Learning Conceptual Space Representations of Interrelated Concepts

Zied Bouraoui
CRIL - CNRS & Univ Artois, France
bouraoui@cril.univ-artois.fr

Steven Schockaert
Cardiff University, UK
SchockaertS1@Cardiff.ac.uk

Abstract

Several recently proposed methods aim to learn conceptual space representations from large text collections. These learned representations associate each *object* from a given domain of interest with a point in a high-dimensional Euclidean space, but they do not model the *concepts* from this domain, and can thus not directly be used for categorization and related cognitive tasks. A natural solution is to represent concepts as Gaussians, learned from the representations of their instances, but this can only be reliably done if sufficiently many instances are given, which is often not the case. In this paper, we introduce a Bayesian model which addresses this problem by constructing informative priors from background knowledge about how the concepts of interest are interrelated with each other. We show that this leads to substantially better predictions in a knowledge base completion task.

1 Introduction

Conceptual spaces are geometric representations of knowledge, in which the objects from some domain of interest are represented as points in a metric space, and concepts are modelled as (possibly vague) convex regions [Gärdenfors, 2000]. The theory of conceptual spaces has been extensively used in philosophy, e.g. to study metaphors and vagueness [Douven *et al.*, 2013], and in psychology, e.g. to study perception in domains such as color [Jäger, 2009] and music [Forth *et al.*, 2010], and in other areas such as Robotics [Chella *et al.*, 2003] and Machine Vision [Chella *et al.*, 1997]. However, the lack of automated methods for learning conceptual spaces from data has held back its adoption in the field of artificial intelligence. While a number of such methods have recently been proposed [Jameel *et al.*, 2017], an important remaining problem is that these methods typically do not explicitly model concepts, i.e. they only learn the representations of the objects, while it is the concept representations that are mostly needed in applications such as knowledge base completion. The problem we study in this paper is to induce these missing concept representations from these object representations.

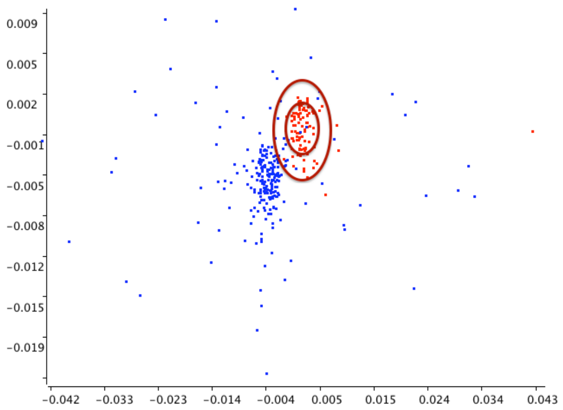

Figure 1: Two dimensions from a vector space embedding of places. Places known by SUMO or Wikidata to be train stations are shown in red.

To illustrate the considered problem, Figure 1 shows two dimensions from a higher-dimensional conceptual space of places. The red dots correspond to places which are asserted to be train stations in the SUMO ontology or on WikiData. From this red point cloud, we can learn a soft boundary for the concept *train station*, which is illustrated by the ellipsoidal contours in the figure. We can then plausible conclude that points which are within these boundaries are likely to be train stations. In accordance with prototype theory, this model essentially assumes that the likelihood that an object is considered to be an instance of a train station depends on its distance to a prototype. Note that by considering ellipsoidal rather than spherical contours, we can take into consideration that different dimensions may have different levels of importance in any given context. In principle, there are several strategies that could be used to find suitable ellipsoids for a given concept, e.g. we could train a support vector machine with a quadratic kernel or we could fit a Gaussian distribution. The key problem with these methods, however, is that conceptual spaces usually have hundreds of dimensions, whereas only a few instances of each concept may be available, making learned concept representations potentially unreliable.

While this cannot be avoided in general, in many applications we have some background knowledge about how the considered concepts are interrelated. In this paper, we propose a Bayesian model which exploits such background knowledge while jointly learning a representations for all concepts. In particular, we assume that concepts can be modelled using Gaussians, and we use the available background knowledge to construct informative priors on the parameters of these Gaussians. We will consider two kinds of such background knowledge. First, we will consider logical dependencies between the concepts, which can be encoded using description logic (DL). For instance, SUMO encodes the knowledge that each instance of *TrainStation* is an instance of *terminalBuilding* or of *transitBuilding*, which we should intuitively be able to exploit when learning representations for these concepts. Second, we will use the fact that many concepts themselves also correspond to objects in some conceptual space. For example, while train station is a concept in a conceptual space of *places*, it is an object in a conceptual space of *place types*. As we will see below, these representations of concepts-as-points can be used to implement a form of analogical reasoning. We experimentally demonstrate the effectiveness of our proposed model in a knowledge base completion task. Specifically, we consider the problem of identifying missing instances of the concepts from a given ontology. We show that our method is able to find such instances, even for concepts for which initially only very instances were known, or even none at all.

