Learning to Represent Knowledge Graphs with Gaussian Embedding

Shizhu He, Kang Liu, Guoliang Ji and Jun Zhao
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
{shizhu.he, kliu, guoliang.ji, jzhao}@nlpr.ia.ac.cn

ABSTRACT

The representation of a knowledge graph (*KG*) in a latent space recently has attracted more and more attention. To this end, some proposed models (e.g., TransE) embed entities and relations of a KG into a *"point"* vector space by optimizing a global loss function which ensures the scores of positive triplets are higher than negative ones. We notice that these models always regard all entities and relations in a same manner and ignore their (un)certainties. In fact, different entities and relations may contain different certainties, which makes identical certainty insufficient for modeling. Therefore, this paper switches to *density-based* embedding and propose **KG2E** for explicitly modeling the certainty of entities and relations, which learn the representations of KGs in the space of multi-dimensional Gaussian distributions. Each entity/relation is represented by a Gaussian distribution, where the mean denotes its position and the covariance (currently with diagonal covariance) can properly represent its certainty. In addition, compared with the symmetric measures used in *point-based* methods, we employ the KL-divergence for scoring triplets, which is a natural asymmetry function for effectively modeling multiple types of relations. We have conducted extensive experiments on link prediction and triplet classification with multiple benchmark datasets (WordNet and Freebase). Our experimental results demonstrate that our method can effectively model the (un)certainties of entities and relations in a KG, and it significantly outperforms state-of-the-art methods (including TransH and TransR).

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Knowledge Graph Representation*

Keywords

Distributed Representation, Gaussian Embedding, Knowledge Graph

1. INTRODUCTION

Knowledge representation and reasoning (*KR&R*) is a fundamental issue for artificial intelligence (*AI*) and knowledge management (*KM*) [6]. To fulfill this aim, recent researchers devote to knowledge graph (*KG*) which provides an effective represent mechanism for knowledge and has become useful resources to support many intelligent applications, such as expert system [10], web search [24][27] and question answering [31][2][11]. Commonly, a KG, such as Freebase[1] [1], NELL[2] [7] or WordNet[3] [18], describes knowledge as many relational data and represent them as interlinked subject-property-object (*SPO*) triplet facts. Usually, a triplet fact (*head entity, relation, tail entity*) (denoted as (*h, r, t*)) consists of two entities and a relation between them.

With the expansion of domains and the increase of data size, representation of KGs is required to support generalization, robust inference and other desirable functionalities [23]. However, traditional representations of KGs are based on a (*hard*) symbolic logic representation framework, which heavily rely on the learned logic inference rules for knowledge reasoning [15][29]. Thus, they lack certain ability for supporting numerical computation in continuous spaces, and cannot be effectively extended to large-scale KGs, such as Freebase. To address this problem, a new approach based on representation learning was recently proposed by attempting to embed a KG into a low-dimensional continuous vector space that preserves certain properties of the original graph [19][13][5]. The representations in (*soft*) latent space could act as a supplement for the symbolic representations, which are learned by optimizing a global loss function that involves all entities and relations in the entire graph. As a result, each entity and relation encodes global KG information through mutual effects and restrictions with the others. These embedding representations can be used in many applications, for example, verifying the correctness of one triplet fact, predicting the relations between two entities and reasoning about the implications among relations.

The promising methods (introduced in Section "Related Work") usually represent an entity as an n-dimensional vector $\mathbf{h}$ (or $\mathbf{t}$) and regard it as a *"point"* in low-dimensional spaces. A relation in KGs is represented as an operation between two *"points"* (e.g., translation as a vector [4], linear transformation as a matrix [20], and mixed operation [25]). In this way, a relation-dependent scoring function $f_r(h,t)$, such as $-||\mathbf{h}+\mathbf{r}-\mathbf{t}||_{\ell_{1/2}}$, is defined to measure the correctness of the fact (h,r,t) in the embedding space. The embedding of a KG is learned to ensure that the score of a positive triplet (e.g., (h,r,t)) is higher (or lower) than that of a corresponding (mostly corrupted) negative triplet (e.g., (h',r,t)).

Based on this paradigm, multiple models are proposed, such as TransE [4], TransH [30], and TransR [16]. Although these models are proved to be effective in many scenarios, we notice that

CIKM'15, October 19–23, 2015, Melbourne, Australia.

DOI: http://dx.doi.org/10.1145/2806416.2806502.

[1] www.freebase.com/
[2] www.rtw.ml.cmu.edu/rtw/
[3] www.wordnet.princeton.edu/

different entities and relations often share the same margin when separating a positive triplet and its corresponding negative triplet, and the (un)certainties of entities and relations in KGs are totally neglected. In fact, different entities and relations may contain different certainties, which makes identical certainty insufficient for modeling. We consider that the (un)certainty of one entity/relation represents the confidence for indicating its semantic when scoring a triplet as context. For example, the certainty of relation *spouse* is obviously larger than *nationality* when inferring a person (e.g., for predicting *Hillary Clinton*, we may have more confidence to know who is she when knowing her husband (*spouse*) is *Bill Clinton* than knowing she was born on (*nationality*) *USA*.). Thus, we argue that if we set a larger margin for separating (*spouse*) related positive and negative triplets in the embedding model, we could obtain better performances.

