Towards Conceptual Logic Tensor Networks

Lucas Bechberger[1]

[1]Institute of Cognitive Science, Osnabrück University, Wachsbleiche 27, 49090 Osnabrück, Germany

Abstract

The symbol grounding problem refers to the necessity of grounding abstract symbolic knowledge (such as encoded in formal ontologies) in the real world through perception and action. The cognitive framework of conceptual spaces provides a potential way for solving the symbol grounding problem by proposing an intermediate representational layer: Concepts are represented by regions in low-dimensional similarity spaces, which are in turn grounded in subsymbolic processing. Logic tensor networks provide a general mechanism for learning membership functions in the presence of both bottom-up information (i.e., training examples) and top-down constraints in the form of logical rules. In this paper, we propose to combine logic tensor networks with conceptual spaces in order to ground predicates from the symbolic layer in conceptual regions while taking into account logical constraints from abstract background knowledge. We discuss several potential membership functions for concepts and argue that this approach can be used to provide a cognitive grounding for formal ontologies.

Keywords

Conceptual Spaces, Logic Tensor Networks, Symbol Grounding, Cognitive AI, Neurosymbolic AI

1. Introduction

Formal ontologies provide us with ways of encoding knowledge in a logical format. Large-scale applications such as Google's knowledge graph[1] illustrate the usefulness of such a structured representation for encoding entities, classes, and their respective relations. However, ontologies as used in current technical systems suffer from the symbol grounding problem [1, 2]: The symbols they contain are not directly linked to the real world, but are usually defined based on other symbols. Research in knowledge graph embeddings [3], where entities are identified with points in a feature space, provide only a partial solution to this problem, since the dimensions of this feature space are usually not tied to perception and action.

Deep neural networks have become the predominant approach in many machine learning tasks, including the areas of computer vision [4] and natural language processing [5]. They are able to extract compact representations from raw perceptual input without the need for extensive manual feature engineering. However, their recent successes have been accompanied with an urge for more human-like, explainable AI [6], since their overall input-output mapping

CAOS 2021: 5th Workshop on Cognition And OntologieS, held at JOWO 2021: Episode VII The Bolzano Summer of Knowledge, September 11-18, 2021, Bolzano, Italy

lbechberger@uos.de (L. Bechberger)
https://lucas-bechberger.de (L. Bechberger)
0000-0002-1962-1777 (L. Bechberger)

[1] https://blog.google/products/search/introducing-knowledge-graph-things-not/.

is opaque and cannot be easily analyzed or interpreted by human experts.

Neural-symbolic integration [7] offers the possibility to combine the interpretability of symbolic systems with the learning capabilities of artificial neural networks. Also the area of cognitive AI [8], i.e., intelligent systems inspired by findings from cognitive psychology, can help to align artificial systems more closely with human cognition.

The cognitive framework of conceptual spaces [9] unifies aspects of both the neural-symbolic tradition and cognitive AI. It proposes a geometric representation of conceptual knowledge based on psychological similarity spaces and offers an intermediate level of representation between the connectionist and the symbolic approach. The individual dimensions spanning such a conceptual space correspond to cognitively meaningful features of the inputs (such as *hue*, *saturation*, and *brightness* for colors). Concepts (such as the color *blue*) can then be defined as convex regions in this space. Conceptual spaces thus provide an indirect way of grounding symbolic descriptions in perception. They have seen a wide variety of applications in artificial intelligence, linguistics, psychology, and philosophy [10, 11].

Logic tensor networks [12, 13] (LTNs) are a type of neural network which uses fuzzy membership functions in order to ground symbolic predicates in a given embedding space. When optimizing the parameters of these membership functions, LTNs can take into account both bottom-up information in the form of training examples and top-down constraints in the form of general logical rules. For instance, conceptual hierarchies from an ontology can be used to enforce a subsethood relation between the respective membership functions.

In this paper, we propose to use logic tensor networks to ground ontologies in conceptual spaces. Since conceptual spaces are based on meaningful dimensions, this aids the interpretability of the resulting embedding. Conceptual spaces can be grounded in psychological dissimilarity ratings [14, 15] or the features extracted by deep neural networks from raw perceptual inputs. Therefore, a successful grounding of a given ontology in a conceptual space indirectly solves the symbol grounding problem in a cognitively plausible way. Moreover, by explicitly considering relations between classes, logic tensor networks can help to leverage background knowledge in order to learn conceptual regions.

The remainder of this paper is structured as follows: In Section 2, we describe both conceptual spaces and logic tensor networks in more detail. In Section 3, we then argue why a combination of these two frameworks seems promising and discuss possible implementations of different membership functions. Section 4 then concludes this paper.

2. Background

In the following, we will first give a general overview of the conceptual spaces framework (Section 2.1), before introducing logic tensor networks (Section 2.2).

2.1. Conceptual Spaces

A conceptual space as proposed by Gärdenfors [9] is a similarity space spanned by a small number of interpretable, cognitively relevant quality dimensions (e.g., *temperature*, *time*, *hue*, *pitch*). One can measure the distance between two observations with respect to each of these dimensions and aggregate them into a global notion of semantic distance. Semantic similarity is then defined as an exponentially decaying function of distance, i.e., $Sim(x,y) = e^{-c \cdot d(x,y)}$ with a sensitivity parameter $c > 0$.

The overall conceptual space can be structured into so-called domains, which represent, for example, different perceptual modalities such as *color*, *shape*, *taste*, and *sound*. The *color* domain, for instance, can be represented by the three dimensions *hue*, *saturation*, and *brightness*, while the *sound* domain is spanned by the dimensions *pitch* and *loudness*. Based on psychological evidence [16, 17], distance within a domain is measured with the Euclidean metric, while the Manhattan metric is used to aggregate distances across domains.

Gärdenfors defines properties like *red*, *round*, and *sweet* as convex regions within a single domain (namely, *color*, *shape*, and *taste*, respectively). A property thus corresponds to a set of observations from a single perceptual modality. Concept hierarchies are an emergent property of this spatial representation: If the *sky blue* region is a subset of the *blue* region, this implicitly encodes that *sky blue* is a special shade of *blue*. Based on properties, Gärdenfors now defines full-fleshed concepts like *apple* or *dog* by using one convex region per domain, a set of salience weights (which represent the relevance of the given domain to the given concept), and information about cross-domain correlations. The *apple* concept may thus be represented by the regions *red*, *sweet*, and *round* in the domains of *color*, *taste*, and *shape*, respectively.