2 Related Work

Learning Conceptual Spaces One common strategy to obtain conceptual spaces is to learn them from human similarity judgments using multidimensional scaling. Clearly, however, such a strategy is only feasible in small domains. To enable larger-scale application, a number of approaches have recently been proposed for learning Euclidean[1] conceptual space representations in a purely data-driven way. In our experiments, we will in particular rely on the MEmbER model from [Jameel *et al.*, 2017], which learns vector space representations that can be seen as approximate conceptual spaces. For instance, in contrast to most other vector space models, objects of the same semantic type are grouped in lower-dimensional subspaces within which dimensions corresponding to salient features (i.e. quality dimensions) can be found. In other words, this approach can be seen as learning a set of conceptual spaces, one for each considered semantic type, which are themselves embedded in a higher-dimensional vector space. Most importantly for this paper, the objective of the MEmbER model directly imposes the requirement that all entities which are strongly related to a given word (i.e. whose textual descriptions contain sufficiently many occurrences of the word) should be located within some ellipsoidal region of the space. It thus aims to learn a representation in which concepts can be faithfully modelled as densities with ellipsoidal contours, such as Gaussians.

Throughout this paper, we will assume that natural concepts can be modelled as (scaled) Gaussians. This corresponds to a common implementation of prototype theory [Rosch, 1973], in which the prototype of a concept is represented as a point and the membership degree of an object is proportional to its exponential squared distance to the prototype. Note that, in general, prototypes do not have to be modelled as points, e.g. a more general approach is to model prototypes as regions [Douven *et al.*, 2013]. However, the restriction to prototype points is a useful simplifying assumption if we want to learn reasonable concept representations from small numbers of instances. Similarly, while in principle it would be useful to model concepts as Gaussian mixture models, a strategy which was proposed in [Rosseel, 2002] to generalize both prototype and exemplar models, this would only be feasible if a large number of instances of each concept were known.

Knowledge Graph Completion The main application task considered in this paper is knowledge base completion, i.e. identifying plausible facts which are missing from a given knowledge base. Broadly speaking, three types of highly complementary methods have been considered for this task. First, some methods focus on identifying and exploiting statistical regularities in the given knowledge base, e.g. by learning predictive latent clusters of predicates [Kok and Domingos, 2007; Rocktäschel and Riedel, 2016; Sourek *et al.*, 2016] or by embedding predicates and entities in a low-dimensional vector space [Bordes *et al.*, 2013]. The second class consists of approaches which extract facts that are asserted in a text corpus. For example, starting with [Hearst, 1992], a large number of methods for learning taxonomies from text have been proposed [Kozareva and Hovy, 2010; Alfarone and Davis, 2015]. Several authors have proposed methods that use a given incomplete knowledge base as a form of distant supervision, to learn how to extract specific types of fine-grained semantic relationships from a text corpus [Mintz *et al.*, 2009; Riedel *et al.*, 2010]. Thirdly, some methods, including ours, aim to explicitly represent concepts in some underlying feature space. For example, [Neelakantan and Chang, 2015] represents each Freebase entity using a combination of features derived from Freebase itself and from Wikipedia, and then uses a max-margin model to identify missing types. In [Bouraoui *et al.*, 2017], description logic concepts were modelled as Gaussians in a vector space embedding. Crucially, these existing works consider each concept in isolation, which requires that large numbers of instances are known, and this is often not the case.

Few Shot Learning Considerable attention has also been paid to the problem of learning categories for which no, or only few training examples are available, especially within the area of image recognition. For example, in one common setting, each category is defined w.r.t. a set of features, and the assumption is that we have training examples for some of the categories, but not for all of them. Broadly speaking, the aim is then to learn a model of the individual features, rather than the categories, which then makes it possible to make pre-

[1] While the use of Euclidean spaces is quite natural, another common choice is to use a two-level representation, where a concept at the top level is a weighted set of properties, each of which corresponds to a convex region in a different Euclidean space. Such representations open up interesting possibilities, but there are currently no established methods for learning them in an automated way.

dictions about previously unseen categories [Palatucci *et al.*, 2009; Romera-Paredes and Torr, 2015]. Other approaches instead exploit the representation of the category names in a word embedding [Socher *et al.*, 2013]. We will similarly exploit vector space representations of concept names.

3 Background

We will rely on description logic encoding of how different concepts are related, and we will use a Bayesian approach for estimating Gaussians modelling these concepts. In this section, we briefly recall the required technical background on these two topics. For a more comprehensive discussion, we refer to [Baader *et al.*, 2003] and [Murphy, 2007] respectively.

