Leveraging a domain ontology in (neural) learning from heterogeneous data

Tomas Martin[a], Petko Valtchev[a] and Abdoulaye Baniré Diallo[a]

[a]Centre de Recherche en Intelligence Artificielle (CRIA), UQAM, Montreal, Canada

Abstract

Injecting domain knowledge into a neural learner to alleviate reliance on high-quality data and improve explainability is a rapidly expanding research trend. While most of the effort focused on regular topology formats such as sequences and grids, we consider graph datasets. Moreover, instead of knowledge graph (KG) embedding that underlies the majority of graph-centered methods, we propose a dedicated pattern mining-based approach. As our patterns are ontologically-generalized, they achieve multiple objectives: domain knowledge infusion, generalization capacity enhancement, interpretability, etc.

Keywords

Domain Ontology, Symbolic methods, Sub-symbolic methods, Neural networks, Graph pattern mining

1. Introduction

Nowadays, implementing decision support systems to help practitioners in complex activities has become a current practice in many fields. Many of these systems, traditionally, have used machine learning to predict the outcome of a specific problem in the user's environment and use the prediction to suggest concrete actions. Deep learning has arrived with a promise to expand the areas where automation is successfully applied in problem solving, hence the expectation for high-quality decision support to profuse.

However, predicting or learning representations on such complex domains typically requires the availability of large amounts of data of sufficiently high quality. Unfortunately, in practice, such datasets are not always readily available. Conversely, often quantities of machine-readable expert knowledge do exist, and could potentially complement already available data. Since they reflect at least partly the expertise that underlies decision making in the field, it is only natural to look for ways to inject that knowledge into the learning process to try to guide it and compensate the scarceness of high-quality data.

For several decades, ontologies, i.e. structured representation of domain concepts and their relations [1], have been promoted as the appropriate tool for making domain knowledge available for machine processing [2].

Proceedings of the CIKM 2020 Workshops, October 19-20, Galway, Ireland.

EMAIL: martin.tomas@courrier.uqam.ca (T. Martin); valtchev.petko@uqam.ca (P. Valtchev); diallo.abdoulaye@uqam.ca (A.B. Diallo)

2. Motivation

Daily activities in agro-industrial sector, e.g. a maintenance of a dairy farm, like those in other areas related to life sciences, generate large amounts of data. The underlying data sources reflect complementary aspects such as farm yield, environment, animal health, genetics, etc. The recent trend of *precision(-based) agriculture* looks at exploiting this data to support the decision making of domain stake-holders [3]: farmers, agronomists, dairy companies, insurers, etc.

Yet, in order to be effective, any recommendation will have to reflect existing practices and, more generally, at least partly reflect the general knowledge from the domain. For instance, at the end of each lactation a cow gets dry for a while. Yet there is no a straightforward way to train a neural model on milk yield data: The ensuing abrupt drop in milk yield is hard to digest for, at least, the most popular deep learning architectures [4]. Indeed, these models do not seem to properly grasp the dynamics in a cow life-cycle, e.g. lactation, calvings, drying, etc.

While there are still work-arounds left to explore, one legitimate research question is whether injecting some domain knowledge would help here. In a broader approach, we investigate the impact of feeding complementary data, e.g. on genetics and animal health, and organizing the overall dataset under a domain ontology (DO) providing additional descriptive knowledge.

While supplementing a neural learner with domain knowledge stemming from an ontology is definitely appealing, it is also a challenging task, mainly due to the "impedance mismatch", i.e. the divergence in the respective levels of knowledge expression and manipulation [5].

3. Current State Of The Art

While symbolic representations, as a way to capture knowledge, have clearly dominated the AI field since its inception, recently sub-symbolic ones –in the form of trained neural networks– have rapidly gained in popularity and use [6]. By trading discrete and man-made (i.e. modelling) entities of the former for more machine-made (artificial) and loosely defined "patterns", the later breaks free of prior knowledge in order to, arguably, benefit for a more powerful yet difficult to interpret representation language. At its core, information is distilled throughout a network as a set of waves (or pulses) representing captured knowledge.