In this paper, we argue that the (un)certainties in KGs could be influenced by multiple factors, including imbalance between the relation's head and tail, different number of the linked triplets for different relations and entities, and the ambiguous of the relations and so on. For example, we hypothesize that an entity containing fewer triplets has more uncertainty, and a relation linking more triplets with more complex contexts has more uncertainty as well. We especially perform statistics on the entities and relations in Freebase[4], and Figure 1 shows the statistics of the *person* type in the *people* domain. In Figure 1, the upper left indicates the diversity numbers of triplets contained in 6 randomly sampled entities; the upper right expresses the distribution of the ratios of entity numbers in the head and tail parts with some typical relations; and the bottom shows the distribution of triplet numbers with multiple relations. This figure illustrates that popular entities (e.g., the politician *Hillary Clinton*) contain more relations and facts than unpopular ones (e.g., the writer *Murray Silverstein*). Furthermore, different parts of relations can contain very different numbers of entities, (such as the *head* part and *tail* part in *gender* or *nationality*), and, high-frequency relations (e.g., *nationality*) link more entity pairs than low-frequency ones (e.g., *religion*). It indicates that the variations of uncertainty with different entities and relations in a KG is vary enormously, and it is desirable to consider this problem when learning the representations of a KG in a latent space.

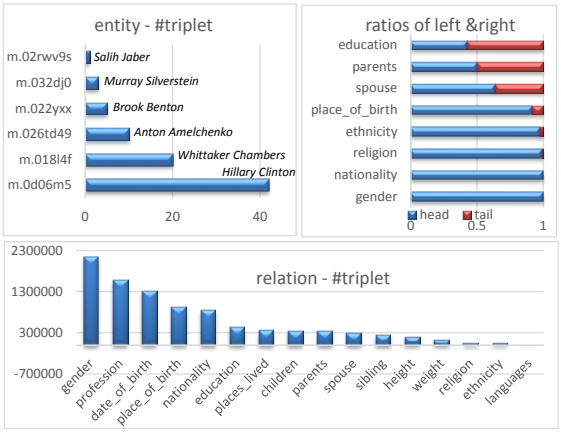

Figure 1: Various densities in a KG.

To address the aforementioned problem, this paper proposes a new *density-based* embedding method, **KG2E**[5], to model KGs and learn the representations in the space of multi-dimensional Gaussian distributions. Inspired by [28], our approach models each entity and relation with a multi-dimensional Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (currently with diagonal covariance for computing efficiency). The mean vector of such multi-dimensional Gaussian distribution indicates its position, and the covariance matrix indicates the corresponding (un)certainty which impacts on others, as shown in Figure 2. Similar to previous methods, we also design a scoring function $f_r(h,t)$ to measure the correctness of the triplet fact (h,r,t) in the embedding space. In our scoring function, the distributions between $\mathcal{H}-\mathcal{T}$ and $\mathcal{R}$ are similar when (h,r,t) holds, where $\mathcal{H}$, $\mathcal{R}$ and $\mathcal{T}$ indicate the Gaussian distributions of h, r and t, respectively. Different from previous methods, which adopts *point-based* scoring functions (dot products, cosine distance, $\ell_{1/2}$ distances, etc.), we employ the KL-divergence between two probability distribution (the entity-pair distribution and the relation distribution) as the scoring function for all triples in KGs, which is a naturally asymmetric and effective way for incorporating covariance (denotes (un)certainties of entities and relations in KGs) into the model. Moreover, we will employ another scoring function based on the expected likelihood [12] to inspect the different performances of asymmetric and symmetric measures.

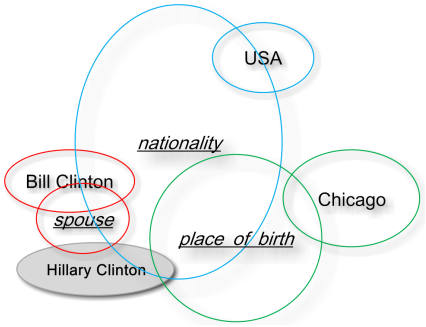

Figure 2: Illustrated of the means and (diagonal) variances of entities and relations in a Gaussian Embedding. The label indicates its position. *Relations* attach with underlined labels. Circles with the same color indicate a fact for *Hillary Clinton*. In the representations, we might infer that *Hillary Clinton* was born in (*place_of_birth*) *Chicago*, and is an (*nationality*) *American*.

We have conducted extensive experiments on link prediction and triplet classification with multiple benchmark datasets such as WordNet and Freebase. The experimental results demonstrate the effectiveness of our method. In particular, the proposed model can effectively address one-to-many, many-to-one and reflexive relations, and significantly outperforms state-of-the-art methods (including TransH [30] and TransR [16]) by as much as ∼30% when predicting head (tail) entities in many-to-one (one-to-many) relations.