Description Logics Description logics are a family of logics which are aimed at formalizing ontological knowledge about the concepts from a given domain of interest. The basic ingredients are individuals, concepts and roles, which at the semantic level respectively correspond to objects, sets of objects, and binary relations between objects. A knowledge base in this context consists of two parts: a TBox, which encodes how the different concepts and roles from the ontology are related, and an ABox, which enumerates some of the instances of the considered concepts and roles. The TBox is encoded as a set of concept inclusion axioms of the form $C \sqsubseteq D$, which intuitively encodes that every instance of the concept C is also an instance of the concept D. Here, C and D are so-called concept expressions. These concept expressions are either atomic concepts or complex concepts. In this paper we will consider complex concepts that are constructed in the following ways:

- If C and D are concept expressions, then $C \sqcap D$ and $C \sqcup D$ are also concept expressions, modelling the intersection and union of the concepts C and D respectively.
- If C is a concept expression and R is a role, then $\exists R.C$ and $\forall R.C$ are also concept expressions. Intuitively, an individual belongs to the concept $\exists R.C$ if it is related (w.r.t. the role R) to some instance from C; an individual belongs to $\forall R.C$ if it can only be related (w.r.t. R) to instances of C.

From a given knowledge base, we can typically infer further concept inclusion axioms and ABox assertions, although the complexity of such reasoning tasks crucially depends on the specific description logic variant that is considered (e.g. which types of constructs are allowed and what restrictions are imposed on concept inclusion axioms). Note that the methods we propose in this paper are independent of any particular description logic variant; we will simply assume that an external reasoner is available to infer such axioms.

Bayesian Estimation of Gaussians Suppose a set of data points $x_1, ..., x_n \in \mathbb{R}$ have been generated from a univariate Gaussian distribution G with a known variance σ and an unknown mean. Rather than estimating a single value μ^* of this mean, in the Bayesian setting we estimate a probability distribution M over possible means. Suppose our prior beliefs about the mean are modelled by the Gaussian $P = \mathcal{N}(\mu_P, \sigma_P^2)$. After observing the data points $x_1, ..., x_n$ our beliefs about μ are then modelled by the distribution M defined by:

$$M(\mu) \propto p(x_1, ..., x_n \,|\, \mu, \sigma^2) \cdot P(\mu) \tag{1}$$

It can be shown that this distribution M is a Gaussian $\mathcal{N}(\mu_M, \sigma_M^2)$, where:

$$\sigma_M^2 = \frac{\sigma^2 \sigma_P^2}{n\sigma_P^2 + \sigma^2} \qquad \mu_M = \sigma_M^2 \left(\frac{\mu_P}{\sigma_P^2} + \frac{\sum_i x_i}{\sigma^2} \right)$$

Now consider a setting where the variance of the Gaussian is unknown, but the mean is known to be μ. For computational reasons, prior beliefs on the variance are usually modelled using an inverse χ^2 distribution (or a related distribution such as inverse Gamma); let us write this as $Q = \chi^{-2}(\nu_Q, \sigma_Q^2)$. Intuitively, this means that we a priori believe the variance is approximately σ_Q^2, with ν_Q expressing the strength of this belief. After observing the data, our beliefs about the possible values of σ^2 are modelled by the distribution S, defined by:

$$S(\sigma^2) \propto p(x_1, ..., x_n \,|\, \mu, \sigma^2) \cdot P(\sigma^2) \tag{2}$$

It can be shown that $S = \chi^{-2}(\nu_S, \sigma_S^2)$ where:

$$\nu_S = \nu_Q + n \qquad \sigma_S^2 = \frac{\nu_Q \sigma_Q^2 + \sum_i (x_i - \mu)^2}{\nu_S}$$

4 Learning Concept Representations

We assume that a description logic ontology is given, and that for each individual a mentioned in the ABox of this ontology, a vector representation $v_a \in \mathbb{R}^n$ is available. For our experiments, these representations will be obtained using the MEmbER model, although in principle other vector space models could also be used. The task we consider is to learn a Gaussian $G_C = (\mu_C, \Sigma_C)$ for each concept which is mentioned in the TBox or ABox, as well as for all constituents of these concepts (e.g. if the concept $C_1 \sqcup ... \sqcup C_k$ is mentioned then we also learn Gaussians for $C_1, ..., C_k$), along with a scaling factor $\lambda_C > 0$, such that the probability that the individual a belongs to concept C is given by:

$$P(C|v_a) = \lambda_C \cdot G_C(v_a) \tag{3}$$

Intuitively, the variance of G_C encodes how much the instances of C are dispersed across the space, while λ_C allows us to control how common such instances are. Formally, if we assume that the prior on v_a is uniform, λ_C is proportional to the prior probability $P(C)$ that an individual belongs to C.