There are in principle three ways of obtaining the dimensions of a conceptual space [9, Sections 1.7, 1.9, and 6.5]: Firstly, if the domain of interest is well understood, one can manually define the dimensions of the similarity space.

A second approach is based on machine learning algorithms for dimensionality reduction. For instance, unsupervised artificial neural networks (ANNs) such as autoencoders or self-organizing maps can be used to find a compressed representation for a given set of input stimuli. This task is however solved by minimizing a mathematical error function which seems to be not satisfactory from a psychological point of view.

A third popular way of obtaining a conceptual similarity space is based on psychological dissimilarity ratings. These dissimilarity ratings are collected for a fixed set of stimuli in a psychological experiment. They are then converted into an n-dimensional geometric representation of the stimulus set by using a technique called "multidimensional scaling" (MDS), which ensures that geometric distances between pairs of stimuli reflect their psychological dissimilarity [15]. While the similarity spaces produced by MDS are grounded in psychological experiments, they do not readily generalize to unseen stimuli [18].

Recently, a hybrid approach has been proposed [19, 20, 21, 22], where MDS is used to initialize the similarity space and ANNs are then trained to generalize the mapping to novel inputs.

Gärdenfors argues that the convexity requirement relates conceptual spaces to the prototype theory of concepts [23], which assumes that concept membership is based on similarity to a prototype. This can explain why some members of a category are deemed to be more typical than others. Gärdenfors [9, Section 3.8] now argues that if concepts are represented by convex regions, a prototype can be obtained by computing the center of gravity for the conceptual region. Conversely, Gärdenfors [9, Section 3.9] shows that by assuming a prototype-based representation, one can easily generate convex regions. For instance, if *color* properties such as *red* and *orange* are represented by their prototypical points in color space (e.g., their corresponding focal colors), one can partition the overall space into convex regions by assigning each point in the space to its closest prototype. This way of partitioning a space is called a Voronoi tessellation and will be discussed in more detail in Section 3.2.

Since conceptual spaces can be interpreted as an intermediate layer of representation between the traditional symbolic and subsymbolic layers, they can also help to solve the symbol grounding problem: Individual observations, which correspond to high-dimensional activation vectors in the subsymbolic layer, are represented by points in the lower-dimensional conceptual space and can be mapped onto constants and variables from the symbolic layer. Predicates from the symbolic layer (such as *apple* and *red*) can be mapped onto concepts and properties in the conceptual layer. The symbols from the symbolic layer can therefore be indirectly grounded in subsymbolic perception through the conceptual layer.

Conceptual spaces in their original formulation focus mostly on concepts that can be defined based on perceptual properties. Relations between objects can be represented using product spaces, for instance by defining *longerThan* as a convex region in $\mathbb{R}^+ \times \mathbb{R}^+$ (where each dimension represents the length of one individual object) [9, Section 3.10.1]. Relational concepts like *robber* or *seat* can be represented based on their respective roles in events like *robbing* and *sitting* (which involve agent, patient, theme, action, and result) [24, Sections 6.7 and 8.4].

2.2. Logic Tensor Networks

Logic Tensor Networks (LTNs) [12, 13, 25, 26] provide a principled way of using neural computations to connect feature spaces with symbolic rules through fuzzy logic.[2]

Logic Tensor Networks integrate knowledge representation, learning, and reasoning using a differentiable fuzzy first-order logic language called “Real Logic”. Real Logic is a first-order language containing constant symbols (representing individual observations such as *Bob* or *Paris*), function symbols (representing mappings between observations, e.g., *homeTownOf*), predicate symbols (representing concepts and relations such as *lawyer* and *livesIn*), and variable symbols (representing lists of observations) [13]. All of these language constituents are typed with respect to a set of domains $\mathcal{D}$: We can require that *Bob* belongs to the domain of *people* and *Paris* to the domain of *cities*, while the function *homeTownOf* takes only *people* as input and returns *cities*. The individual parts of the language can now be combined into formulas such as $\mathit{lawyer}(\mathit{Bob})$, or $\forall x : (\mathit{lawyer}(x) \rightarrow \mathit{homeTownOf}(x) = \mathit{Paris})$. These formulas are constructed using logical connectives (such as $\rightarrow$) and quantifiers (such as $\forall$) and have a fuzzy degree of truth in the interval $[0, 1]$.

In order to relate the semantics of the logical language to actual data points, Real Logic makes use of a so-called *grounding* function G_θ, which maps terms (i.e., constants, variables, and results of function applications) onto points in a feature space, and both functions and predicates onto neural networks. The networks implementing predicates are required to return a value from the interval [0,1] and can thus be interpreted as defining a fuzzy membership function of the respective concept or relation in the given feature space. Relations such as

[2] See https://github.com/logictensornetworks/logictensornetworks for the implementation of this framework.

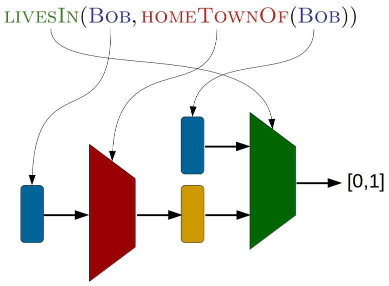

Figure 1: Grounding of a formula containing a function symbol (*homeTownOf*), a predicate symbol (*livesIn*), and a constant symbol (*Bob*).

livesIn are implemented as fuzzy regions in a product space of domains (in this case *people* and *cities*). Overall, the grounding associates any formula expressible in the language with a real number in [0,1], representing its degree of truth.

This is done as follows (see Figure 1): First, all terms (such as *Bob*) are grounded into vectors, and all function and predicate symbols (such as *homeTownOf* an *livesIn*) are grounded into their respective neural networks. The structure of the symbolic formula then determines which neural network is applied to which feature vector. If predicates such as $married(x, y)$ are applied to variables such as $x = (Bob, John)$ and $y = (Alice, Mary, Susan)$, the resulting grounding is a matrix containing the degree of truth for each possible combination of observations [13].

In order to ground logical connectives such as $\wedge$, $\vee$, $\neg$, and $\rightarrow$, the corresponding operators from fuzzy logic are used. Badreddine et al. [13] note that many standard operators from fuzzy logic are not well-suited for the context of neural networks, since they may cause vanishing or exploding gradients. They recommend using the product norm $T_{prod}(x, y) = x \cdot y$ for implementing the conjunction and its complement $S_{prod}(x, y) = x + y - x \cdot y$ for the disjunction. For the negation, $N(x) = 1 - x$ is used.

