories $\langle O \rangle$, $\langle attr(O) \rangle$, and $\langle rel(O, \_) \rangle$ refer to atoms that represent a classification, an attributive feature, and a relational feature of any object $o \in O$, respectively. The trivial pattern

$$P_0 = is\_a(A, large\_town)$$

is the key atom supplied by S. From the viewpoint of semantics, what distinguishes a spatial pattern is the presence of at least one atom $\langle rel(S, R_k) \rangle$ that expresses a spatial relation, e.g. $intersects(A, B)$, like in

$$P_1 = is\_a(A, large\_town), intersects(A, B), is\_a(B, road)$$

which can be generated when solving $\imath(\mathcal{P})$ with respect to the alphabet $\mathcal{A} = \{intersects(S, R_1), adjacent\_to(S, R_2)\}$ for $\mathcal{L}$. In order to illustrate θ_{OI}-subsumption, let us consider the following pattern

$$P_2 = is\_a(A, large\_town), intersects(A, B), is\_a(B, road), \\ intersects(A, C), is\_a(C, road)$$

that also belongs to $\mathcal{L}$. It is straightforward to check that P_2 θ_{OI}-subsumes P_1 but not viceversa. Since $P_2 \prec_{\text{OI}} P_1$, P_2 can be generated by computing $\rho_{\text{OI}}(P_1)$. Note that the patterns P_1 and P_2 are equivalent under θ-subsumption, thus causing the generation of improper refinements that yield redundancy in the result. This is not desirable since more elements of the domain may be involved in P_2 than in P_1. Indeed, by assuming a natural language interpretation, P_1 and P_2 state that *'a large town intersects a road'* and *'a large town intersects two (distinct) roads'* respectively. Another possible refinement of P_1 w.r.t. $\mathcal{A}$ is the following pattern

$$P_3 = is\_a(A, large\_town), intersects(A, B), is\_a(B, road), \\ adjacent\_to(A, C), is\_a(C, water)$$

Let us suppose that the pattern

$$P_4 = is\_a(A, large\_town), adjacent\_to(A, B), is\_a(B, water)$$

has been generated while refining P_0 and found infrequent. Since $P_3 \preceq_{\text{OI}} P_4$, Proposition 4.3 holds and causes P_3 to be pruned. The portion of space that highlights the relations between P_0, P_1, P_2, P_3 and P_4 is reported in Figure 1.

In SPADA the assumption of Object Identity is currently implemented as language bias. The system relies on a more sophisticated rewriting rule that adds inequality atoms to candidate patterns any time there is a need for distinguishing between multiple instances of the same class of spatial objects (e.g. $road$). For instance, the pattern P_2 is rewritten as

$$P_2^{\text{OI}} = is\_a(A, large\_town), intersects(A, B), is\_a(B, road), \\ intersects(A, C), is\_a(C, road), C \neq B.$$

It is noteworthy that patterns of interest to SPADA are a tractable case of conjunctive queries with inequalities. Let us consider the hypergraph associated to P_2^{OI} and reported in Figure 2. It shows that P_2^{OI} is an acyclic conjunctive query with inequalities. Indeed, the inclusion of edges corresponding to inequality atoms (dashed hyperedges) destroys acyclicity of P_2. It has been proved that the class of acyclic conjunctive queries with inequalities is fixed parameter (f.p.) tractable, both with respect to the query size and the number of variables as the parameter [18]. Furthermore, such queries can be evaluated in f.p. polynomial time in the input and the output. Despite f.p. tractability of DATALOG$^{\text{OI}}$ patterns, we maintain that the performance of SPADA can be further improved by implementing Object Identity as search bias. Indeed, Definition 3.1 suggests to embed the evaluation of inequalities in the computation of the substitutions.

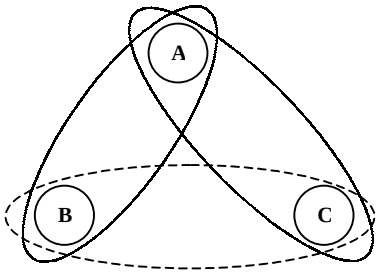

Figure 2. Hypergraph of an acyclic conjunctive query with inequalities.