Given that the number of known instances of the concept C might be far lower than the number of dimensions n, it is impossible to reliably learn the covariance matrix Σ_C without imposing some drastic regularity assumptions. To this end, we will make the common simplifying assumption that Σ_C is a diagonal matrix. The problem of estimating the multivariate Gaussian G_C then simplifies to the problem of estimating n univariate Gaussians. In the following, for a multivariate Gaussian $G = \mathcal{N}(\mu_G, \Sigma_G)$, we write $\mu_{G,i}$ for the i^{th} component of μ_G and $\sigma_{G,i}^2$ for the i^{th} diagonal element of Σ_G.

To find the parameters of these Gaussians, we will exploit background knowledge about the logical relationships between the concepts. However, this means that the parameters of the Gaussian corresponding to some concept C may depend on the parameters of the Gaussians corresponding to other concepts. To cope with the fact that this may result in cyclic dependencies, we will rely on Gibbs sampling, which is explained next. This process will crucially rely on the construction of informative priors on the parameters of the Gaussians, which is discussed in Sections 4.2 and 4.3. Finally, Section 4.4 explains how the scaling factors λ_C are estimated.

4.1 Gibbs Sampling

The purpose of Gibbs sampling is to generate sequences of parameters $\mu_C^0, \mu_C^1, ...$ and $\Sigma_C^0, \Sigma_C^1, ...$ for each concept. To make predictions, we will then average over the samples in these sequences. We will write $\mu_{C,i,j}$ for the i^{th} component of μ_C^j and $\sigma^2_{C,i,j}$ for the i^{th} diagonal element of Σ_C^j.

The initial parameters μ_C^0 and Σ_C^0 are chosen as follows. If $v_1, ..., v_k$ are the vector representations of the known instances of C and $k \geq 2$, we choose:

$$\mu_C^0 = \frac{1}{k}\sum_l v_l \qquad \sigma^2_{C,i,0} = \frac{1}{k-1}\sum_l (v_{l,i} - \mu_{C,i,0})^2$$

where we write $v_{l,i}$ for the i^{th} coordinate of v_l. If $k \leq 1$ the parameters μ_C^0 and $\sigma^2_{C,i,0}$ are estimated based on the superconcepts of C in the ontology; more details about these corner cases are provided in [Bouraoui and Schockaert, 2018].

After the initial parameters have been chosen, we repeatedly iterate over all concepts. In the i^{th} iteration ($i > 0$), we choose the next samples μ_C^i and Σ_C^i for each concept C, according to (1) and (2) respectively. To do this, however, we first need to define prior probabilities on Σ_C^i and μ_C^i. These prior probabilities will be constructed by taking into account the available background knowledge about how the different concepts are interrelated, as we explain in detail in Sections 4.2 and 4.3. In particular, the prior probabilities on Σ_C^i and μ_C^i will be defined in function of the parameters of the Gaussians of the other concepts. When using Gibbs sampling, we always use the most recent samples of the parameters of these other concepts.

For the ease of presentation, we will write μ_B^* and Σ_B^* for the most recent samples of μ_B and Σ_B. In other words, $\mu_B^* = \mu_B^{i-1}$ or $\mu_B^* = \mu_B^i$ holds, depending on whether μ_B was already updated in the current iteration of the Gibbs sampler, and similar for Σ_B^*. We also write G_B^* for $\mathcal{N}(\mu_B^*, \Sigma_B^*)$, i.e. the most recent estimation of the Gaussian G_B. We also use the notations $\mu_{B,i,*}$ and $\sigma^2_{B,i,*}$ to refer to the i^{th} component of μ_B^* and the i^{th} diagonal element of Σ_B^* respectively.

4.2 Priors on the Mean

The type of information that is available to construct a prior on the mean μ_C^i is different for atomic and for complex concepts, which is why we discuss these cases separately.

Atomic Concepts

For an atomic concept A, we use two types of information to construct the prior $P_A = \mathcal{N}(\mu_{P_A}, \Sigma_{P_A})$ that is used for sampling μ_A^i. First, the TBox may contain a number of axioms of the form $A \sqsubseteq C_1, ..., A \sqsubseteq C_k$. If $A \sqsubseteq C_l$ holds then μ_A should correspond to a plausible instance of C_l. In particular, we would expect the probability $G^*_{C_l}(\mu_A)$ to be high.

Second, if a vector representation v_A of the concept A itself is available, it can also provide us with useful information about the likely values of μ_A. Suppose $B_1, ..., B_r$ are atomic concepts such that the TBox contains or implies the axioms $B_1 \sqsubseteq C_l, ..., B_r \sqsubseteq C_l$ (in addition to $A \sqsubseteq C_l$). We will refer to $B_1, ..., B_r$ as the siblings of A w.r.t. C_l. The information that we want to encode in the prior P_A is that the vector differences $v_{B_1} - \mu^*_{B_1}, ..., v_{B_r} - \mu^*_{B_r}$ should all be similar to the vector difference $v_A - \mu_A$. This is motivated by the fact that, in the context of word embeddings, analogical word pairs typically have similar vector differences [Mikolov *et al.*, 2013; Vylomova *et al.*, 2016]. In particular, it corresponds to the intuitive assumption that the relation between the prototype of a concept and the vector space embedding of the concept name should be analogous for all concepts, and in particular for all siblings of A. This intuition can be encoded by estimating a Gaussian $E_{C_l} = \mathcal{N}(\mu_{E_{C_l}}, \Sigma_{E_{C_l}})$ from the vector differences $v_{B_1} - \mu^*_{B_1}, ..., v_{B_r} - \mu^*_{B_r}$ and the representations $v_{B_1}, ..., v_{B_r}$ themselves, as follows (assuming $r \geq 2$):