Since Real Logic is a first-order language, it also needs to provide a grounding for the universal and the existential quantifier. Mapping $\forall x$ to the minimum over all entries of the variable x may be a straightforward choice, but does not tolerate exceptions and may thus not be suitable for real-life applications, if one assumes a certain amount of noise both in the background knowledge and in the empirical data (e.g., incorrect labels) [26, 27]. Instead, the generalized mean $mean_p(x_1, \dots, x_d) = \left(\frac{1}{d}\sum_{i=1}^{d} x_i^p\right)^{\frac{1}{p}}$ is used, where the parameter p controls the "strictness" of the aggregator.[3] The current version of the framework [13] proposes to use different variants of the generalized mean for grounding both the universal and the existential quantifier. This of course breaks the duality between the existential and the universal quantifier, but seems to be necessary to enable robust gradient-based learning. One could envision to use both an exception-tolerant and a rigid classical version for both quantifiers. One would then however need to specify which version to apply in which contexts.

[3] Note that $mean_1$ corresponds to the standard arithmetic mean and $mean_{-1}$ to the harmonic mean. If $x = (x_1, \dots, x_d)$ is a difference vector, then $mean_p(x)$ is equivalent to a Minkowski metric.

Satisfiability specifies the degree to which a grounding G_θ satisfies a given set K of formulas by simply aggregating the truth values of all formulas $\phi \in K$ [12, 25, 26]. Donadello et al. [27] propose to use the generalized mean also for this purpose, since using a conjunction over the formulas can lead to undesired behavior in gradient-based optimization.

The knowledge represented in logic tensor networks consists of both the formulas ϕ in the logical language (corresponding to symbolic top-down information) and the grounding G_θ obtained from observations (corresponding to subsymbolic bottom-up information) [13]. One can encode different types of constraints into the system: For instance, one can explicitly fix the grounding for some of the symbols (e.g., mapping a given constant to a concrete feature vector). Also a parametric definition of predicates is possible by specifying the structure of the respective neural network, but leaving its exact parameter settings undetermined. Moreover, different kinds of formulas can be used to constrain the system: Factual propositions such as *lawyer*(*Bob*) encode facts about individual constants (which corresponds to providing labels for training examples), while generalized propositions such as $\forall x : (\textit{lawyer}(x) \rightarrow \textit{homeTownOf}(x) = \textit{Paris})$ allow to specify data-independent general background knowledge.

Learning in logic tensor networks takes place through gradient descent on the parameter values θ of the grounding G_θ in order to maximize the satisfiability of the overall set of formulas K [13]. In practice, maximizing satisfiability may need to be accompanied by a regularization term on the parameters θ to prevent overfitting [13, 26]. Once a grounding has been established, it can be used for answering concrete queries about the truth value of a given formula or about the embedding of a given term.

Since LTNs unify subsymbolic machine learning aspects with symbolic logical constraints, they can be applied to a variety of problems. Badreddine et al. [13] have given a principled overview of different tasks that can be solved with LTNs. These include classification, regression, clustering, semi-supervised pattern recognition, embedding learning, and knowledge base completion. LTNs have also been used to learn transitive predicates (such as *hyponymOf*) for simple ontologies based only on one-hop examples [28]. Moverover, Bianchi et al. [29] have recently illustrated the capability of LTNs to connect pre-trained entity embeddings with a subset of the DBpedia ontology [30]. Other practical applications include semantic image interpretation [26, 27, 31], incorporating fairness constraints into deep neural networks [32], and supplementing reinforcement learning algorithms with semantic knowledge [33].

3. Towards a Fruitful Combination

Logic tensor networks offer the possibility to combine bottom-up information in the form of training examples with top-down information in the form of general rules. They thus make an ideal candidate for closing the gap between the conceptual and the symbolic layer.

In Section 3.1, we show how logic tensor networks can reflect the general properties of the conceptual spaces framework. Afterwards, we investigate different membership functions, using a distinction into partitional (Section 3.2) and nonpartitional (Section 3.3) approaches.

3.1. General Considerations

The knowledge approach to concepts from psychology [34, Chapter 6] emphasizes the crucial role of world knowledge in the learning and application of concepts and can thus be linked to both formal ontologies and the influence of logical rules on the learning process in LTNs. Moreover, LTNs take into account information about points in a feature space, and the grounding of predicates usually gives rise to a membership function with one or more receptive fields. LTNs can therefore also be related to the prototype [23] and exemplar theories [35] of concepts.

These observations and the combination of bottom-up and top-down information make LTNs quite interesting from the perspective of conceptual spaces: If we use a conceptual space as a feature space, then the LTN can implement a two-way interaction between the conceptual and the symbolic layer. Just as with the conceptual spaces framework, observations can be represented by points and concepts can be represented as regions in the feature space. Moreover, LTNs are able to encode the domain structure of a conceptual space and they use a similar way of encoding simple relations as regions in a product space. Finally, the usage of fuzzy sets and fuzzy logic allows us to represent vague conceptual boundaries.

How exactly can we apply LTNs to conceptual spaces? Both properties and concepts can be represented by predicates with a convex membership function. While properties refer only to a single domain, concepts are defined on a concatenation of domains. We can require that the predicate *red*(x) is defined on the three-dimensional *color* domain, while the predicate *apple*$(x = (x_c, x_t, x_s))$ involves the domains of *color*, *taste*, and *shape*. Individual observations can then be represented by one point per domain. Function symbols could potentially be used to represent actions and changes: Applying a function symbol like *lift* could for instance translate into a simple vector addition in the *location* domain that increases the altitude of the given object. Finally, basic relations are defined by considering regions in product spaces of multiple domains. In order to represent more complex relational knowledge, one could try to use the event structure proposed by Gärdenfors [24, Chapter 9].

An advantage of using logic tensor networks for grounding ontologies in conceptual spaces is their large variety of inference and learning methods. In addition to learning conceptual regions from observations, they can for instance also generate an embedding of an unobserved object based on a symbolic description. For example, "object x is a *red apple*" can be translated into a point in the conceptual space by maximizing the satisfiability of *apple*$(x) \wedge$ *red*(x). This spatial representation of object x can then in turn be used to make further inferences, for instance about the *taste* domain (e.g., by evaluating *sweet*(x)). Thus, the geometric embedding can give rise to common-sense inferences not easily realizable within the symbolic layer.

Moreover, LTNs do not only provide an embedding of entities and classes, but they are also able to enforce the validity of general rules, which may reduce the required number of training examples. This makes them especially attractive for bridging the conceptual and the symbolic layer, since they can harness the whole expressivity of formal ontologies in order to guide the machine learning process. This also related to embodied and enactivist approaches to cognition [36], which assume that top-down information strongly influences bottom-up perception and conceptualization of the environment. One may furthermore speculate that the enforcement of general logical rules can help to prevent catastrophic interference, where continued learning causes a neural network to forget previously learned knowledge [37].