In summary, our main contributions are as follows:

- We propose a new method for learning the representations of a KG. Different from previous methods, we specially consider the (un)certainties of entities and relations the a KG.
- We model each entity/relation into a Gaussian distribution. In particular, the mean vector denote its position and the covariance matrix (currently diagonal for computing efficiency) are used to describe the (un)certainties of the entities and relations. To our knowledge, the proposed method is the first work which models KGs with *"density-based"* embedding, in contrast to the existing *"point-based"* embedding methods.

[4]We use the dump data released on 2014-06-29.

[5]This name has two meanings. The first indicates mapping **K**nowledge **G**raph *to* **E**mbedding and the second indicates the representation of a **KG** with **G**aussian **E**mbedding.

- We use two methods (symmetric and asymmetric) to compute the scores of triplets and find that the asymmetric method (KL-divergence between two Gaussian distributions) is more suitable for learning the representation of a KG with Gaussian embedding.
- The experimental results demonstrate that our proposed method can effectively model the (un)certainty of entities and relations in a KG, and it significantly outperforms state-of-the-art methods in multiple related tasks.

2. RELATED WORK

Currently, the proposed methods mainly represent KGs in a low-dimensional latent space. We briefly summarize the most relevant work in Table 1[6]. These methods embed entities into a vector space and define a (mainly relation-dependent) scoring function to measure the compatibility of (h, r, t). The differences in these models are the defined scoring functions $f_r(h, t)$.

We first highlight TransE [4] and its variants (TransH [30] and TransR [16]) because they are simple and effective and achieve the state-of-the-art performance in the majority of related tasks, especially in KGs with thousands of relations. Inspired by the word2vec [17], which finds the learning word vectors with a neural network own linear relations, such as, *vec('Paris') - vec('France')* $\approx$ *vec('Rome') - vec('Italy')*, that is, the difference in word vectors is similar when they are attached to the same relation (e.g., *capital_of*, corresponding to the above example), TransE [4] represents a relation as a vector r indicating the semantic translation from the head entity h to the tail entity t, aiming to satisfy the equation **t - h** $\approx$ **r** when triplet (h, r, t) holds. TransE effectively handles one-to-one relations but has issues in handling one-to-many, many-to-one and many-to-many relations. For example, consider a one-to-many relation **r** with multiple tail entities $\mathbf{t}_i$ satisfying **h + r** $\approx \mathbf{t}_i$ for $\forall i \in \{1, ..., m\}, (h, r, t_i) \in KG$, and it outputs invalid representations ($\mathbf{t}_1$ =...= $\mathbf{t}_m$) for distinguishing entities.

To address the aforementioned issues in TransE, TransH [30] and TransR [16] are proposed to enable an entity to have distinct representations when involved in different relations. For a triplet (h, r, t), TransH first projects a head/tail entity vector (**h/t**) into a relation-dependent hyper-plane by the following formulas: $\mathbf{h}_\perp = \mathbf{h}(\mathbf{I} - \mathbf{w}_r^T\mathbf{w}_r)$ and $\mathbf{t}_\perp = \mathbf{t}(\mathbf{I} - \mathbf{w}_r^T\mathbf{w}_r)$, where $\mathbf{w}_r$ is the vector that spans the hyper-plane. It then measures the score using the function $||\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp||_{\ell_{1/2}}$ in the hyper-plane of the relation r. TransR is slightly different from TransH: it transforms a head/tail entity vector into a relation-dependent sub-space with $\mathbf{h}_r = \mathbf{h}\mathbf{M}_r$ and $\mathbf{t}_r = \mathbf{t}\mathbf{M}_r$, where $\mathbf{M}_r$ represents the transform matrix from entity space to the sub-space of relation r.

However, the TransH and TransR methods only partly address the issues encountered by TransE. For example, consider TransR in a one-to-many relation. It tends to learning $\mathbf{t}_i\mathbf{M}_r = \mathbf{t}_j\mathbf{M}_r$, for $\forall i, j \in \{1, ..., m\}$, (h, r, t_i), and $(h, r, t_j) \in KG$, and as a result, the different part between $\mathbf{t}_i$ and $\mathbf{t}_j$ only depends on the number of eigenvalues equal to zero[7]. In addition, the existing methods have difficulty learning valid representations for reflexive relations because they use the same operation for head and tail entities. For example, when triplets (h, r, t) and (t, r, h) both hold for a reflexive relation r, TransE, TransH and even TransR tend to make (**h** $\approx$ **t**) and (**r** $\approx$ **0**). Whereas **h** equals **t** may be a good character for a reflexive relation, the same vector (**0**) for represents all reflexive relations is not helpful for related tasks. Issues often exist in the *"point" based* embedding models, in which "point" vectors are typically compared by dot products, cosine-distance or $\ell_{1/2}$ norm, all of which provide for symmetric comparison between instances. The proposed *"density" based* methods represent entities and relations by Gaussian distributions that explicitly modeling the uncertainty of KGs and the asymmetry function scores (h, r, t) and (t, r, h) using different parameters associated with not only the head and tail entities but also its order.