In a broader scope, injecting domain knowledge in a machine learning process has been extensively researched and proven helpful in many practical situations [7]. More recently, since deep learning has moved centre stage, the focus has shifted on making neural networks collaborate with symbolic knowledge sources, mostly knowledge graphs (KG) and, somewhat more modestly, domain ontologies. In [5], the authors propose a classification of methods for feeding domain knowledge to artificial neural networks (ANNs), in particular, to deep ones. Their own proposal, called knowledge-infused learning (K-IL), addresses a variety of issues with ANN, in particular, reliance on large datasets of sufficient quality, biases in training data selection, complexity, etc. The proposed answer represents a spectrum of fine-grained transformations of the ANN architecture reflecting the content of a KG that range from correcting the loss function to modifying the propagation through the network via connection weights.

The broader trend of using KG in the form of embeddings –of vertices, edges or both– into a vector space, e.g. in order to support various natural language processing tasks, has been highly prolific for almost a decade now (see [8] for a somewhat outdated survey). While initial work by Bordes *et al.* [9, 10] looked at embedding a triple from a KG using energy-based methods to force plausible combinations of component embeddings, in [11] the focus is exclusively on vertices, i.e. domain entities. The proposed *RDF2vec* method generates a set of entity sequences, through random walks and iterative neighbourhood encoding techniques, which are then fed to word embedding methods. In a medical context, the authors of [12], present a somewhat different approach toward leveraging a KG in neural learning: In order to assess patient risk from a series of health events, they translate the neighbourhoods of an event-centred KG into attention filters for a LSTM-based ANN.

Overall, KG embedding is not straightforwardly portable to DOs as in the case of KGs amalgamation is favoured by them being on the same abstraction level as the training data. In contrast, classes and properties from a DO represent abstractions, i.e. sets of data objects and object-to-object links, hence the apparent mismatch with the instance-centered *modus operandi* of an ANN. Yet given the strive for (proper) generalization in ANNs, the ontological structure, with its capacity to generalize along expert-validated conceptual hierarchies (and property ones, for that matters), is a natural ally.

Nonetheless, a few studies have tackled the exploitation of generic knowledge from a DO in neural learning. For instance, in [13], the authors exploit a DO (a topic hierarchy, in actuality) of sound events to enhance a neural classifier. They propose to replicate the hierarchical structure of the DO in the ANN topology by: (1) allotting a layer per level in the *is-a* hierarchy and (2) enforcing fixed distance values between pairs of example embeddings, which roughly translate the examples' topological distance within the hierarchy. In a similar vein, the method in [14] simulates the topology of the DO graph in learning the representations of its classes and properties. A class is thus reduced to the union of its data properties, those of its sub-classes and of related classes. On a following step, the method learns instance representations, from the representations in the DO, and uses them in behaviour prediction.

Besides, different ways of making ontologies and ANN collaborate have been explored, e.g. ontology learning [15] or neural reasoning with ontologies [16]. For example, [17] approaches the latter task as a translation problem with noisy-data.

On a broader scope, while feature vector-oriented ANNs have shined on sequence- and grid-shaped data, i.e. with values arguably more important than –the highly regular– topology, graph data, due to its inherent sparsity, requires more fine-grained generalization (i.e. chemical functional groups, biological pathways, telecom network configurations, etc.). Graph Convolutional Neural Networks (GCNN) constitute a recent and promising approach for learning such regularities [18, 19]. By applying convolution layers on top of each other, they recursively aggregate n^{th}-order neighbourhood information from the graph and can achieve good generalization on such datasets. Yet due to their inherent bias toward frequent regularities, the very local, rare and context-specific ones will arguably be missed. And, clearly, this behaviour compounds whenever quality data prove scarce.