$$\mu_{E_{C_l}} = v_A + \frac{1}{r}\sum_{u=1}^{r}(\mu^*_{B_u} - v_{B_u})$$

$$\sigma^2_{E_{C_l},j} = \frac{1}{r-1}\sum_{u=1}^{r}(v_{A,j} + \mu^*_{B_u,j} - v_{B_u,j} - \mu_{E_{C_l},j})^2$$

where $v_{A,j}$ is the j^{th} coordinate of v_A, and similar for $v_{B_u,j}$. Using this Gaussian, we can encode our intuition by requiring that $E_{C_l}(\mu_A)$ should be high. If $r = 1$, then $\sigma^2_{E_{C_l},j}$ is estimated based on the superconcepts of C_l; details about this corner case can be found in the online appendix.

Combining both types of background knowledge, we choose a prior P_A which encodes that $P_A(\mu_A)$ is proportional to $G^*_{C_1}(\mu_A) \cdot ... G^*_{C_l}(\mu_A) \cdot E_{C_1}(\mu_A) \cdot ... \cdot E_{C_l}(\mu_A)$, as follows:

$$\sigma^2_{P_A,j} = \left(\sum_{u=1}^{k}\frac{1}{\sigma^2_{C_u,j,*}} + \frac{1}{\sigma^2_{E_{C_u},j}}\right)^{-1}$$

$$\mu_{P_A,j} = \sigma^2_{P_A,j}\sum_{u=1}^{k}\left(\frac{\mu_{C_u,j,*}}{\sigma^2_{C_u,j,*}} + \frac{\mu_{E_{C_u},j}}{\sigma^2_{E_{C_u},j}}\right)$$

For the ease of presentation, here we have assumed that A has at least one sibling w.r.t. each C_l. In practice, if this is not the case, the corresponding Gaussian $E_{C_l}(\mu_A)$ is simply omitted. Note that when the translation assumption underlying the Gaussians E_{C_l} is not satisfied, the associated variances $\sigma^2_{E_{C_u},j}$ will be large, and accordingly the information from the vector space embeddings will be mostly ignored.

Complex Concepts

To construct a prior on μ_C^i for a complex concept C, we can again use concept inclusion axioms of the form $C \sqsubseteq C_1, ..., C \sqsubseteq C_k$ entailed from the ontology, but for complex concepts we do not have access to a vector space representations. However, additional prior information can be derived from the atomic concepts and roles from which C is constructed. For example, let $C \equiv D_1 \sqcap ... \sqcap D_s$. Then we can make the assumption that $P(C|v)$ is proportional to $G^*_{D_1}(c) \cdot ... \cdot G^*_{D_s}(c)$. The product of these Gaussians is proportional to a Gaussian H^*_C with the following parameters:

$$\sigma^2_{H_C,j} = \left(\sum_{i=1}^{s} \frac{1}{\sigma^2_{G_{D_i},j,*}} \right)^{-1}$$

$$\mu_{H_C,j} = \sigma^2_{H_C,j} \sum_{i=1}^{s} \frac{\mu_{G_{D_i},j,*}}{\sigma^2_{G_{D_i},j,*}}$$

This leads to the following choice for the prior:

$$\sigma^2_{P_C,j} = \left(\frac{1}{\sigma^2_{H_C,j,*}} + \sum_{u=1}^{k} \frac{1}{\sigma^2_{C_u,j,*}} \right)^{-1}$$

$$\mu_{P_C,j} = \sigma^2_{P_C,j} \left(\frac{\mu_{H_C,j,*}}{\sigma^2_{H_C,j,*}} + \sum_{u=1}^{k} \frac{\mu_{C_u,j,*}}{\sigma^2_{C_u,j,*}} \right)$$

For complex concepts of the form $D_1 \sqcup ... \sqcup D_s$, $\exists R\,.\,C$ and $\forall R\,.\,C$, a similar strategy can be used. Details about these cases can be found in the online appendix.

4.3 Priors on the Variance

For a concept C, we write $Q_{C,j} = \chi^{-2}(\nu_{Q_{C,j}}, \sigma^2_{Q_{C,j}})$ for the prior on $\sigma^2_{C,j}$, the j^{th} diagonal element of Σ_C. We now discuss how the parameters $\nu_{Q_{C,j}}$ and $\sigma^2_{Q_{C,j}}$ are chosen.