While LTNs have already been used in the context of ontologies [28, 29], their underlying Real Logic is not intended as a language for writing domain ontologies. Moreover, to the best of our knowledge, its formal properties (e.g., expressivity, decidability, or complexity) have not been thoroughly analyzed, yet. For our current purposes, Real Logic is merely used as a translation tool for encoding relevant domain knowledge from a given ontology as constraints for a machine learning process and for extracting structured knowledge (which may then be added to the ontology) from a trained machine learning system. The combination of conceptual spaces and LTNs is thus in principle applicable to any symbolic language.

Finally, we would like to mention the recent work by Singh et al. [38], who combine the computational power of deep ANNs with a psychological model of categorization. Their end-to-end model learns both a similarity space and a prototype-based categorization model at once. One could envision a similar application of LTNs: The input domain contains raw images, which are then mapped by a function symbol (implemented as deep neural network) into a low-dimensional conceptual space. In this conceptual space, one can then define membership functions for the different concepts under consideration. The whole system could then be trained based on labeled examples, but also using additional background knowledge based on human similarity ratings and general rules from the symbolic layer. In the terms of cognitive psychology, this would result in a combination of prototype theory (represented by convex membership functions) with the knowledge view on concepts (represented by the presence of constraints from background knowledge), spanning all three layers of representation.

When viewed from the perspective of cognitive science, logic tensor networks can of course not be labeled as a cognitively plausible learning mechanism: They rely on batch-processing large amounts of (typically labeled) data with gradient descent. One can of course argue that LTNs are not used to model the human concept acquisition process itself, but rather to take a shortcut to the resulting concept inventory. However, it would certainly also be interesting to extend LTNs such that they can work in an incremental way. A potential example application in this context are language games [39], where a population of agents needs to negotiate a common conceptualization of the world, receiving only indirect feedback through the success or failure of their interactions.

In the following, we will take a look at different membership functions from the conceptual spaces literature and discuss their applicability in logic tensor networks. For illustration purposes, we will consider a one-dimensional conceptual space with two concepts C_1 and C_2 as well as three data points x_1, x_2, x_3, which are supposed to belong to C_1, but not to C_2. Since the parameters of the membership functions are optimized through gradient descent, we will especially focus on their derivatives.

3.2. Partitional Membership Functions

Let us first consider membership functions which partition the underlying conceptual space, i.e., which aim to assign each point to exactly one concept. We start with Gärdenfors' approach of identifying concepts with a prototypical point and creating a Voronoi tessellation of the space

[9, Section 3.9]: One starts from a set of prototypical points $p_1, \dots, p_n$ for the n concepts under consideration. Each point x in the conceptual space is then assigned to its closest prototype p_i based on the distances $d(x, p_i)$. As Gärdenfors [9, Section 3.9] argues, a Voronoi tessellation based on the Euclidean metric partitions the overall space into convex regions.

Since the Voronoi tessellation gives us a partitioning of the overall space, the membership function of each concept C_i is constant almost everywhere and undefined on the border line to a neighboring conceptual region (see Figure 2a). Therefore, the derivative of this membership function with respect to any variable is either zero or undefined, which is highly problematic for gradient descent. For example, consider the point x_3, which is currently misclassified as belonging to C_2 instead of C_1. In gradient-based optimization, the prototypes p_1 and p_2 are updated by computing the derivative $\frac{d\mu_i(x_3)}{dp_i}$ and then slightly increasing or decreasing the value of p_i, depending on the sign of the derivative and whether we want to increase or decrease $\mu_i(x_3)$. In the case of Figure 2a, we however note that both derivatives are zero – small changes to p_1 and p_2 do not result in any changes to $\mu_i(x_3)$. Thus, gradient descent is incapable of making any update to the prototypes.

In order to make gradient-based learning possible, we need a soft version of the Voronoi approach. We can express the classification decision of the Voronoi tessellation as follows (where $c > 0$ is a sensitivity parameter):

$$i = argmin_j \; d(x, p_j) = argmax_j(-c \cdot d(x, p_j))$$

Instead of the crisp $argmax$ function (which results in flat membership values), we can now apply the so called *softmax* function, which is commonly used in neural networks to provide an output probability distribution over a set of mutually exclusive classes:

$$softmax(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Here, $softmax(z)_i$ gives the probability for class i, given a vector of raw confidence values z. If we combine this with the Voronoi tessellation approach, we obtain a soft Voronoi tessellation with the following membership function:

$$\mu_i(x) = softmax(-c \cdot d(x, p))_i = \frac{e^{-c \cdot d(x, p_i)}}{\sum_j e^{-c \cdot d(x, p_j)}}$$

As we can see, the numerator reflects the semantic similarity of x and p_i, while the denominator is the sum over all similarities to all prototypes. The resulting membership function can thus also be interpreted as a normalized version of semantic similarity. Figure 2b illustrates this membership function: We now have a continuous transition from high membership values to low membership values. Moreover, the derivative of this membership function is defined on the whole conceptual space and nonzero in all cases - even points such as x_1 have a very small, but nonzero derivative.

We should highlight at this point that we assume that the same sensitivity parameter c is used for all concepts. If we allow different values $c_1 \neq c_2$, we can control the size of the respective conceptual regions (smaller values of c leading to larger regions). However, these different