Beyond pure generalization capabilities, dealing with actionable and surprising patterns mixing different abstraction levels is to be expected: conceptually, a sequentially layered generalization procedure might not prove enough to extract such regularities.

Taking a step back, we consider three ongoing trends each one following a founding principle: (i) K-IL supports the use of external domain knowledge as a way to bring improvements on both predictive power and explainability; (ii) G(C)NN approaches consider preserving topology as critical when working on graph data; (iii) Contextual mechanisms lead to better results on both static (e.g. text translation) and dynamic (e.g. user behaviour) predictive tasks [20].

To the best of our knowledge, no prior research has jointly addressed the above three concerns. Here, we present a novel approach for learning in complex domains that does this. By delegating most of the knowledge/pattern extraction effort to a dedicated symbolic method, we subsequently feed those patterns as input features to an architecture-agnostic neural learner. Thus, offering ontologically-generalized graph-shaped features *a priori* overlapping with a GNN's convolved high-level patterns. Nevertheless, ontological based generalization plays nice with robustness properties: by going beyond mere boolean encoding of attributes (i.e. vertices, edges) with the help of a DO's conceptual structure, it helps the symbolic learning to not fall into overfitting pitfalls.

4. Vision & Approach details

First, by bringing some ontological concepts into the data as higher-order regularities we aim to make explicit the shared conceptual structure that remains invisible in the raw data. The rationale therefore is while exact values may mismatch, more abstract types describing those values would coincide. For instance, two groups of lactating cows may be treated for mastitis –a common bacterial infection of the udder– by using amoxicillin and penicillin, respectively. Now knowing that these are both β-lactams helps extend a common sub-graph comprising, at least, nodes for cow and mastitis, with a further node for that class of antibiotics. Obviously, this increase in the shared portions of the data graphs w.r.t. to their raw versions would not be possible without an ontology covering the antibiotics. In a more general vein, inserting typing information and property generalizations helps reveal hidden commonalities that would not be easily spotted neither by a human expert, neither by a sub-symbolic learner.

Next, our goal is to find all significant fragments of such shared structure in a set of data graphs. These include abstractions on both vertices (ontology classes) and edges (ontology properties). As an illustrative example, Figure 1 presents a possible pattern, illustrating possible causes for a shorter than average first lactation of a young cow. Here, frequently, both the young cow and its ancestor have been treated with different kind of antibiotics.

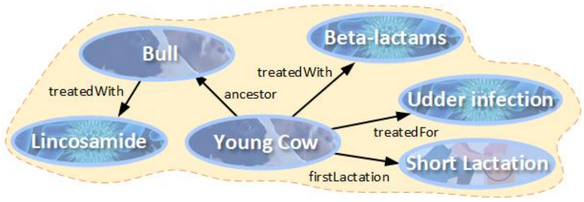

Figure 1: An example of ontological graph pattern.

The resulting graph structure can be qualified as doubly-labelled, i.e. on both vertices and edges, multigraphs. Practically, we first discover the interesting patterns and then, in a feature engineering step, we assign them as higher-level descriptors of the matching data graphs.

Another palpable advantage of using the ontology-based patterns is that they offer an integrated view of the shared structure: Edges standing for properties connect class vertices, thus providing context to each of them. On the other hand, pattern components, as well as whole patterns, pertain to potentially varying abstraction levels.

More concretely, Figure 2 details our hybrid strategy where graph patterns are first mined (step 3) and then fed (step 5) into a neural network (step 6) with graph data supported by a DO as our main input (steps 2 and 1 respectively), complementary to regular tabular data (step 4). The mining of ontological patterns from that graph data uses the domain ontology as backbone for the exploration (e.g. ontological types, resources as vertices and properties as edges). Between steps 3 and 5, an optional post-processing step can also further refine the patterns to emphasize contrasts, or synthesize by approximating, if required by the learning task. The resulting ontological graph patterns allow the original data to be encoded with the new features supported by domain knowledge before feeding the augmented data to the ANN (steps 5 and 6).

