Towards Situation Discovery for Clustering Instances

Luis Palacios[1,2], Yue Ma[1], Chantal Reynaud[1], Gaëlle Lortal[2]

[1] Laboratoire de Recherche en Informatique, Université Paris-Sud, France
[2] Thales TRT Palaiseau, France
{luis.palacios,ma,rc}@lri.fr, gaelle.lortal@thalesgroup.com

In this paper, we are interested in the problem of identifying a set of individuals in an ontology can be *distinguished* from the rest in the sense that, for such a set, we can find a proper formal definition, called a *situation*, that covers merely the individuals in this set. In the form of a concept description, a situation gives a detailed characterization of a set of instances, thus serving as an explanation for the set. This is useful for problems such as clustering instances in an explainable way. We formally define the problem of finding situations in Description Logics (DLs) and discuss a first algorithm for this problem.

Concept Learning in DLs is to learn definitions of classes from existing ontologies and instance data. Most methods in this area are based on Inductive Logic Programming techniques [5,3]. Distinct from these approaches [3,4] where a set of instances is given as positive examples of the target concept, the challenge of learning situations consists in discovering distinguishable sets of instances among exponentially many. Moreover, our work deviates from standard clustering problems in that our aim is to cluster individuals in such a way that we can have a DL concept definition that describes exactly these individuals but no others.

Approach Definitions Given an ontology $\mathcal{O}$, we formally define our problem in terms of *situations* in an ontology, for which we need to first define representative concepts.

Definition 1 (A representative concept). *Let Δ be a set of all individuals in an ontology $\mathcal{O}$, and let $\mathcal{X} \subseteq \Delta$. For a concept C, we say that $\mathcal{X}$ is* represented *by C (or C represents $\mathcal{X}$) w.r.t. $\mathcal{O}$ and Δ, if: (1) $C(x)$ holds for all $x \in \mathcal{X}$, i.e. $\mathcal{O} \models C(x)$, and (2) $C(y)$ does not hold for any $y \in \Delta \setminus \mathcal{X}$, i.e., $\mathcal{O} \not\models C(y)$. If there exists a concept C that represents $\mathcal{X}$, we say that $\mathcal{X}$ is representable.*

Example 1 (Representative Concept). Consider the set of individuals $\Delta = \{f_1, f_2, f_3\}$, a subset of individuals $\mathcal{X} = \{f_1, f_2\}$, and the following ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T} = \{C \equiv \exists r.\top\}$ and $\mathcal{A} = \{A(f_1), B(f_2), E(f_3), r(f_1, f_3), r(f_2, f_3)\}$. It holds that $\mathcal{X}$ is represented by C. But there is no $\mathcal{ELO}$ concept that can represent the set $\{f_2, f_3\}$.

A set of individuals that can be distinguished have to share some common properties merely among them, which are made explicit by the concepts that represent them. For example, the individuals f_1 and f_2 share the property of being connected to some individual via r role.

Lemma 1. *Given an ontology $\mathcal{O}$ and a set Δ of individuals, we have: (1) $\top$ is a representative concept for Δ and (2) $\bot$ is a representative concept for $\emptyset$.*

Proposition 1. *Given an ontology $\mathcal{O}$, the set Δ of individuals in $\mathcal{O}$, and $\mathcal{X} \subseteq \Delta, \mathcal{X}' \subseteq \Delta$. If $\mathcal{X}$ and $\mathcal{X}'$ are representable, then $\mathcal{X} \cap \mathcal{X}'$ is representable in $\mathcal{ELO}$.*

However, the conclusion is no longer true for the union ($\cup$) or set complement ($\setminus$).

The next lemma tells that any concept naturally represents a special set of individuals.

Lemma 2. *Let C be a concept, $\mathcal{O}$ be an ontology and Δ be the set of individuals in $\mathcal{O}$. Then C represents the set $S = \{x \in \Delta \mid \mathcal{O} \models C(x)\}$.*

Note that when a concept represents an empty set of individuals, it means that this concept is irrelevant to characterize the properties of the individuals from this ontology.

Proposition 2. *Given an $\mathcal{ELO}$ ontology $\mathcal{O}$, a set Δ of individuals, a concept C and a set $\mathcal{X} \subseteq \Delta$, we consider the following two decision problems: (1) Representability: Does C represent $\mathcal{X}$ w.r.t. $\mathcal{O}$? (2) Representability$_n$: For an integer $n > 0$, is there a concept C with $|C| < n$ that represents $\mathcal{X}$ w.r.t. $\mathcal{O}$?*

*We have Representability is in **PTime** and Representability$_n$ is in **ExpTime**.*

The concepts representing $\mathcal{X}$ are equivalent in the sense of their instances. We call each of these equivalence classes a *situation* in $\mathcal{O}$.