To our knowledge, Linear Relational Embedding (LRE) [20] is the pioneer work in learning representations of multi-relational data that represent concepts as vectors and binary relations as transform matrices. For concepts and their relations (i, r, j), LRE learns to maximize the generatation probability from concept i and relation r to concept j in proportion to $\exp(-||\mathbf{R_r}\mathbf{v_i} - \mathbf{v_j}||^2)$. Subsequently, many approaches have followed this line. The unstructured model (UM) [4] was proposed as the simplified version of TransE by assigning all translation vectors **r** = 0. However, it cannot distinguish different relations. Structured embedding (SE) [5] adopts two different relation-specific matrices for head and tail entities but cannot capture precious semantics of relations because the two matrices are separated in optimization. The latent factor model (LFM) [13] considers the second-order correlations between entities embedding with a quadratic form. The single layer model (SLM) and neural tensor network (NTN) were proposed by Socher [25]. The SLM is a naive baseline of the NTN and scores triplets using relation-specific weights $\mathbf{u}_r$ with a non-linear operation (tanh) for triplet representation $W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r$. To date, the NTN is the most expressive model based on multi-layer neural networks. As shown in Table 1, the NTN extends the SLM by considering the second-order correlations between entity embedding (similar to LFM), feeding into a non-linear hidden layer, and then combining with a linear output parameterized by the relation. However, the NTN is not sufficiently simple to handle the large-scale KGs with numerous relations.

In addition to these methods based on the *ranking loss* framework, there is another line of related work that focuses on learning the latent representations for KGs by *tensor (matrix) decomposition and completion*; it was inspired by the wide usage of decomposition in recommended system [14] and relation extraction [32]. The collective matrix factorization model RESCAL [19] was proposed to model KGs, which regarding a KG as a 3-model tensor and learning the latent representations (entity as a vector and relation as a matrix) by reconstructing the original graph. RESCAL can be used to cluster related entities and relations. Clustering concepts have been widely used in modeling multi-relational data, such as tensor factorization based on Bayesian clustering [26] and jointly spectral clustering [8]. We compare our method with RESCAL in our experiments. Vilnis and McCallum firstly proposed the Gaussian Embedding models learning the representations of words [28], which inspired this work. However, they mainly focus on the word representations based on the contexts of text, this work focus on the entity/relation representations in KGs based on the inter-linked relationships between them.

3. REPRESENTING A KG WITH GAUSSIAN EMBEDDING

A Gaussian distribution is capable of representing (un)certainty explicitly. We represent KGs with Gaussian embedding. In this

[6]For convenient comparison, we use a different expression for TransH.

[7]Decompose matrix $\mathbf{M_r}$ with singular value decomposition (SVD): $(\mathbf{t_i} - \mathbf{t_j})\mathbf{M_r} = (\mathbf{t_i} - \mathbf{t_j})\mathbf{S}\begin{pmatrix}\Sigma & 0\\ 0 & 0\end{pmatrix}\mathbf{D} = \mathbf{0}$, ensuring that (most) parts of $\mathbf{t_i}$ and $\mathbf{t_j}$ are equal because the eigenvalues of the upper part of matrix ($\mathbf{S\Sigma D}$) cannot contain zero.

Model	Score function $f_r(h,t)$	# Parameters
LRE (2001) [20]	$\exp(-\|\|\mathbf{W_r h}-\mathbf{t}\|\|^{\mathbf{2}})$, $W_r \in \mathbb{R}^{k_e \times k_e}$	$k_e n_e + k_r n_r^2$
SE (2011) [5]	$\|\|W_{rh}\mathbf{h} - W_{rt}\mathbf{t}\|\|_{\ell_{1/2}}$, $W_{rh}, W_{rt} \in \mathbb{R}^{k_r \times k_e}$	$k_e n_e + 2k_r n_r$
LTM (2012) [13]	$\mathbf{h}^T W_r \mathbf{t}$, $W_R \in \mathbb{R}^{k_e \times k_e}$	$k_e n_e + k_r n_r$
UM (2012) [3]	$\|\|\mathbf{h}-\mathbf{t}\|\|_2^2$	$k_e n_e$
TransE (2013) [4]	$\|\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|\|_{\ell_{1/2}}$, $\mathbf{r} \in \mathbb{R}_r^k$	$k_e(n_e + n_r)$
SLM (2013) [25]	$\mathbf{u}_r^T g(W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$, $\mathbf{u}_r, \mathbf{b}_r \in \mathbb{R}^s$, $W_{rh}, W_{rt} \in \mathbb{R}^{s \times k_e}$	$n_e k_e + n_r(2sk_r + s)$
NTN (2013) [25]	$\mathbf{u}_r^T g(\mathbf{h}^T \mathbf{W}_r \mathbf{t} + W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$, $\mathbf{W}_r \in \mathbb{R}^{k_e \times k_e \times s}$	$n_e k_e + n_r(sk_e^2 + 2sk_e + s)$
TransH (2014) [30]	$\|\|\mathbf{h}(\mathbf{I} - \mathbf{w}_r^T \mathbf{w}_r) + \mathbf{r} - \mathbf{t}(\mathbf{I} - \mathbf{w}_r^T \mathbf{w}_r)\|\|_{\ell_{1/2}}$, $\mathbf{w}_r, \mathbf{r} \in \mathbb{R}_r^k$	$k_e n_e + 2k_r n_r$
TransR (2015) [16]	$\|\|\mathbf{h}\mathbf{M}_r + \mathbf{r} - \mathbf{t}\mathbf{M}_r\|\|_{\ell_{1/2}}$, $\mathbf{r} \in \mathbb{R}_r^k, \mathbf{M}_r \in \mathbb{R}^{k_r \times k_r}$	$k_e n_e + (k_r + k_r^2) n_r$
KG2E_KL (this paper)	$\frac{1}{2}\{tr(\mathbf{\Sigma}_r^{-1}(\mathbf{\Sigma}_h + \mathbf{\Sigma}_t)) + \boldsymbol{\mu}^T \mathbf{\Sigma}_r^{-1} \boldsymbol{\mu} - \log \frac{det(\mathbf{\Sigma}_h + \mathbf{\Sigma}_t)}{det(\mathbf{\Sigma}_r)}\}$, $\boldsymbol{\mu} = \boldsymbol{\mu}_h - \boldsymbol{\mu}_t - \boldsymbol{\mu}_r$	$2k_e n_e + 2k_r n_r$
KG2E_EL (this paper)	$\frac{1}{2}\{\boldsymbol{\mu}^T \mathbf{\Sigma}^{-1} \boldsymbol{\mu} + \log\det \mathbf{\Sigma}\}$, $\mathbf{\Sigma} = \mathbf{\Sigma}_h + \mathbf{\Sigma}_t + \mathbf{\Sigma}_r$	$2k_e n_e + 2k_r n_r$