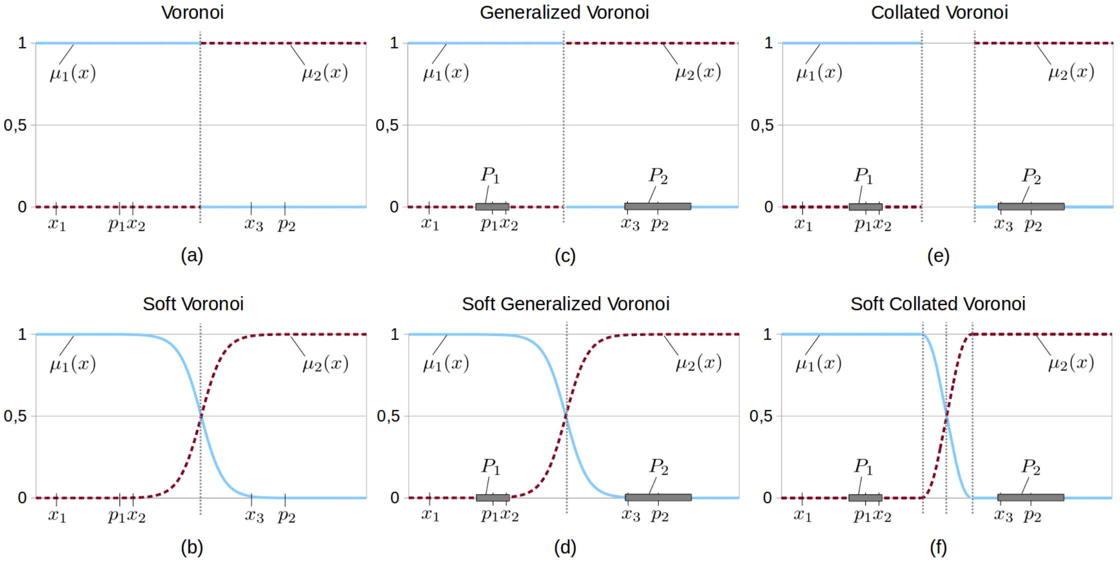

Figure 2: Partitional membership functions in a one-dimensional conceptual space.

values may cause some unintended effects. For instance, Figure 3a illustrates the case where $c_1 \gg c_2$, which causes the membership function $\mu_2(x)$ to be no longer convex.

Generalized Voronoi tessellations [9, Section 4.9] allow to encode differently sized conceptual regions by considering prototypical regions P_i instead of prototypical points p_i. These prototypical regions are usually represented as disks with a central point p_i and a radius r_i. Based on these prototypical regions, one can now generate a generalized Voronoi tessellation by assigning each point x in the conceptual space to the concept whose prototypical region is closest. In the case of disks, this corresponds to finding the concept C_i for which $d(x, P_i) = \max(0, d(x, p_i) - r_i)$ is smallest. Concepts with larger prototypical regions (as reflected through a larger value of r_i) thus result in larger conceptual regions in the generalized Voronoi tessellation (see Figure 2c). Again, by using the $softmax$ instead of the $argmax$ function, this can be generalized to a soft notion of concept membership (see Figure 2d).

Also Douven et al. [40] consider prototypical regions instead of prototypical points. However, they create all possible Voronoi diagrams by picking a single point $p_i \in P_i$ for all prototypical regions P_i. These individual Voronoi tessellations are then aggregated into a so called “collated Voronoi diagram”: A point x is assigned to concept C_i if and only if it has been assigned to C_i in *all* individual Voronoi diagrams. Douven et al. identify borderline cases as points x that belong to different conceptual regions for different Voronoi diagrams. These borderline points are not assigned to any concept and represent vagueness in concept boundaries. In Figure 2e, we again note that the derivative is zero within the conceptual regions and undefined in the border area.

Decock and Douven [41] extend the work of Douven et al. [40] by providing a degree of

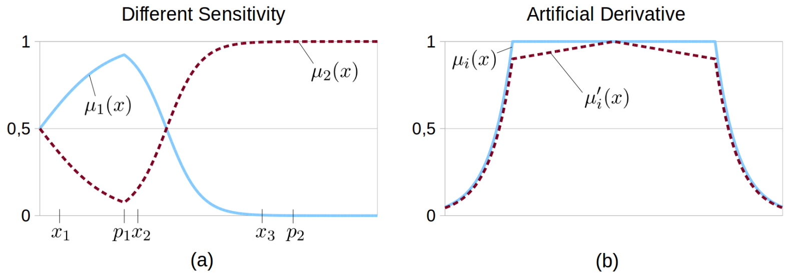

Figure 3: (a) Unintended results for the soft Voronoi membership and different sensitivity parameters $c_1 \gg c_2$. (b) Workaround for zero gradient inside prototypical regions.

membership for borderline cases. They define the membership of a point x to a concept C_i as the fraction of individual Voronoi diagrams for which x belongs to the conceptual region of C_i. Decock and Douven note that if the prototypical regions P_i have an infinite number of points, then the membership function is s-shaped (cf. Figure 2f). However, we can observe that the membership function is flat for large parts of the conceptual space, namely, for all non-borderline points. This is again highly problematic for gradient descent.

3.3. Nonpartitional Membership Functions

The usage of Voronoi tessellations for conceptual spaces has not been without challenge in the literature. For instance, Lewis and Lawry [42] argue that partitioning the conceptual space may be adequate for individual domains such as *color*, but that it is not suitable for a combination of multiple domains. It seems implausible that every single point in a high-dimensional space has to be assigned to exactly one category: On the one hand, some regions in the overall conceptual space may not be covered by any existing concept. Points in such regions should be recognized as outliers or members of a novel, previously unknown category. On the other hand, conceptual regions may also overlap, for instance in order to represent conceptual hierarchies.

Lewis and Lawry [42] have also made a general proposal for nonpartitional membership functions: A point x in the conceptual space is said to belong to concept C_i if its distance to the prototypical region P_i is not greater than a threshold distance ϵ_i. Lewis and Lawry assume that the threshold ϵ_i is not known, but that a probability distribution δ_i over its possible values is available. The degree of membership of a point x to a concept C_i is then given by the probability of $d(x, P_i)$ being smaller than ϵ_i:

$$\mu_i(x) = \mathbb{P}_\delta(d(x, P_i) \leq \epsilon_i) = \int_{d(x,P_i)}^{\infty} \delta_i(\epsilon_i) d\epsilon_i$$

Lewis and Lawry are in general open to different forms for the probability distribution δ_i. If we use $\delta_i(\epsilon_i) = c_i \cdot e^{-c_i \cdot \epsilon}$, then concept membership reflects similarity to the prototypical region:

$$\mu_i(x) = \int_{d(x,P_i)}^{\infty} c_i \cdot e^{-c_i \cdot \epsilon_i} d\epsilon_i = \left[-e^{-c_i \cdot \epsilon_i}\right]_{\epsilon_i = d(x,P_i)}^{\epsilon_i \to \infty} = 0 - \left(-e^{-c_i \cdot d(x,P_i)}\right) = e^{-c_i \cdot d(x,P_i)}$$