Definition 2 (Situation in $\mathcal{O}$). *Given an ontology $\mathcal{O}$, a set Δ of individuals in $\mathcal{O}$, and a set $\mathcal{X} \subseteq \Delta$, a* situation *for $\mathcal{X}$ w.r.t. $\mathcal{O}$ is: $||\mathcal{X}||^{\mathcal{O}}_{\Delta} = \{C \mid C$ represents $\mathcal{X}$ w.r.t.$\mathcal{O}$ and $\Delta\}$.*

Intuitively, a situation in $\mathcal{O}$ explicitly characterizes, via concept descriptions, a given set of individuals in the ontology.

Proposition 3. *Given an ontology $\mathcal{O}$ and the set Δ of individuals in $\mathcal{O}$, we assume $\mathcal{O} \models A \equiv B$. Then $A \in ||\mathcal{X}||^{\mathcal{O}}_{\Delta}$ if and only if $B \in ||\mathcal{X}||^{\mathcal{O}}_{\Delta}$ for any $\mathcal{X} \subseteq \Delta$. But the inverse does not hold.*

Next we define the problem of discovering situations for a set of instances w.r.t. $\mathcal{O}$.

Definition 3 (Situation discovery problem). *Let $\mathcal{O}$ be an ontology and Δ a set of individuals in $\mathcal{O}$. For $\mathcal{X} \subseteq \Delta$, the situation discovery problem is to compute the following set: $\Xi(\mathcal{X}) = \{\mathcal{X}_1, \ldots, \mathcal{X}_n \mid \mathcal{X}_i \subseteq \mathcal{X}, ||\mathcal{X}_i||^{\mathcal{O}}_{\Delta} \neq \emptyset\}$. That is, to find all the subsets of $\mathcal{X}$ that are representable w.r.t. $\mathcal{O}$.*

By Lemma 1, it is easy to see that $\emptyset$ is representable by $\bot$, therefore $\emptyset \in \Xi(\mathcal{X})$.

The following result shows the monotonicity of the set of situations with an increase in domain elements. However, a set that is representable is not necessarily distinguishable any more if more elements are added. But a concept that can represent a set of instances still represents some (probably a different) set.

Proposition 4. *Let $\mathcal{O}$ be an ontology and Δ be a set of individuals in $\mathcal{O}$. Consider $\Delta_1 \subseteq \Delta$. Suppose that $\mathcal{X} \in \Xi(\Delta_1)$ is represented by a concept A w.r.t. $\mathcal{O}$ and Δ_1. Then we have: (1) $||\mathcal{X}||^{\mathcal{O}}_{\Delta} \subseteq ||\mathcal{X}||^{\mathcal{O}}_{\Delta_1}$ (2) $\mathcal{X}$ is not necessarily representable w.r.t. Δ. (3) The concept A still represents some set of individuals $\mathcal{X}'$, that is, $\mathcal{X}' \in \Xi(\Delta)$.*

An algorithm to compute situations in $\mathcal{ELO}$ [1,2] is given in the long version of our paper. The intuition is that: we first construct a refinement operator α to find the most specific concept, called MSR, that best represents all instances in a given set $\mathcal{X}$. Then any refinement of such MSR obtained by the operator α w.r.t. some $x \in \mathcal{X}$ will produce a new situation characterized by a concept refined from the MSR for $\mathcal{X}$ by the operator α. By iterating this process, the situations in $\mathcal{O}$ can be discovered.

A prototype to support diagnosis in the avionics sector has already been implemented, and a real application in industry of this work will be discussed in a separate paper.

References

1. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of IJCAI'05*, 2005.
2. F. Baader, I. Horrocks, C. Lutz, and U. Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.
3. N. Fanizzi, C. d'Amato, and F. Esposito. Dl-foil concept learning in description logics. In *International Conference on Inductive Logic Programming*, pages 107–121. Springer, 2008.
4. J. Lehmann and P. Hitzler. Concept learning in description logics using refinement operators. *Machine Learning*, 78(1-2):203, 2010.
5. S.-H. Nienhuys-Cheng and R. De Wolf. *Foundations of inductive logic programming*, volume 1228. Springer Science & Business Media, 1997.