Atomic Concepts

Let A be an atomic concept. We will exploit two types of information about the variance $\sigma^2_{A,j}$. First, if $A \sqsubseteq C$ then $\sigma^2_{A,j} \leq \sigma^2_{C,j}$ should hold. Second, we might expect that the covariance matrix of G_A is similar to that of the most closely related concepts. Specifically, let $B_1, ..., B_k$ be all the atomic siblings of A (i.e. each B_l is an atomic concept, and there is some C such that both $A \sqsubseteq C$ and $B_l \sqsubseteq C$ appear in the TBox). Then one possibility is to choose $\sigma^2_{Q_{C,j}}$ as the average of $\sigma^2_{B_1,j}, ..., \sigma^2_{B_k,j}$. If we have a vector space representation for A and for some of its (atomic) siblings, then we can improve on this estimation by only considering the most similar siblings, i.e. the siblings whose vector space representation is closest in terms of Euclidean distance. In particular, let $\mathcal{B} \subseteq \{B_1, ..., B_k\}$ be the set of the κ most similar siblings of A, and let $C_1, ..., C_l$ be the set of concepts for which the TBox contains the axiom $A \sqsubseteq C_l$, then we choose:

$$\sigma^2_{Q_{A,j}} = \min\left(\min_{l} \sigma^2_{C_l,j,*}, \frac{1}{\kappa} \sum_{B \in \mathcal{B}} \sigma^2_{B,j,*} \right)$$

The parameter $\nu_{Q_{C,j}}$ intuitively reflects how strongly we want to impose the prior on $\sigma^2_{A,j}$. Given that even closely related concepts could have a considerably different variance, we set $\nu_{Q_{A,j}}$ as a small constant η for each A and j.

Complex Concepts

For a complex concept C, the covariance matrix of H^*_C, with H^*_C the Gaussian constructed for complex concepts in Section 4.2, can be used to define the prior on $\sigma^2_{C,j}$. Let $C_1, ..., C_k$ be the concepts for which the TBox contains or implies the axiom $C \sqsubseteq C_l$ then we choose:

$$\sigma^2_{Q_{C,j}} = \min(\min_{l} \sigma^2_{C_l,j,*}, \sigma^2_{H_C,j,*})$$

Furthermore, we again set $\nu_{Q_{C,j}} = \eta$.

4.4 Making Predictions

For a given individual with vector representation v, we can estimate the probability $P(C|v)$ that this individual is an instance of concept C as follows:

$$P(C|v) = \frac{\lambda_C}{N} \sum_{i=1}^{N} p(v; \mu^i_C, \Sigma^i_C) \tag{4}$$

Note that compared to (3), in (4) the parameters of G_C are averaged over the Gibbs samples. As usual with Gibbs sampling, the first few samples are discarded, so here μ^1_C and Σ^1_C refer to the first samples after the burn-in period.

The number of Gibbs sample that we used is equal to 1000 where each sample is generated after 25 every 25 iterations. The burn-in period that we use is fixed to 200 samples.

We estimate the scaling factor λ_C by maximizing the likelihood of the training data. In particular, let $v_1, ..., v_s$ be the vector representations of the known instances of C, and let $u_1, ..., u_r$ be the vector representations of the individuals which are not asserted to belong to C. Then we choose the value of λ_C that maximizes the following expression:

$$\sum_{i=1}^{s} \log(\lambda_C P(v_i|C)) + \sum_{i=1}^{r} \log(1 - \lambda_C P(u_i|C))$$

which is equivalent to maximizing:

$$s\lambda_C + \sum_{i=1}^{r} \log(1 - \lambda_C P(u_i|C)) \tag{5}$$

Note that for concepts without any known instances, we would obtain $\lambda_C = 0$, which is too drastic. To avoid this issue, we replace s by $s + 1$ in (5), which is similar in spirit to the use of Laplace smoothing when estimating probabilities from sparse frequency counts. Furthermore note that the estimation of λ_C relies on a closed world assumption, i.e. we implicitly assume that individuals do not belong to C if they are not asserted to belong to C. Since this assumption may not be correct (i.e. some of the individuals $u_1, ..., u_r$ may actually be instances of C, even if they are not asserted to be so), the value of λ_C we may end up with could be too low. This is typically not a problem, however, since it simply means that the predictions we make might be more cautious then they need to be.

5 Experimental Results

In this section, we experimentally evaluate our method[2] against a number of baseline methods. In our experiments, we have used SUMO[3], which is a large open-domain ontology.

