Multi-Label Learning with a Cone-Based Geometric Model

Mena Leemhuis[1][0000−0003−1017−8921], Özgür L. Özçep[1][0000−0001−7140−2574], and Diedrich Wolter[2][0000−0001−9185−0147]

[1] University of Lübeck, Lübeck, Germany
mena.leemhuis@student.uni-luebeck.de
oezcep@ifis.uni-luebeck.de
[2] University of Bamberg, Bamberg, Germany
diedrich.wolter@uni-bamberg.de

Abstract. Recent approaches for knowledge-graph embeddings aim at connecting quantitative data structures used in machine learning to the qualitative structures of logics. Such embeddings are of a hybrid nature, they are data models that also exhibit conceptual structures inherent to logics. One motivation to investigate embeddings is to design conceptually adequate machine learning (ML) algorithms. This paper investigates a new approach to embedding ontologies into geometric models that interpret concepts by closed convex cones. As a proof of concept this cone-based embedding was implemented in a ML algorithm for weak supervised multi-label learning. The system was tested with the gene ontology and showed a performance similar to comparable approaches, but with the advantage of exhibiting the conceptual structure underlying the data.

Keywords: Concept Learning · Knowledge Graph Embedding · Multi-Label Learning.

1 Introduction

Recent approaches to knowledge-graph embeddings [11] aim at linking quantitative data structures used in machine learning (ML), such as (low-dimensional) Euclidean spaces, to the qualitative structures of logics. Conceptual structures of logics like that of first-order logic (FOL) are characterized by the respective domain of models considered, specific individuals, as well as the relations and functions defined. By restricting the language of FOL, several specialized logics can be defined, each giving rise to a certain repertoire of structures that can be expressed. In this work we consider a subclass of description logics (DL) [2] that is particularly suited to the representation of concept structures. Therefore, DL presents an ideal candidate when investigating the link between data models and structures of logics. Once a link between a specific embedding and a specific logic has been established, embeddings induce logic structures in the quantitative domain, say Euclidean space.

Embeddings present a promising approach to the development of concept-level machine learning. Assume a knowledge graph is given, i.e., triples stating relations between objects, then specific concept definitions learned by means of ML techniques get enhanced by the structure of the knowledge graph. Put differently, one obtains a grounding of abstract entities mentioned in the knowledge graph that respects the relational structure of that graph. Pushing the idea even further, one may even consider embedding to go beyond capturing single knowledge graphs, but to represent whole ontologies—as is the case for the convex region based geometric models of [6].

In this paper we consider geometric models based on convex cones as possible groundings of concepts. One reason for considering convex entities is their adequacy from a linguistic-cognitive point of view to model natural concepts, as has been argued by Gärdenfors in his book on conceptual spaces [4]. We are motivated to consider cones as they enable us to define negation (using polarity [8]) which pushes forward the expressivity of structures exhibited by embeddings. We note that Gärdenfors [4, p. 202] considered the representation of negation (and quantification) as particularly difficult. Most importantly, for this paper, convex regions are computationally attractive as efficient methods from the area of convex optimization [3] become available to realizing ML algorithms.

The contribution of this paper is to show how cone-based geometric models can be used for the important ML task of (weak) supervised multi-label learning [5], while retaining the conceptual structures defined by an ontology. We present a cone-based semantics for DL ontologies defined in the language of propositional $\mathcal{ALC}$ [2]. Based on this semantics we propose a new ML method for acquiring an embedding. This paper concentrates on the application of the cone-based embedding to Machine Learning. The theoretical basis of the cone-based approach like characterizing the link of cone models to models in the sense of logics is not in the scope of this paper. For these aspects we refer the reader to our [7] which considers full $\mathcal{ALC}$ (not just propositional $\mathcal{ALC}$).

Multi-label problems are problems in which each entity has to be attached one or more labels. They may be regarded as a generalization of the ML task of classification, which assigns exactly one label to each object. The multi-labeling problem is considered to be a hard ML task [5], but is particularly important for mastering non-trivial conceptual structures: every entity may be a member of several conceptual classes.