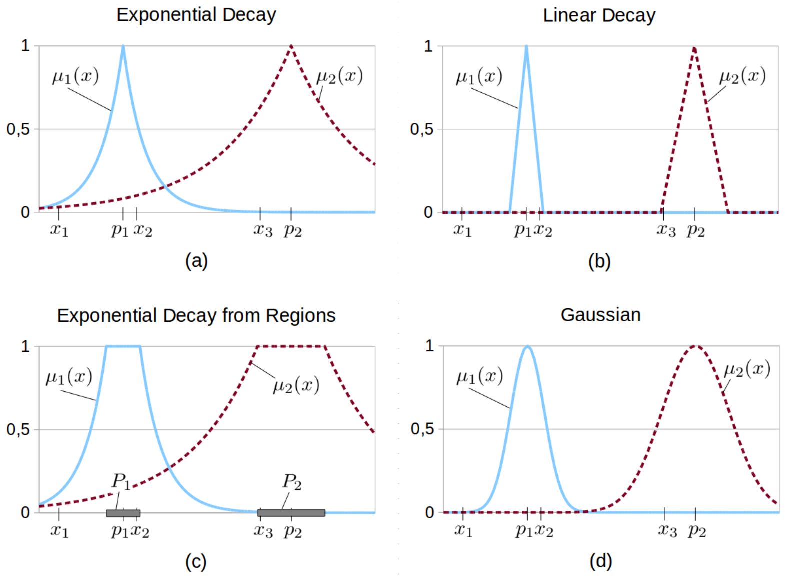

Figure 4: Nonpartitional membership functions in a one-dimensional conceptual space.

The shape of the resulting similarity function for $P_i = \{p_i\}$ (i.e., prototypical points) is illustrated in Figure 4a. As we can see, all points in the similarity space receive a non-zero membership value. Moreover, the derivative of the membership function is defined for all points except for the prototypes p_1 and p_2. In practical applications, this theoretical shortcoming can be overcome by defining the derivative in this point to equal zero. Furthermore, we are able to control the size of the conceptual regions by choosing different sensitivity parameters $c_1 \neq c_2$.

However, we can also note that the derivative of the membership function is proportional to the membership value itself: The largest derivatives are observed for the points with the highest membership in the concept. Since gradient descent algorithms typically take into account not only the direction, but also the magnitude of the gradient, this can lead to undesired effects. Consider for instance the point x_2 in Figure 4a, which has a fairly high membership to C_1. The derivative $\frac{d\mu_1(x_2)}{dp_1}$ is quite large and will thus cause the gradient descent algorithm to increase p_1 considerably. In the resulting configuration, $\mu_1(x_2)$ may however be smaller than before, since p_1 may have moved considerably past x_2. This seems to be a major shortcoming of this similarity-based approach to concept membership.

The examples by Lewis and Lawry [42] often make use of uniform distributions $\delta_i(\epsilon_i) = Uniform(0, r_i)$. As we can see in Figure 4b, the membership curve has in this case a triangular shape and its derivative is therefore constant for all points with a partial membership. However, both concept membership and its derivative are zero for most parts of the similarity space.

These considerations can of course also be generalized to prototypical regions P_i, which subsumes our own formalization of the conceptual spaces framework [43, 44, 45]: There exists a well-defined region with full membership, which in our case is based on the union of axis-aligned cuboids. Membership is then defined as similarity to this prototypical region.

In Figure 4c, we can see two problems with this approach: On the one hand, we again have the problem of large derivatives for large membership values as already discussed for Figure 4a. On the other hand, the membership function is constant for all points in the prototypical region, hence, the derivative is zero. If an observation such as x_3 is confidently misclassified as belonging to C_2, then gradient descent is not able to move P_2 away from x_3.

The problem of a zero derivative could be circumvented as follows: We define a new membership function $\mu'_i(x) := (1 - \epsilon) \cdot \mu_i(x)$ for some small $\epsilon > 0$. Furthermore, we identify the central point $p_i \in P_i$. The membership value for $x \in P_i$ is then increased based on its distance to p_i, such that $\mu'_i(p_i) = 1$ and that $\mu'_i(x) = 1 - \epsilon$ for points on the border of P_i. This provides a small slope for the membership function inside the prototypical region and thus a nonzero derivative (cf. Figure 3b). However, it remains to be seen whether such a workaround is useful in practice.

Despite these shortcomings, there are however reasonably strong arguments for using a membership function like the one proposed in our formalization: Firstly, by using a union of axis-aligned cuboids, our formalization is able to represent correlations between domains. This is an important aspect of human conceptualization [46, 47] which is not captured by any of the aforementioned approaches. Secondly, one can apply a variety of operations defined in the context of our formalization in order to reason on the learned concepts. For instance, relations such as conceptual similarity and conceptual betweenness are not defined in LTNs, but they become immediately available with the use of our proposed formalization of concepts. Thirdly, logical formulas in LTNs always have to be evaluated on a set of data points which requires that one keeps all examples in memory. Our formalization on the other hand provides closed formulas for computing the validity of such logical formulas – the original data points are not needed any more and the computation can potentially be faster. However, the operations defined in our formalization are based on the minimum norm, while LTNs are commonly used with the product norm. Therefore, the numeric results of the computations might differ.

Motivated by the problem of large gradients for large membership values, we also consider multivariate Gaussian functions, whose membership value can be defined as follows with a symmetric, positive semi-definite matrix Σ:

$$\mu_i(x) = e^{-\frac{1}{2}(x-p_i)^T \Sigma^{-1} (x-p_i)}$$

Figure 4d illustrates the usage of such Gaussian functions in our one-dimensional similarity space.[4] As one can see, this type of membership function does not suffer from the gradient size problem as identified in Figure 4a: The derivative is small both for points with a very low and for points with a very high membership. It is largest for points with an intermediate level of membership, i.e., points that are currently treated as borderline members. Another advantage of multivariate Gaussian functions is that they are able to encode correlations between dimensions as well as different distribution widths through their covariance matrix Σ.

[4] We can model this with $\delta_i(\epsilon_i) = \frac{\epsilon_i}{\sigma_i^2} \cdot e^{-\frac{\epsilon_i^2}{2\sigma_i^2}}$ in the one-dimensional case using the approach by Lewis and Lawry [42]

However, the usage of Gaussians in the context of conceptual spaces is somewhat unsatisfactory from a theoretical standpoint. The notion of similarity is not based on the Euclidean distance $d_E(x, p_i) = \sqrt{\sum_d (x_d - p_{id})^2}$, but on the squared Mahalanobis distance $d_M(x, p_i) = \sqrt{(x - p_i)^T \Sigma^{-1} (x - p_i)}$. Applying the Mahalanobis distance corresponds to transforming the similarity space with the covariance matrix, and then computing the Euclidean metric in the transformed space. This implicit transformation of the similarity space would in our opinion cause a major modification of the original framework. Nevertheless, the simplicity and computational attractiveness of multivariate Gaussians make them an interesting candidate for experimental investigations, such that one should not hastily dismiss them.