Table 1: Representative models for representing a KG in a latent space. We mainly compare the models' scoring functions $f_r(\mathbf{h},\mathbf{t})$ and their complexities (the numbers of parameters). n_e and n_r are the number of unique entities and relations, respectively. k_e and k_r are the dimensions of entity and relation in the latent embedding space, $\mathbf{h},\mathbf{t} \in \mathbb{R}^{k_e}$. s is the number of slices of a tensor used in NTN. I indicates the identity matrix used in TransH. $g(x)$ is a non-linear function used in neural networks such as $tanh$.

section, we firstly introduce the framework for learning KG embedding. We then present the proposed **KG2E** method and the learning strategy.

First, we describe some common notations: h, r and t denote the head entity, relation and tail entity for a triplet fact (h,r,t). The mathematical symbols $\mathcal{H}$, $\mathcal{R}$ and $\mathcal{T}$ denote the corresponding Gaussian distributions: $\mathcal{H} \sim \mathcal{N}(\boldsymbol{\mu}_h, \mathbf{\Sigma}_h)$ (similarly for $\mathcal{R}$ and $\mathcal{T}$). The mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$ indicate the corresponding embedding representations for the Gaussian distribution, and $\mathcal{E}$ and $\mathcal{R}$ are the sets of entities and relations in KGs, respectively.

3.1 Background

We follow the *energy-based* framework to learn the representations of a KG, as commonly used in learning word embedding [17] and knowledge graph embedding [4] in a large scale corpus. The core of this framework is an *energy function* $\mathcal{E}_{\boldsymbol{\theta}}(x)$ that scores the input x, parameterized by $\boldsymbol{\theta}$. The goal of *energy-based* learning is to learn the parameters of the energy function to ensure that the score of an observed positive example is higher (or lower, depending on the definition) than those of negative (mainly constructed) examples. This approach is often associated with a loss function $\mathcal{L}$ that provides gradients of the parameters given the predictions of the energy function according to some specific supervision. The framework is also called *ranking loss* learning because the defined loss function is based on the ranks of positive and negative samples.

In *energy-based* KG embedding models, the parameters $\boldsymbol{\theta}$ correspond to our learned representations, and the input x correspond to observed true triplet facts in a KG. That is, the defined *energy function* renders the score of a true fact higher (or lower) than that of a false fact, parameterized with the representations $\boldsymbol{\theta}$.

3.2 Gaussian Embedding for a KG

We will describe the *energy functions* used in our proposed method that measure the score of a triplet (h,r,t). Borrowing concepts from translation-based methods [4][30][16], we consider the transformation result from the head entity to the tail entity to be akin to the relation in the positive triplet. We use the following simple formula to express this transformation: $\mathcal{H} - \mathcal{T}$, which corresponds to the probability distribution $\mathcal{P}_e \sim \mathcal{N}(\boldsymbol{\mu}_h - \boldsymbol{\mu}_t, \mathbf{\Sigma}_h + \mathbf{\Sigma}_t)$ (we hypothesize that the head entity and tail entity are independent with regard to some specific relation). As a result, combined with the probability distribution of relation $\mathcal{P}_r \sim \mathcal{N}(\boldsymbol{\mu}_r, \mathbf{\Sigma}_r)$, the most important step is to measure the similarity between $\mathcal{P}_e$ and $\mathcal{P}_r$.

KL-divergence is a straightforward method of measuring the similarity of two probability distributions and is naturally asymmetric. Moreover, we use another similarity method based on the expected likelihood or probability product kernel [28][12] to inspect the difference in performance between asymmetric and symmetric measures. We illustrate the two similarity measures in detail below.