There are several types of weak supervised learning problems. We concentrate on handling inexact data, i.e., the training data set may include labels that are not fine-grained [12]. This represents a typical case of how humans would label an entity: we may claim a lion to be a carnivore, but omit class labels such as mammal or animal. A particular feature of our cone models is their ability to express partial knowledge: elements are not required to be labeled with respect to every class. In case of the lion the ML method may thus refrain from assigning a class label like “can swim” or its negation “cannot swim”, if neither evidence is given in the training data. By linking a given ontology to the ML model by means of an embedding it is guaranteed that the result, i.e., the grounding of

entities, conforms to the ontology. Coming back to the example of animals, we would be able to guarantee that no animal will be labeled herbivore if it is known to eat other animals.

In this paper we consider a medical scenario based on the gene ontology [1] that could for example be used to make disease predictions. The ML method described in this paper demonstrates that learning with cone-based models is competitive to related approaches for multi-label learning [10]. Above all, we are able to demonstrate that a given ontology can be exploited and leads to better results in learning.

2 Preliminaries

The family of description logics is a family of variable-free fragments of FOL that are designed, in particular, for the representation of ontologies. Hence, DLs provide a good balance of expressivity and computational feasibility. They can be classified by the set of concept-constructors offered. Any DL vocabulary contains a set of constants N_c, a set of (concept names) N_C and role names (corresponding to binary relations). We consider here the propositional part of the logic $\mathcal{ALC}$ [2]. The set of Boolean $\mathcal{ALC}$ concepts C is defined according to the following context-free grammar:

$$C \to A \mid \bot \mid \top \mid C \sqcup C \mid C \sqcap C \mid \neg C, \tag{1}$$

with atomic concepts $A \in N_C$ and an arbitrary concepts C. An $\mathcal{ALC}$ interpretation $(\Delta, \cdot^{\mathcal{I}})$ consists of the domain Δ (the space of possible elements) and an interpretation function $\cdot^{\mathcal{I}}$ mapping constants to elements in Δ and concept names to subsets of Δ. The semantics of arbitrary concepts is given in Table 1.

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$

Table 1. Syntax and semantics for Boolean $\mathcal{ALC}$ for an interpretation $\mathcal{I}$

An ontology $\mathcal{O}$ is a pair $(\mathcal{T}, \mathcal{A})$. The terminological-box ($\mathcal{T}$-box) $\mathcal{T}$ contains general concept inclusions of the form $C \sqsubseteq D$ stating that C is a subconcept of D, for arbitrary concepts C and D. The assertional-box ($\mathcal{A}$-box) $\mathcal{A}$ consists of facts of the form $C(a)$, $a \in N_c$, which says that a is in the extension of C.

3 Geometric Models

In our cone-based models, Boolean $\mathcal{ALC}$ ontologies are embedded into geometric models of a Euclidean vector space with a linear product $\langle \cdot, \cdot \rangle$ that measures the similarity of vectors (representation of objects) by the cosine. The geometric model $\mathcal{I}$ represents the $\mathcal{T}$-box axioms in a geometric way and is region-based, that means in particular, when $A^{\mathcal{I}} \sqsubseteq B^{\mathcal{I}}$, then A is a subspace of B in the model. The main idea is to split the vector space into convex regions. To preserve the convexity under disjunction and negation, a special convex structure—namely an axis-aligned cone (al-cone)—is used.

Definition 1. *An al-cone is a special case of a closed convex cone. An al-cone in the n-dimensional space is of the form*

$$(X_1, ..., X_n) \text{ where each } X_i \in \{\mathbb{R}, \mathbb{R}_+, \mathbb{R}_-, \{0\}\}. \tag{2}$$

The negation of arbitrary cones X (and in particular of an al-cone), is defined by its polar cone [8] X°, which is the set of all vectors leading to a negative or zero similarity with all vectors in X.

$$X^\circ = \{v \in \mathbb{R}^n | \forall w \in X : \langle v, w \rangle \leq 0\}. \tag{3}$$

For better readability, subsequently $\mathbb{R}, \mathbb{R}_+, \mathbb{R}_-, \{0\}$ are replaced by $u, +, -, 0$.