4. Conclusions

In this paper, we have introduced both conceptual spaces and logic tensor networks. We have argued that a combination of the two frameworks is a promising direction of research: Conceptual spaces can provide a grounding for the feature spaces considered in logic tensor networks and allow us to use relatively simple membership functions for representing predicates. Logic tensor networks on the other hand can help us to bridge the gap between the conceptual and the symbolic layer by learning concepts not only based on labeled examples, but also based on general logical top-down constraints. Moreover, the resulting system can provide a cognitive grounding for formal ontologies: Individual concepts from the ontology can be grounded in regions of a conceptual space, whose dimensions are grounded in psychological data and/or perceptual sensor information. Moreover, this grounding can take into account the most important part of ontologies, namely the relations between concepts. Since the envisioned system unifies both bottom-up and top-down processes, the information from the conceptual layer can furthermore give rise to additional rules for the symbolic ontology.

Moreover, we have also discussed several possible membership functions for concepts in conceptual spaces and their applicability to gradient-based optimization methods. If we consider partitional approaches, a soft version of generalized Voronoi tessellations seems to be most promising: It is capable of representing conceptual regions of different sizes and comes with a derivative that is guaranteed to be non-zero everywhere. If we are however interested in nonpartitional membership functions, multivariate Gaussians seem to be preferable from a computational point of view: They are able to explicitly encode correlations between dimensions, they can take into account concepts of varying size, and they provide a meaningful non-zero gradient everywhere. Nevertheless, also the membership function of our own formalization of the conceptual spaces framework should be explored, since it provides us with a large number of operations for downstream reasoning processes.

Our proposal has so far been only a theoretical one. In order to evaluate its actual merit, practical experiments need to be conducted. Ideally, these experiments should consider all membership functions discussed in this paper in order to confirm or refute our theoretical analyses. There are several data sets that can serve as test beds for a first study, including the conceptual spaces extracted by Banaee et al. [48] and Derrac and Schockaert [49], as well as the robotics data set by Spranger et al. [50]. Since the strength of LTNs stems from their ability to incorporate top-down rules to compensate for scarce training data, especially the movie spaces from Derrac and Schockaert [49] are relevant: Each movie is annotated with its genres, a set of plot keywords, at its age restriction. Using techniques such as the apriori algorithm [51], one can extract rules from the co-occurrence statistics of the labels and then simulate few-shot learning [52] by showing only a small part of the available examples, but providing the general rules as additional constraints to the system. After such initial experiments, studies with actual ontologies are needed in order to ensure that all relevant pieces of ontological information (especially relations of varying complexity) can be adequately encoded by the proposed approach.

References

[1] S. Harnad, The Symbol Grounding Problem, Physica D: Nonlinear Phenomena 42 (1990) 335–346. doi:10.1016/0167-2789(90)90087-6.

[2] P. Gärdenfors, How to Make the Semantic Web More Semantic, in: Formal Ontology in Information Systems, 2004, pp. 19–36.

[3] O. Lütfü Özçep, M. Leemhuis, D. Wolter, Cone Semantics for Logics with Negation, in: C. Bessiere (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, International Joint Conferences on Artificial Intelligence Organization, 2020, pp. 1820–1826. doi:10.24963/ijcai.2020/252, main track.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners (2020). arXiv:2005.14165.

[6] G. Marcus, E. Davis, Rebooting AI: Building Artificial Intelligence We Can Trust, Pantheon, 2019.

[7] A. d. Garcez, T. R. Besold, L. De Raedt, P. Földiak, P. Hitzler, T. Icard, K.-U. Kühnberger, L. C. Lamb, R. Miikkulainen, D. L. Silver, Neural-Symbolicy Learning and Reasoning: Contributions and Challenges, in: AAAI 2015 Spring Symposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches, 2015.

[8] A. Lieto, Cognitive Design for Artificial Minds, Routledge, 2021.

[9] P. Gärdenfors, Conceptual Spaces: The Geometry of Thought, MIT Press, 2000.

[10] F. Zenker, P. Gärdenfors (Eds.), Applications of Conceptual Spaces, Springer Science + Business Media, 2015. doi:10.1007/978-3-319-15021-5.

[11] M. Kaipainen, F. Zenker, A. Hautamäki, P. Gärdenfors (Eds.), Conceptual Spaces: Elaborations and Applications, volume 405, Springer, 2019.

[12] L. Serafini, A. d'Avila Garcez, Logic Tensor Networks: Deep Learning and Logical Reasoning from Data and Knowledge (2016). arXiv:1606.04422.

[13] S. Badreddine, A. d'Avila Garcez, L. Serafini, M. Spranger, Logic Tensor Networks, 2021. arXiv:2012.13635.

[14] L. Bechberger, M. Scheibel, Analyzing Psychological Similarity Spaces for Shapes, in: M. Alam, T. Braun, B. Yun (Eds.), Ontologies and Concepts in Mind and Machine, Springer International Publishing, Cham, 2020, pp. 204–207.

[15] I. Borg, J. F. Groenen, Modern Multidimensional Scaling: Theory and Applications, Springer Series in Statistics, 2nd ed., Springer-Verlag New York, 2005.

[16] F. Attneave, Dimensions of Similarity, The American Journal of Psychology 63 (1950) 516–556. doi:10.2307/1418869.

[17] R. N. Shepard, Attention and the Metric Structure of the Stimulus Space, Journal of Mathematical Psychology 1 (1964) 54–87. doi:10.1016/0022-2496(64)90017-3.

[18] R. M. Battleday, J. C. Peterson, T. L. Griffiths, From Convolutional Neural Networks to Models of Higher-Level Cognition (and Back Again), Annals of the New York Academy of Sciences (2021). doi:10.1111/nyas.14593.

[19] L. Bechberger, K.-U. Kühnberger, Generalizing Psychological Similarity Spaces to Unseen Stimuli – Combining Multidimensional Scaling with Artificial Neural Networks, Springer International Publishing, Cham, 2021, pp. 11–36. doi:10.1007/978-3-030-69823-2_2.

[20] L. Bechberger, K.-U. Kühnberger, Mapping Line Drawings Into Shape Space - Combining Convolutional Neural Networks with Psychological Similarity Spaces, Machine Learning (under review).

[21] C. A. Sanders, R. M. Nosofsky, Using Deep-Learning Representations of Complex Natural Stimuli as Input to Psychological Models of Classification, in: Proceedings of the 2018 Conference of the Cognitive Science Society, Madison., 2018.