[2] Implementation and data available at https://github.com/flexilogalgo

[3] http://www.adampease.org/OP/

	SVM-Linear				SVM-Quad				Gibbs			
	Pr	Rec	F1	AP	Pr	Rec	F1	AP	Pr	Rec	F1	AP
$1 \leq \lvert X\rvert \leq 5$	0.033	0.509	0.062	0.055	0.086	0.046	0.060	0.144	0.258	0.508	0.343	0.328
$5 < \lvert X\rvert \leq 10$	0.084	0.922	0.154	0.067	0.116	0.404	0.180	0.163	0.202	0.474	0.283	0.340
$10 < \lvert X\rvert \leq 50$	0.111	0.948	0.199	0.081	0.151	0.382	0.216	0.247	0.242	0.886	0.380	0.276
$\lvert X\rvert > 50$	0.153	0.217	0.180	0.230	0.224	0.721	0.342	0.260	0.361	0.678	0.471	0.404

Table 1: Results of the proposed model and the baselines.

	Gibbs-flat				Gibbs-emb				Gibbs-DL			
	Pr	Rec	F1	AP	Pr	Rec	F1	AP	Pr	Rec	F1	AP
$1 \leq \lvert X\rvert \leq 5$	0.212	0.416	0.281	0.290	0.201	0.540	0.293	0.262	0.226	0,498	0.311	0.304
$5 < \lvert X\rvert \leq 10$	0.186	0.368	0.247	0.273	0.173	0.357	0.233	0.262	0.417	0.192	0.263	0.328
$10 < \lvert X\rvert \leq 50$	0.199	0.496	0.284	0.210	0.207	0.513	0.295	0.233	0.218	0.670	0.329	0.251
$\lvert X\rvert > 50$	0.316	0.312	0.314	0.328	0.321	0.373	0.345	0.321	0.344	0.450	0.390	0.369

Table 2: Results for the variants of the proposed model.

An important advantage of using SUMO is that several of its concepts and individuals are explicitly mapped to WordNet, which itself is linked to WikiData. This means that we can straightforwardly align this ontology with our entity embedding. For concepts and individuals for which we do not have such a mapping, we use BabelNet[4] to suggest likely matches. We split the set of individuals into a training set I_{train} containing 2/3 of all individuals, and a test set I_{test} containing the remaining 1/3. All ABox assertions involving individuals from I_{train} are used as training data.

The considered evaluation task is to decide for a given assertion $A(a)$ (meaning "a is an instance of A") whether it is correct or not. As positive examples, we use all assertions from the ABox involving individuals from I_{test}. To generate negative test examples, we use the following strategies. First, for each positive example $A(a)$ and each concept $B \neq A$ such that the TBox implies $A \sqsubseteq B$, we add a negative example by randomly selecting an individual x such that $B(x)$ can be deduced from SUMO while $A(x)$ cannot. Second, for each positive example $A(a)$, we also add 10 negative examples by randomly selecting individuals among all those that are not known to be instances of x. Note that even if $A(x)$ is not asserted by SUMO, it may be the case that x is an instance of A. This means that in a very small number of cases, the selected negative examples might actually be positive examples. The reported results are thus a lower bound on the actual performance of the different methods. Importantly, the relative performance of the different methods should not be affected by these false negatives.

The performance is reported in terms of average precision (AP), and micro-averaged precision (Pr), recall (Rec) and F1 score. To compute the AP scores, we rank the assertions from the test data (i.e. the correct ABox assertions as well as the constructed negative examples), across all considered concepts, according to how strongly we believe them to be correct, and then we compute the average precision of that ranking. To give a clearer picture of the performance of the different methods, however, we will break up the results according to the number of training examples we have for each concept (see below). Note that AP only evaluates our ability to rank individuals, and hence does not depend on the scaling factors λ_A. The precision, recall and F1 scores, however, do require us to make a hard choice.

As baselines, we have considered a linear and quadratic support vector machine (SVM). We will refer to our model as *Gibbs*. We also consider three variants of our method: *Gibbs-flat*, in which flat priors are used (i.e. no dependencies between concepts are taken into account), *Gibbs-emb*, in which the priors on the mean and variance are only obtained from the embedding (i.e. no axioms from the TBox are taken into account), and *Gibbs-DL*, in which the priors on the mean and variance are only obtained from the TBox axioms (i.e. the embedding is not taken into account).

The results are summarized in Table 2, where $|X|$ refers to the number of training examples for concept X. Overall, our model consistently outperforms the baselines in both F1 and MAP score. For concepts with few known instances, the gains are substantial. Somewhat surprisingly, however, we even see clear gains for larger concepts. Regarding the variants of our model, it can be observed that using TBox axioms to estimate the priors on the mean and variance (*Gibbs-DL*) leads to better results than when a flat prior is used. The model *Gibbs-emb*, however, does not outperform *Gibbs-flat*. This means that the usefulness of the embedding, on its own, is limited. However, the full model, where the embedding is combined with the TBox axioms, does perform better than *Gibbs-DL*.