3.2.1 Asymmetric similarity: KL-divergence

We optimize the following energy function based on the KL divergence between the entity-transformed distribution and relation distribution and denote it as KL.

$$\begin{aligned}\mathcal{E}(h,r,t) &= \mathcal{E}(\mathcal{P}_e, \mathcal{P}_r) = \mathcal{D}_{\mathcal{KL}}(\mathcal{P}_e, \mathcal{P}_r) \\ &= \int_{x \in \mathcal{R}^{k_e}} \mathcal{N}(x; \boldsymbol{\mu}_r, \mathbf{\Sigma}_r) \log \frac{\mathcal{N}(x; \boldsymbol{\mu}_e, \mathbf{\Sigma}_e)}{\mathcal{N}(x; \boldsymbol{\mu}_r, \mathbf{\Sigma}_r)} dx \\ &= \frac{1}{2}\Big\{ tr(\mathbf{\Sigma}_r^{-1}\mathbf{\Sigma}_e) + (\boldsymbol{\mu}_r - \boldsymbol{\mu}_e)^T \mathbf{\Sigma}_r^{-1} (\boldsymbol{\mu}_r - \boldsymbol{\mu}_e) \\ &\qquad - \log \frac{det(\mathbf{\Sigma}_e)}{det(\mathbf{\Sigma}_r)} - k_e \Big\}\end{aligned} \tag{1}$$

In the upper formula, $tr(\mathbf{\Sigma})$ and $\mathbf{\Sigma}^{-1}$ indicate the trace and inverse of the covariance matrix, respectively. Considering the simplified diagonal covariance, we can compute the trace and inverse of the matrix simply and effectively.

The gradient of the log determinant is $\frac{\partial \log detA}{\partial A} = A^{-1}$, the gradient $\frac{\partial x^T A^{-1} y}{\partial A} = -A^{-T} x y^T A^{-T}$, and the gradient $\frac{\partial tr(X^T A^{-1} Y)}{\partial A} = -(A^{-1} Y X^T A^{-1})^T$ [21]. We can compute the gradients of this energy function with respect to the mean vectors and covariance matrix (currently acting as a vector) as follows:

$$\frac{\partial \mathcal{E}(h,r,t)}{\partial \boldsymbol{\mu}_r} = \frac{\partial \mathcal{E}(h,r,t)}{\partial \boldsymbol{\mu}_t} = -\frac{\partial \mathcal{E}(h,r,t)}{\partial \boldsymbol{\mu}_h} = \Delta'_{hrt} \tag{2}$$

$$\frac{\partial \mathcal{E}(h,r,t)}{\partial \mathbf{\Sigma}_r} = \frac{1}{2}(\mathbf{\Sigma}_r^{-1}\mathbf{\Sigma}_e\mathbf{\Sigma}_r^{-1} + \Delta'_{hrt}\Delta'^T_{hrt} + \mathbf{\Sigma}_r^{-1})) \tag{3}$$

$$\frac{\partial \mathcal{E}(h,r,t)}{\partial \mathbf{\Sigma}_h} = \frac{\partial \mathcal{E}(h,r,t)}{\partial \mathbf{\Sigma}_t} = \frac{1}{2}(\mathbf{\Sigma}_r^{-1} - \mathbf{\Sigma}_e^{-1}) \tag{4}$$

$$where\ \Delta'_{hrt} = \mathbf{\Sigma}_r^{-1}(\boldsymbol{\mu}_r + \boldsymbol{\mu}_t - \boldsymbol{\mu}_h), \mathbf{\Sigma}_e = \mathbf{\Sigma}_h + \mathbf{\Sigma}_t$$

We can define a symmetric similarity measure based on KL divergence as follows:

$$\mathcal{E}(h,r,t) = \frac{1}{2}(\mathcal{D}_{\mathcal{KL}}(\mathcal{P}_e,\mathcal{P}_r) + \mathcal{D}_{\mathcal{KL}}(\mathcal{P}_r,\mathcal{P}_e))$$

However, this measure lacks any gains in performance in terms of link prediction and triplet classification, likely because the discriminative ability of this formula is not distinct from the previous function for positive and negative triplets.

3.2.2 Symmetric similarity: expected likelihood

The dot product between the entity mean and relation mean is not a suitable measure of similarity because it does not integrate the covariance and cannot consider the diversity of uncertainty among different entities/relations. Therefore, we take the inner product between two distributions themselves to measure the similarity between $\mathcal{P}_e$ and $\mathcal{P}_r$.

$$\begin{aligned}\mathcal{E}(\mathcal{P}_e,\mathcal{P}_r) &= \int_{x\in\mathcal{R}^{k_e}} \mathcal{N}(x;\boldsymbol{\mu}_e,\boldsymbol{\Sigma}_e)\mathcal{N}(x;\boldsymbol{\mu}_r,\boldsymbol{\Sigma}_r)dx \\ &= \mathcal{N}(0;\boldsymbol{\mu}_e-\boldsymbol{\mu}_r,\boldsymbol{\Sigma}_e+\boldsymbol{\Sigma}_r)\end{aligned} \tag{5}$$

For better computation and comparison, we use the logarithm of the upper formula as the final energy function and denote it as EL.