[22] C. A. Sanders, R. M. Nosofsky, Training Deep Networks to Construct a Psychological Feature Space for a Natural-Object Category Domain, Computational Brain & Behavior 3 (2020) 229–251.

[23] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, P. Boyes-Braem, Basic Objects in Natural Categories, Cognitive Psychology 8 (1976) 382–439. doi:10.1016/0010-0285(76)90013-x.

[24] P. Gärdenfors, The Geometry of Meaning: Semantics Based on Conceptual Spaces, MIT Press, 2014.

[25] L. Serafini, A. S. d'Avila Garcez, Learning and Reasoning with Logic Tensor Networks, Springer International Publishing, Cham, 2016, pp. 334–348. doi:10.1007/978-3-319-49130-1_25.

[26] L. Serafini, I. Donadello, A. d. Garcez, Learning and Reasoning in Logic Tensor Networks: Theory and Application to Semantic Image Interpretation, in: Proceedings of the Symposium on Applied Computing, SAC '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 125–130. doi:10.1145/3019612.3019642.

[27] I. Donadello, L. Serafini, Compensating Supervision Incompleteness with Prior Knowledge in Semantic Image Interpretation, in: 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1–8.

[28] F. Bianchi, P. Hitzler, On the Capabilities of Logic Tensor Networks for Deductive Reasoning, in: Proceedings of the AAAI Spring Symposium on Combining Machine Learning with Knowledge Engineering, AAAI-MAKE, 2019.

[29] F. Bianchi, M. Palmonari, P. Hitzler, L. Serafini, Complementing Logical Reasoning with Sub-symbolic Commonsense, in: P. Fodor, M. Montali, D. Calvanese, D. Roman (Eds.),

Rules and Reasoning, Springer International Publishing, Cham, 2019, pp. 161–170.

[30] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: A Nucleus for a Web of Open Data, in: K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, P. Cudré-Mauroux (Eds.), The Semantic Web, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 722–735.

[31] I. Donadello, L. Serafini, A. d'Avila Garcez, Logic Tensor Networks for Semantic Image Interpretation, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), 2017.

[32] B. Wagner, A. d'Avila Garcez, Neural-Symbolic Integration for Fairness in AI, in: A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021), 2021.

[33] S. Badreddine, M. Spranger, Injecting Prior Knowledge for Transfer Learning into Reinforcement Learning Algorithms using Logic Tensor Networks (2019). arXiv:1906.06576.

[34] G. Murphy, The Big Book of Concepts, MIT Press, 2002.

[35] D. L. Medin, M. M. Schaffer, Context Theory of Classification Learning, Psychological Review 85 (1978) 207.

[36] A. K. Engel, A. Maye, M. Kurthen, P. König, Where's the Action? The Pragmatic Turn in Cognitive Science, Trends in Cognitive Sciences 17 (2013) 202–209. doi:10.1016/j.tics.2013.03.006.

[37] M. McCloskey, N. J. Cohen, Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem, volume 24 of *Psychology of Learning and Motivation*, Academic Press, 1989, pp. 109–165. doi:10.1016/S0079-7421(08)60536-8.

[38] P. Singh, J. Peterson, R. Battleday, T. Griffiths, End-to-end Deep Prototype and Exemplar Models for Predicting Human Behavior, in: Proceedings for the 42nd Annual Meeting of the Cognitive Science Society, 2020.

[39] L. Steels, The Talking Heads Experiment: Origins of Words and Meanings, Language Science Press, 2015.

[40] I. Douven, L. Decock, R. Dietz, P. Égré, Vagueness: A Conceptual Spaces Approach, Journal of Philosophical Logic 42 (2011) 137–160. doi:10.1007/s10992-011-9216-0.

[41] L. Decock, I. Douven, What Is Graded Membership?, Noûs 48 (2014) 653–682. doi:10.1111/nous.12003.

[42] M. Lewis, J. Lawry, Hierarchical Conceptual Spaces for Concept Combination, Artificial Intelligence 237 (2016) 204–227. doi:10.1016/j.artint.2016.04.008.

[43] L. Bechberger, K.-U. Kühnberger, A Thorough Formalization of Conceptual Spaces, in: G. Kern-Isberner, J. Fürnkranz, M. Thimm (Eds.), KI 2017: Advances in Artificial Intelligence: 40th Annual German Conference on AI, Dortmund, Germany, September 25–29, 2017, Proceedings, Springer International Publishing, 2017, pp. 58–71. doi:10.1007/978-3-319-67190-1_5.

[44] L. Bechberger, K.-U. Kühnberger, Formal Ways for Measuring Relations between Concepts in Conceptual Spaces, Expert Systems 0 (2018) e12348. doi:10.1111/exsy.12348, e12348 EXSY-Apr-18-107.R1.

[45] L. Bechberger, K.-U. Kühnberger, Formalized Conceptual Spaces with a Geometric Representation of Correlations, Springer International Publishing, Cham, 2019, pp. 29–58.

doi:10.1007/978-3-030-12800-5_3.

[46] D. Billman, E. Heit, Observational Learning from Internal Feedback: A Simulation of an Adaptive Learning Method, Cognitive Science 12 (1988) 587 – 625. doi:10.1016/0364-0213(88)90014-6.

[47] D. L. Medin, E. J. Shoben, Context and Structure in Conceptual Combination, Cognitive Psychology 20 (1988) 158–190. doi:10.1016/0010-0285(88)90018-7.

[48] H. Banaee, E. Schaffernicht, A. Loutfi, Data-driven Conceptual Spaces: Creating Semantic Representations for Linguistic Descriptions of Numerical Data, Journal of Artificial Intelligence Research 63 (2018) 691–742.

[49] J. Derrac, S. Schockaert, Inducing Semantic Relations from Conceptual Spaces: A Data-Driven Approach to Plausible Reasoning, Artificial Intelligence 228 (2015) 66–94. doi:10.1016/j.artint.2015.07.002.

[50] M. Spranger, M. Loetzsch, L. Steels, Language Grounding in Robots, Springer US, Boston, MA, 2012, pp. 89–110. doi:10.1007/978-1-4614-3064-3_5.

[51] R. Agrawal, R. Srikant, Fast Algorithms for Mining Association Rules in Large Databases, in: VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile, 1994, pp. 487–499. URL: http://www.vldb.org/conf/1994/P487.PDF.

[52] Y. Wang, Q. Yao, J. T. Kwok, L. M. Ni, Generalizing from a Few Examples: A Survey on Few-Shot Learning, ACM Comput. Surv. 53 (2020). doi:10.1145/3386252.