6 CONCLUSIONS

Many tasks of multi-relational data mining are not feasible or could be tackled by consuming a huge amount of computational resources. The discovery of frequent DATALOG patterns is among them. Nevertheless, biases can help to solve these hard data mining problems at least in some restricted and yet meaningful cases. This work is an effort in this direction. The object identity assumption does not affect the expressive power of DATALOG, but reduces the complexity of refinement operators for searching spaces of DATALOG patterns. In the context of frequent pattern discovery, the generality order based on θ_{OI}-subsumption seems to be promising. Indeed we have proven the existence of ideal refinement operators for $\preceq_{\text{OI}}$-ordered spaces and the monotonicity of $\preceq_{\text{OI}}$ with respect to pattern support. Furthermore, this ordering has turned out to be appropriate for organizing efficiently the space of DATALOG patterns over structured domains. Features of such spaces have been illustrated by means of an example of frequent pattern discovery in spatial data. We have made reference to an ILP method for mining spatial association rules, called SPADA, which currently implements object identity as language bias. For the future, we plan to implement object identity in SPADA as search bias and conduct experiments to evaluate the performance of the downward refinement operator ρ_{OI}.

REFERENCES

[1] R. Agrawal and R. Srikant, 'Fast Algorithms for Mining Association Rules', in *Proceedings of the 12th VLDB Conference*, (1994).

[2] L Badea and M. Stanciu, 'Refinement operators can be (weakly) perfect', in *Proceedings of the 9th International Workshop on Inductive Logic Programming*, eds., S. Džeroski and P. Flach, volume 1634 of *LNAI*, 21–32, Springer, (1999).

[3] S. Ceri, G. Gottlob, and L. Tanca, *Logic Programming and Databases*, Springer, 1990.

[4] L. Dehaspe and H. Toivonen, 'Discovery of frequent DATALOG patterns', *Data Mining and Knowledge Discovery*, **3**, 7–36, (1999).

[5] F. Esposito, N. Fanizzi, S. Ferilli, and G. Semeraro, 'A generalization model based on OI-implication for ideal theory refinement', *Fundamenta Informaticae*, **47**, 15–33, (2001).

[6] F. Esposito, N. Fanizzi, S. Ferilli, and G. Semeraro, 'OI-implication: Soundness and refutation completeness', in *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, ed., B. Nebel, pp. 847–852, Seattle, WA, (2001).

[7] N. Fanizzi, S. Ferilli, G. Semeraro, and F. Esposito, 'On the decidability of OI-implication', in *Proceedings of the Work-in-Progress Track at the 11th International Conference on Inductive Logic Programming*, eds., Céline Rouveirol and Michèle Sebag, pp. 27–38, (2001).

[8] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, AAAI Press/The MIT Press, 1996.

[9] M.R. Garey and D.S. Johnson, *Computers and Intractability - A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.

[10] G. Gottlob, 'Subsumption and implication', *Information Processing Letters*, **24**(2), 109–111, (1987).

[11] D. Haussler, 'Learning conjunctive concepts in structural domains', *Machine Learning*, **4**(1), 7–40, (1989).

[12] N. Helft, 'Inductive generalization: A logical framework', in *Progress in Machine Learning - Proceedings of EWSL87: 2nd European Working Session on Learning*, eds., I. Bratko and N. Lavrač, pp. 149–157, Wilmslow, U.K., (1987). Sigma Press.

[13] J.-U. Kietz, 'A comparative study of structural most specific generalisations used in machine learning', in *Proceedings of the 3rd International Workshop on Inductive Logic Programming*, pp. 149–164, Ljubljana, Slovenia, (1993). J. Stefan Institute Technical Report IJS-DP-6707.

[14] A.C. Klug, 'On conjunctive queries containing inequalities', *Journal of ACM*, **35**(1), 146–160, (1988).

[15] D. Malerba and F.A. Lisi, 'An ILP Method for Spatial Association Rule Mining', in *Notes of the ECML/PKDD 2001 Workshop on Multi-Relational Data Mining*, eds., A. Knobbe and D. van der Wallen, pp. 18–29, (2001). (http://www.informatik.uni-freiburg.de/ ml/ecmlpkdd/WS-Proceedings/w06/lisi.pdf).

[16] H. Mannila and H. Toivonen, 'Levelwise search and borders of theories in knowledge discovery', *Data Mining and Knowledge Discovery*, **1**(3), 241–258, (1997).

[17] S.-H. Nienhuys-Cheng and R. de Wolf, *Foundations of Inductive Logic Programming*, volume 1228 of *LNAI*, Springer, 1997.

[18] C.H. Papadimitriou and M. Yannakakis, 'On the complexity of database queries', in *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 12-14, 1997, Tucson, Arizona*, pp. 12–19. ACM Press, (1997).

[19] R. Reiter, 'Equality and domain closure in first order databases', *Journal of ACM*, **27**, 235–249, (1980).

[20] G. Semeraro, F. Esposito, D. Malerba, N. Fanizzi, and S. Ferilli, 'A logic framework for the incremental inductive synthesis of Datalog theories', in *Proceedings of 7th International Workshop on Logic Program Synthesis and Transformation*, ed., N.E. Fuchs, volume 1463 of *LNCS*, 300–321, Springer, (1998).