Every concept of an ontology is assigned to an al-cone as defined in (2) with respect to the $\mathcal{T}$-box axioms. An operation on an al-cone assignment of a concept is executed dimension-wise. So, e.g., the intersection of $(+, -)$ and $(+, +)$ reduces to considering the intersection of the first components $+$ and $+$ (giving $+$) and the intersection of the second components $-$ and $+$, giving 0. The constants are placed in a region were the corresponding $\mathcal{A}$-box axioms are valid. Special cases are the top concept $\top$, represented as $\{u\}^n$ which thus covers the whole space and the bottom concept $\bot$, which is represented as the point of origin $\{0\}^n$.

A special feature of this geometric model is its ability to model partial knowledge. It is not obligatory that an element is an instance of a concept or of its negation, its assignment can also be unknown. When representing negation with polarity, any point neither contained in an al-cone A nor its polar cone A° represents an entity for which class membership of the class A is unknown.

Figure 1 is an example of a geometric model for an empty $\mathcal{T}$-box and two concepts A and B. The $\mathcal{A}$-box consists of $B(a_1), B(a_2)$ and $\neg A(a_2)$. The element a_1 is in a region where it is neither in A nor in $\neg A$.

The geometric model for a given $\mathcal{T}$-box is constructed based on the set K of all possible fully specified concepts k in the ontology. A concept is fully specified when it contains every atomic concept or its negation. The geometric model has the dimension $d = \left\lceil \frac{|K|}{2} \right\rceil$. No conjunction between fully specified concepts is possible, so every k is placed on one half-axis. The al-cone for each atomic concept can be determined by constructing the union of all k in which it appears positively. The corresponding negative concept can be found by negating the

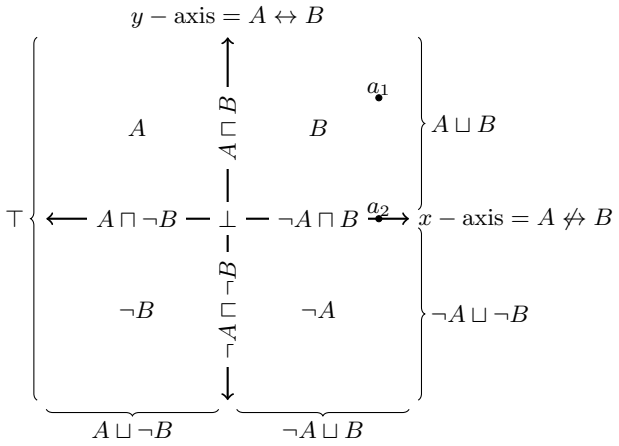

Fig. 1. Example of a geometric model

positive concept. With an empty $\mathcal{T}$-box with n concepts this results in 2^n fully specified concepts and thereby in a geometric model with $d = 2^{n-1}$ dimensions.

With a non-empty $\mathcal{T}$-box the number of possible k decreases, but it is still exponential in the most cases. The construction of the model is similar to the empty case (using the Lindenbaum-Tarski algebra induced by the $\mathcal{T}$-box).

For example the construction of the geometric model with an empty $\mathcal{T}$-Box is conducted as follows: The fully specified concepts are $\{A, B\}, \{A, \neg B\}$, $\{\neg A, B\}$, and $\{\neg A, \neg B\}$. The geometric representation of each of this fully specified concepts is placed on an individual half axis. Thus the geometric representation $\Psi(\cdot)$ is

$$\Psi(\{A, B\}) = (0, +) \tag{4}$$
$$\Psi(\{A, \neg B\}) = (-, 0) \tag{5}$$
$$\Psi(\{\neg A, B\}) = (+, 0) \tag{6}$$
$$\Psi(\{\neg A, \neg B\}) = (0, -). \tag{7}$$
$$\tag{8}$$

The representations of the other concepts are unions of the representations of the fully specified concepts and thus the resulting model is the one shown in Figure 1.

4 Multi-label Classification with a Geometric Model

The geometric model can be used in combination with the $\mathcal{A}$-box axioms given by the training data to train a classifier. To this end, every element x of the training data is mapped to a subspace of the vector space by creating a code vector $cv(x)$ with $cv(x) = \{+, -, 0, u\}^d$. In this way an element is not represented by an

individual point in space but by an al-cone. In every al-cone there could be several individuals. Thus the training elements are embedded into the geometric model, and therefore into the ontology space. The main idea is to use the knowledge incorporated in the geometric model to train a classifier for each dimension of the geometric model.