Pattern mining [21], aims at extracting recurrent data fragments, a.k.a. *patterns*, capturing the most relevant information possible. A mining task is defined by a pair of languages, one for data records and one for patterns, and a relevance (interestingness) criterion. The typical criterion is frequency of appearance, but other criteria such as utility or some domain-related ones

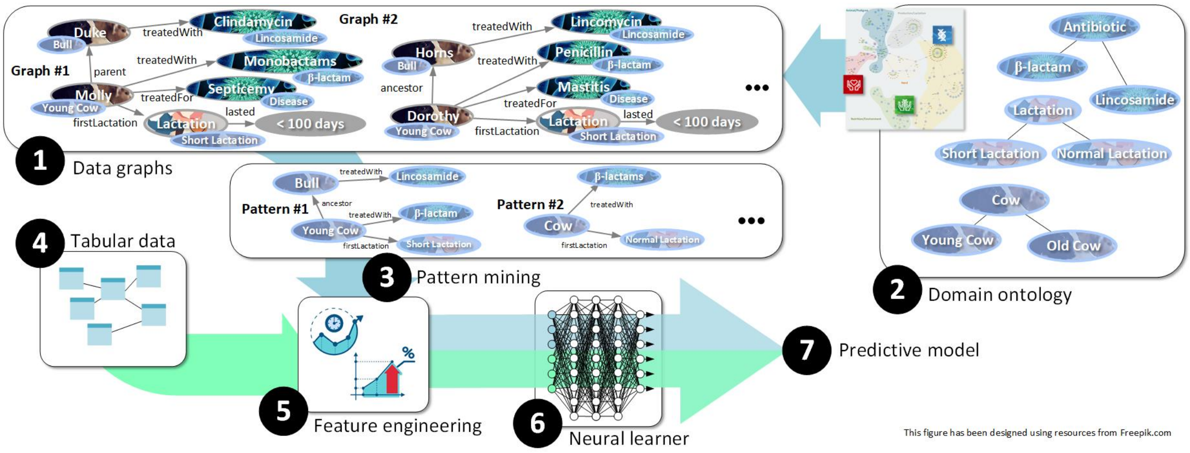

Figure 2: High-level view of our hybrid learning process.

are possible. Moreover, an effective mining method requires a general strategy for pattern space traversal and a technique to perform a pattern-to-data record matching. The later revolves around computing a variation of sub-graph isomorphism, here integrating the conceptual structure of an ontology. Typically, the former entails defining a spanning tree of the pattern space and a canonical representation of graph patterns to avoid generating multiple copies of the same pattern [22].

Ontologies have been used in frequent pattern mining to guide the exploration of the complex pattern spaces such as sequences of objects or simple graphs for some time [23, 24]. For predictive tasks exploiting graph pattern mining a few successful techniques exist such as quantitative structure-activity relationships (QSARs) [25], optimizing objective functions [26] or dedicated pattern ranking metrics [27] exploiting external domain knowledge. While ontologies and patterns have been combined before, to the best of our knowledge, no mining method has targeted data of such complexity.

The downside of the approach is its sensibility on the pattern frequency threshold and the related potential combinatorial explosion in the result. While this is a serious cost issues with graph patterns, possible mitigation strategies exist, e.g. using condensed representations thereof such as *closed patterns* [28].

Overall, expected **immediate benefits** of the ontological knowledge injection into the neural learning process include higher accuracy in predictive architecture and faster convergence.

References

[1] T. Gruber, et al., A translation approach to portable ontology specifications, Knowledge acquisition 5 (1993) 199–220.

[2] F. Kramer, T. Beißbarth, Working with ontologies, in: Bioinformatics, Springer, 2017, pp. 123–135.

[3] A. Barbosa, et al., Modeling yield response to crop management using convolutional neural networks, Computers and Electronics in Agriculture 170 (2020) 105197.