$$\begin{aligned}\mathcal{E}(h,r,t) &= \log\mathcal{E}(\mathcal{P}_e,\mathcal{P}_r) = \log\mathcal{N}(0;\boldsymbol{\mu}_e-\boldsymbol{\mu}_r,\boldsymbol{\Sigma}_e+\boldsymbol{\Sigma}_r) \\ &= \frac{1}{2}\Big\{(\boldsymbol{\mu}_e-\boldsymbol{\mu}_r)^T(\boldsymbol{\Sigma}_e+\boldsymbol{\Sigma}_r)^{-1}(\boldsymbol{\mu}_e-\boldsymbol{\mu}_r)+ \\ &\qquad \log\det(\boldsymbol{\Sigma}_e+\boldsymbol{\Sigma}_r)+k_e\log(2\pi)\Big\}\end{aligned} \tag{6}$$

As in the previous case, we can compute the gradients for this energy function in a closed form.

$$\frac{\partial\mathcal{E}(h,r,t)}{\partial\boldsymbol{\mu}_h} = -\frac{\partial\mathcal{E}(h,r,t)}{\partial\boldsymbol{\mu}_r} = -\frac{\partial\mathcal{E}(h,r,t)}{\partial\boldsymbol{\mu}_t} = \Delta'_{hrt} \tag{7}$$

$$\begin{aligned}\frac{\partial\mathcal{E}(h,r,t)}{\partial\boldsymbol{\Sigma}_h} = \frac{\partial\mathcal{E}(h,r,t)}{\partial\boldsymbol{\Sigma}_r} = \frac{\partial\mathcal{E}(h,r,t)}{\partial\boldsymbol{\Sigma}_t} = \\ \frac{1}{2}(\Delta'_{hrt}\Delta'^{T}_{hrt} - \boldsymbol{\Sigma}'^{-1})\end{aligned} \tag{8}$$

$$where\ \Delta'_{hrt} = \boldsymbol{\Sigma}'^{-1}(\boldsymbol{\mu}_r+\boldsymbol{\mu}_t-\boldsymbol{\mu}_h), \boldsymbol{\Sigma}' = \boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t+\boldsymbol{\Sigma}_r$$

3.3 Learning

We define the following margin-based ranking loss for effective discrimination between observed (positive) triplets and incorrect (negative) triplets:

$$\mathcal{L} = \sum_{(h,r,t)\in\Gamma}\sum_{(h',r',t')\in\Gamma'_{(h,r,t)}} [\mathcal{E}(h,r,t)+\gamma-\mathcal{E}(h',r',t')]_+ \tag{9}$$

where $[x]_+ \triangleq \max(0,x)$ aims to obtain the maximums between 0 and x, γ is the margin separating positive and negative triplets, $\mathcal{E}(h,r,t)$ indicates the energy function formula 1 or 6 for scoring triplets, Γ is the set of positive triplets observed in the KG, and $\Gamma'_{(h,r,t)}$ denotes the set of negative triples corresponding to (h,r,t), which will be introduced below.

Under the open world assumption (OWA), existing KGs contain only correct triplets. The routine method for constructing a negative triplet (h',r',t') is to replace the head or tail entity randomly, such as sampling h' for h and obtaining (h',r,t). To obtain practical corrupted triplets, we follow [30] and assign different probabilities for head/tail entity replacement. The main idea is to provide a greater likelihood of replacing the side that will reduce the possibility of generating false-negative instances. For example, with regard to the relation *gender*, replacing *tail* is more likely to generate true-negative triplets. Following the notation in previous methods [30][16], we will denote the traditional sampling method as "unif" and the new method [30] as "bern". We also generate a negative triplet by corrupting the relation and ensure that it is not a false-negative triplet.

To avoid overfitting, we add some regularization while learning the Gaussian embedding. Considering the different geometric characteristics, we use different regularization strategies for the mean and covariance. The following hard constraints are considered when we minimize the loss $\mathcal{L}$:

$$\forall\ell\in\mathcal{E}\cup\mathcal{R}, ||\boldsymbol{\mu}_\ell||_2 \le 1 \tag{10}$$

$$\forall\ell\in\mathcal{E}\cup\mathcal{R}, c_{min}I \le \boldsymbol{\Sigma}_\ell \le c_{max}I, c_{min}>0 \tag{11}$$

where the constraint 10 ensures that the means remain sufficiently small and the constraint 11 guarantees that the covariance matrices are positive definite and of appropriate size. We can use $\boldsymbol{\Sigma}_{ii} \leftarrow max(c_{min}, min(c_{max}, \boldsymbol{\Sigma}_{ii}))$ to achieve these goals for diagonal covariance.

Algorithm 1: THE LEARNING ALGORITHM OF **KG2E**

Input: An energy function $\mathcal{E}(h,r,t)$, training set $\Gamma=\{(h,r,t)\}$, entity set $\mathcal{E}$ and relation set $\mathcal{R}$, entity and relation sharing embedding dimensions k, margin γ, restriction values c_{min} and c_{max} for covariance, learning rate α and maximum epochs n.