For each dimension of the code-vectors a classifier is trained separately to divide the code-vectors in classes determined by their entry in this dimension, because elements with the same region in one dimension should have some shared attributes. For each dimension $1 \leq i \leq d$ all elements are separated into classes as follows:

$$X_{pos,i} = \{x \mid cv(x)_i = +\} \tag{9}$$

$$X_{neg,i} = \{x \mid cv(x)_i = -\} \tag{10}$$

$$X_{zero,i} = \{x \mid cv(x)_i = 0\} \tag{11}$$

A code-vector with an u at dimension i is ignored.

Training of the classifier is done as follows: For each dimension i the separation of $X_{pos,i}, X_{neg,i}$, and $X_{zero,i}$ is computed as follows. When only one of the three classes in dimension i is used, then training of the classifier is not possible and all elements are assigned to the existing class. When in one dimension there are only two of the three labels chosen, then a binary classifier is trained and the third class is ignored. For three existing classes two classifiers are trained, one separating $+$ from the rest, one separating $-$ from the rest. This is explained in more detail at the end of this chapter.

For classification the classification result for the test element is determined for each dimension separately. The results of every dimension are concatenated and produce a code-vector (an al-cone) for the test element. This code vector is then placed in the geometric model. An element e is said to belong to a concept C if the code-vector of e is covered by the code-vector of C.

Our approach is used for weak supervised learning. In the weak supervised learning scenario, some labels are given, but they can be incomplete or inaccurate and it is possible that not all labels are determinable for a given element. In particular, an individual which is not labeled with a specific concept could be contained in it or its negation.

Each entry in code-vectors shows information about its properties. Our aim is to find a separation of 0 vs. $+$ vs. $-$. So why should this be possible?

First we note that an element whose code-vector is u in dimension i is ignored in this dimension because it does not represent a single piece of information. In a geometric model, every operation can be executed per dimension. By definition, in each dimension, 0 is covered by $+$ and $-$. So $+$ and $-$ are not disjoint. This means that their separability depends on the training data and is not necessarily given. Individuals which are labeled as $+$ (or $-$) could be in fact 0, but never $-$ (or $+$). But when a code vector is 0 in the specific dimension, then it stays 0 even after gaining new knowledge. This property is used for the separation task.

One option for training this separation for all three classes existing is to train two classifiers. The first one separates $+$ from the rest, the second one

separates − from the rest. Of course there are some + (−) which are not fully classified and thus wrongly appear in an area which is in fact 0. By increasing the misclassification cost for a 0 this error is mitigated. The classifier is interpreted as + only when one classifier is + and the second 0 and analogue for −. In the other cases it is classified as 0.

It follows that even the elements which are incompletely classified and thus have a + or − which in fact is a 0 give information because the probability for their appearance is higher close to their actual region.

5 Experiments

Data The method can be used for any ontology expressed in Boolean $\mathcal{ALC}$. Here the Gene Ontology (GO) [1] is used. It does not contain negation or union and is hence a directed acyclic graph. The relations of GO have not been considered. The data set for the experiments is that of Saccharomyces cerevisiae [9]. First the concepts of the training elements are extended in the way that all ancestor concepts of the given concepts are contained. Then every concept without enough elements representing it was deleted to facilitate the training process. The number of concepts was reduced to eight and for every element the most specific concepts were determined. With these concept labels the training and testing was conducted.

Implementation For classification a support vector machine with a polynomial kernel is used, because it is an established method for handling bioinformatic datasets like the one used. For the test of the method the assumption is used that not having a positive label means that it could be contained or not.

For comparison purposes we implemented the approach of Wan and Xu [10]. The approach presented in [10] does not use ontology information. It is based on a variant of the 1-vs-1-classifier. Any two concepts are compared to each other in a ternary way. A separation of elements of concept A, of concept B, and of concept $A \sqcap B$ is learned for all concepts A, B. Via a voting-scheme and a threshold the concepts of an element are obtained. This approach is used for comparison because of its high similarity to the presented approach. Its main difference is that the ontology information is not used. In this way the advantage of using this information is investigated. For better comparability—instead of the Tri-class SVM as used in their approach—we use in our implementation the SVM-architecture presented above.

Results and Discussion Classification of the test set using a six-fold cross validation results in similar performance measures for the presented and the comparison approach (see Table 2).