[4] C. Frasco, et al., Towards an Effective Decision-making System based on Cow Profitability using Deep Learning:, in: Proc. of the 12th ICAART, Valletta, Malta, 2020, pp. 949–958.

[5] A. Sheth, et al., Shades of knowledge-infused learning for enhancing deep learning, IEEE Internet Computing 23 (2019) 54–63.

[6] Y. Bengio, et al., Representation learning: A review and new perspectives, IEEE Transactions on PAMI 35 (2013) 1798–1828.

[7] D. Dou, et al., Semantic data mining: A survey of ontology-based approaches, in: IEEE ICSC 2015, IEEE, 2015, pp. 244–251.

[8] Q. Wang, et al., Knowledge graph embedding: A survey of approaches and applications, IEEE TKDE 29 (2017) 2724–2743.

[9] A. Bordes, et al., Translating embeddings for modeling multi-relational data, in: Advances in neural information processing systems, 2013, pp. 2787–2795.

[10] A. Bordes, et al., A semantic matching energy function for learning with multi-relational data:

Application to word-sense disambiguation, Machine Learning 94 (2014) 233–259.
[11] P. Ristoski, H. Paulheim, Rdf2vec: Rdf graph embeddings for data mining, in: ISWC, Springer, 2016, pp. 498–514.
[12] C. Yin, et al., Domain knowledge guided deep learning with electronic health records, in: 2019 IEEE ICDM, IEEE, 2019, p. 738–747.
[13] A. Jiménez, et al., Sound event classification using ontology-based neural networks, in: In Proc. of the Annual Conference on NeurIPS, 2018.
[14] N. Phan, et al., Ontology-based deep learning for human behavior prediction with explanations in health social networks, Information sciences 384 (2017) 298–313.
[15] M. Casteleiro, et al., Ontology learning with deep learning: a case study on patient safety using pubmed., in: SWAT4LS, 2016.
[16] P. Hohenecker, T. Lukasiewicz, Deep learning for ontology reasoning, CoRR (2017).
[17] B. Makni, J. Hendler, Deep learning for noise-tolerant rdfs reasoning, Semantic Web 10 (2019) 823–862.
[18] K. Xu, et al., How powerful are graph neural networks?, arXiv:1810.00826 [cs, stat] (2019). URL: http://arxiv.org/abs/1810.00826.
[19] H. Yuan, et al., Xgnn: Towards model-level explanations of graph neural networks, 26th ACM SIGKDD (2020) 430–438.
[20] A. Vaswani, et al., Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
[21] C. Aggarwal, J. Han, Frequent Pattern Mining, 2014 ed., Springer, 2014.
[22] X. Yan, J. Han, gspan: Graph-based substructure pattern mining, in: Proc. of the IEEE ICDM 2002, IEEE, 2002, pp. 721–724.
[23] M. Adda, et al., A framework for mining meaningful usage patterns within a semantically enhanced web portal, in: Proc. of the 3rd C* CCSE, 2010, pp. 138–147.
[24] A. Cakmak, G. Ozsoyoglu, Taxonomy-superimposed graph mining, in: Proc. of the 11th EDBT, ACM, 2008, pp. 217–228.
[25] S. Nijssen, J. Kok, Frequent graph mining and its application to molecular databases, in: IEEE Transactions on Systems, Man and Cybernetics, volume 5, IEEE, 2004, pp. 4571–4577.
[26] H. Saigo, et al., gboost: a mathematical programming approach to graph classification and regression, Machine Learning 75 (2009) 69–89.
[27] E. Spyropoulou, et al., Mining interesting patterns in multi-relational data with n-ary relationships, in: International Conference on Discovery Science, Springer, 2013, pp. 217–232.
[28] X. Yan, J. Han, CloseGraph: mining closed frequent graph patterns, in: Proc. of the 9th ACM SIGKDD, ACM, 2003, pp. 286–295.