6 Conclusions and Future Work

We have proposed a method for learning conceptual space representations of concepts. In particular, we associate with each concept a Gaussian distribution over a learned vector space embedding, which is in accordance with some implementations of prototype theory. In contrast to previous work, we explicitly take into account known dependencies between the different concepts when estimating these Gaussians. To this end, we take advantage of description logic axioms and information derived from the vector space representation of the concept names. This means that we can often make faithful predictions even for concepts for which only a few known instances are specified.

[4] BabelNet Java API available at http://babelnet.org

Acknowledgments

This work was supported by ERC Starting Grant 637277.

References

[Alfarone and Davis, 2015] Daniele Alfarone and Jesse Davis. Unsupervised learning of an IS-A taxonomy from a limited domain-specific corpus. In *Proc. IJCAI*, pages 1434–1441, 2015.

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, New York, NY, USA, 2003.

[Bordes *et al.*, 2013] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proc. NIPS*, pages 2787–2795. 2013.

[Bouraoui and Schockaert, 2018] Zied Bouraoui and Steven Schockaert. Learning conceptual space representations of interrelated concepts. *CoRR*, abs/1805.01276, 2018.

[Bouraoui *et al.*, 2017] Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. Inductive reasoning about ontologies using conceptual spaces. In *Proc. AAAI*, pages 4364–4370, 2017.

[Chella *et al.*, 1997] Antonio Chella, Marcello Frixione, and Salvatore Gaglio. A cognitive architecture for artificial vision. *Artif. Intell.*, 89(1-2):73–111, 1997.

[Chella *et al.*, 2003] Antonio Chella, Marcello Frixione, and Salvatore Gaglio. Conceptual spaces for anchoring. *Robotics and Autonomous Systems*, 43(2-3):193–195, 2003.

[Douven *et al.*, 2013] I. Douven, L. Decock, R. Dietz, and P. Égré. Vagueness: A conceptual spaces approach. *Journal of Philosophical Logic*, 42:137–160, 2013.

[Forth *et al.*, 2010] J. Forth, G. A Wiggins, and A. McLean. Unifying conceptual spaces: Concept formation in musical creative systems. *Minds and Machines*, 20:503–532, 2010.

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

[Hearst, 1992] Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proc. COLING*, pages 539–545, 1992.

[Jäger, 2009] Gerhard Jäger. Natural color categories are convex sets. In *17th Amsterdam Colloquium on Logic, Language and Meaning*, pages 11–20, 2009.

[Jameel *et al.*, 2017] Shoaib Jameel, Zied Bouraoui, and Steven Schockaert. MEmbER: Max-margin based embeddings for entity retrieval. In *Proc. SIGIR*, pages 783–792, 2017.

[Kok and Domingos, 2007] Stanley Kok and Pedro Domingos. Statistical predicate invention. In *Proc. ICML*, pages 433–440, 2007.

[Kozareva and Hovy, 2010] Zornitsa Kozareva and Eduard Hovy. A semi-supervised method to learn and construct taxonomies using the web. In *Proc. EMNLP*, pages 1110–1118, 2010.

[Mikolov *et al.*, 2013] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proc. NAACL-HLT*, pages 746–751, 2013.

[Mintz *et al.*, 2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proc. ACL*, pages 1003–1011, 2009.

[Murphy, 2007] Kevin Murphy. Conjugate Bayesian analysis of the Gaussian distribution. Technical report, University of British Columbia, 2007.

[Neelakantan and Chang, 2015] Arvind Neelakantan and Ming-Wei Chang. Inferring missing entity type instances for knowledge base completion: New dataset and methods. In *Proc. NAACL*, pages 515–525, 2015.

[Palatucci *et al.*, 2009] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In *Proc. NIPS*, pages 1410–1418, 2009.

[Riedel *et al.*, 2010] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Proc. ECML/PKDD*, pages 148–163, 2010.

[Rocktäschel and Riedel, 2016] Tim Rocktäschel and Sebastian Riedel. Learning knowledge base inference with neural theorem provers. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 45–50, 2016.

[Romera-Paredes and Torr, 2015] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *Proc. ICML*, pages 2152–2161, 2015.

[Rosch, 1973] Eleanor H Rosch. Natural categories. *Cognitive Psychology*, 4(3):328–350, 1973.

[Rosseel, 2002] Yves Rosseel. Mixture models of categorization. *Journal of Mathematical Psychology*, 46:178–210, 2002.

[Socher *et al.*, 2013] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Proc. NIPS*, pages 935–943, 2013.

[Sourek *et al.*, 2016] Gustav Sourek, Suresh Manandhar, Filip Zelezný, Steven Schockaert, and Ondrej Kuzelka. Learning predictive categories using lifted relational neural networks. In *Proc. ILP*, pages 108–119, 2016.

[Vylomova *et al.*, 2016] Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proc. ACL*, 2016.