An advantage of the presented approach is that it can only have ontological correct results, while the other approach can result in contradictions. In every dimension more training elements and thus more information than in normal 1-vs-1 can be used, because not only elements with the same concept, but also with some similar attributes are used in the same class for training.

The similarity in the results of both approaches is caused by the simple structure of GO, which incorporates no negation or disjunction. Only subsumption needs to be considered. Without negated elements or negated concept inclusions there is no knowledge about concept exclusions and therefore the space of possible concepts per element cannot be restricted. Another reason could be the choice of the binary classifier, which could perhaps improve the generalization quality.

The presented approach has improvement potential w.r.t. the error tolerance. Concepts at the bottom of the tree have only a small al-cone where the elements could be placed. This means, that even a small misclassification in one dimension could prevent the correct classification. One possible solution is to incorporate knowledge of the certainty of the classification for each dimension. For a test element an uncertain result in a dimension could be changed to 0 to reduce its influence.

In a second experiment our method was tested with an empty $\mathcal{T}$-box. This resulted in an accuracy near to zero and demonstrates the usefulness of the ontology information for training. Without this information the knowledge about dependencies of elements cannot be used and elements which have similar attributes can not be separated from elements without similar attributes. With an empty $\mathcal{T}$-box impossible separations are tried to be learned as well. Therefore classifying a test element results in a code-vector not containing any information and thus no assigned concept. This shows that the approach can actually use the knowledge represented in the ontology.

	Accuracy	Precision	Recall
This approach	0.185 ± 0.03	0.190 ± 0.02	0.164 ± 0.03
Wan, Xu [10]	0.197 ± 0.03	0.199 ± 0.03	0.278 ± 0.08

Table 2. Results for the presented method and the approach of Wan and Xu [10]

6 Conclusion

The paper presented a proof-of-concept implementation of an algorithm for weak multi-label learning that relies on a geometric model of a Boolean $\mathcal{ALC}$ ontology. As the test results showed, having a geometric model of a non-empty $\mathcal{T}$-box leads to useful information that can be exploited for multi-labeling.

The tests were conducted for an ontology over a very weak logic (not even containing negation) to show its general applicability for weak supervised learning, but our approach is applicable for general Boolean $\mathcal{ALC}$-ontologies—whereas an approach such as that of [10] can not be used because it can not incorporate ontological knowledge. We expect to get even better results for ontologies

that allow for full negation (and disjunction) because of the higher amount of ontological information contained.

Future work concerns incorporating a method for dimension reduction in order to reduce the exponential size (w.r.t. the number of atoms in the Boolean algebra of concepts induced by the $\mathcal{T}$-box) of the geometric model. Moreover, we plan to improve the approach by using a different classifier for the dimensions instead of the ternary SVM: the idea is to consider certainty of the answer for each dimension and to improve the dimension-wise separation quality.

References

1. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene ontology: tool for the unification of biology. Nature genetics **25**(1), 25 (2000). https://doi.org/10.1038/75556
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
3. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge university press (2004)
4. Gärdenfors, P.: Conceptual Spaces: The Geometry of Thought. The MIT Press, Cambridge, Massachusetts (2000)
5. Gibaja, E., Ventura, S.: Multilabel learning: A review of the state of the art and ongoing research. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery (11 2014)
6. Gutiérrez-Basulto, V., Schockaert, S.: From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules. In: Thielscher, M., Toni, F., Wolter, F. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018. pp. 379–388. AAAI Press (2018)
7. Özçep, Ö.L., Leemhuis, M., Wolter, D.: Cone semantics for logics with negation. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20 (2020), (To appear)
8. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton, NJ (1997)
9. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning **73**(2), 185 (Aug 2008). https://doi.org/10.1007/s10994-008-5077-3
10. Wan, S.P., Xu, J.H.: A multi-label classification algorithm based on triple class support vector machine. In: 2007 International Conference on Wavelet Analysis and Pattern Recognition. vol. 4, pp. 1447–1452 (Nov 2007). https://doi.org/10.1109/ICWAPR.2007.4421677
11. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (Dec 2017). https://doi.org/10.1109/TKDE.2017.2754499
12. Zhou, Z.H.: A brief introduction to weakly supervised learning. National Science Review **5**(1), 44–53 (08 2017). https://doi.org/10.1093/nsr/nwx106