Output: All the Gaussian embeddings (mean vector and covariance matrix) of e and r, where $e\in\mathcal{E}$ and $r\in\mathcal{R}$.

foreach $\ell\in\mathcal{E}\cup\mathcal{R}$ **do**
  ℓ.mean $\leftarrow$ Uniform($\frac{-6}{\sqrt{k}}$, $\frac{6}{\sqrt{k}}$)
  ℓ.cov $\leftarrow$ Uniform(c_{min}, c_{max})
  regularize ℓ.mean and ℓ.cov with constraints 10 and 11
$i \leftarrow 0$
while $i++ \le n$ **do**
  $\Gamma_{batch} \leftarrow$ sample(Γ, b) //sample a minibatch of size B from Γ
  $T_{batch} \leftarrow \emptyset$ //pairs of triplets for learning
  foreach $(h,r,t)\in\Gamma_{batch}$ **do**
    $(h',r,t') \leftarrow$ negSample((h,r,t)) //sampling negative triplet with "unif" or "bern"
    $T_{batch} \leftarrow T_{batch}\cup((h,r,t),(h',r,t'))$
    $(h,r',t) \leftarrow$ negSample((h,r,t)) //sampling negative triplet by corrupting relation
    $T_{batch} \leftarrow T_{batch}\cup((h,r,t),(h,r',t))$
  Update Gaussian embeddings based on Equations 2, 3 and 4 (or 7 and 8, depending on $\mathcal{E}(h,r,t)$) w.r.t. $\mathcal{L} = \sum_{((h,r,t),(h',r',t'))\in T_{batch}}[\mathcal{E}(h,r,t)+\gamma-\mathcal{E}(h',r',t')]_+$
  regularize the means and covariances for each entity and relation in T_{batch} with constraints 10 and 11

We use *Stochastic Gradient Descent* (SGD)[8] in small mini-batches to iteratively update the Gaussian embeddings of entities and relations. In our model, we must first choose an energy function (EL or KL) (hereafter, we denote the corresponding models as KG2E_EL

[8]We also use AdaGrad [9] to optimize the parameters but found no improvement.

and KG2E_KL, respectively.). The detailed learning procedure is described in Algorithm 1. All Gaussian embeddings for entities and relations are first initialized randomly following a uniform distribution. At each main iteration of the algorithm, we first sample a batch of observed triplets, and construct corresponding negative triplets based on the aforementioned sampling methods ("unif" or "bern"). The parameters of Gaussian embedding are then updated by taking a gradient step (using formulas 2, 3 and 4 or 7 and 8, depending on the choice of energy function $\mathcal{E}(h, r, t)$), with a constant learning rate. For each step (including the initial embedding), we ensure that all embeddings satisfy with the constraints 10 and 11.

4. EXPERIMENTS

4.1 Data Sets

In this work, we empirically study and evaluate related methods for two tasks: link prediction [4] and triplet classification [25]. We use datasets commonly used in previous methods, which are built from two typical KGs: WordNet [18] and Freebase [1]. WordNet is a lexical database of the English language. In WordNet, each entity represents a synset consisting of several words, and a word can also belong to different synsets. Relationships between synsets include *hypernym*, *hyponym*, *meronym*, *holonym*, *troponym* and other lexical relations. We adopt two datasets from WordNet, WN18, used in [4] for link prediction, and WN11, used in [25] for triplet classification. Among them, WN18 contains 18 relations and WN11 contains 11. Freebase is a large collaborative knowledge graph of general world facts. For example, the triplet (*Bill Gates, place_of_birth, Seattle*) indicates that the person with entity *Bill Gates* was born in (*place_of_birth*) the location with entity *Seattle*. We adopt two datasets from Freebase, FB13, used in [25] for triplet classification, and FB15k, used in [4] for link predication and triplet classification. Among them, FB13 contains 13 relations and FB15k contains approximately 15,000 entities. The statistics of these datasets are listed in Table 2.

Dataset	$\#\mathcal{R}$	$\#\mathcal{E}$	#Triplet (Train/Valid/Test)		
WN18	18	40,943	141,442	5,000	5,000
FB15k	1,345	14,951	483,142	50,000	59,071
WN11	11	38,696	112,581	2,609	10,544
FB13	13	75,043	316,232	5,908	23,733

Table 2: Datasets used in the experiments.

4.2 Qualitative Analysis

Before evaluation in each specific task and comparison with other methods, we first examine the effectiveness and ability of our proposed method to represent the uncertainty in a KG with a qualitative analysis. The following surveys and observations are based on the representations of embeddings learned by KG2E and using KL-divergence as a similarity measure in FB15k.

First, we want to know the effect of covariance in modeling the uncertainty in a KG. Based on our ideas, an entity/relation with a higher level of uncertainty has a larger covariance (corresponding with determinant or trace). Considering the uncertainty of an entity, we focus on the relationship between the (log) determinant of covariance matrix and its density. The entity/relation density is indicated by the number of corresponding triplets, and the entity density is measured at different positions: head part, tail part or entire set. As shown in Figure 3, there is a clear tendency for the larger determinant of entity covariance to have fewer corresponding triplets, regardless of position. Considering the uncertainty of a relation, we measure the (log) determinant and trace of covariance for 13 relations with */people/person* as *domain*, as shown in Table 3 (each row includes the following information with a relation: label, number of triplets,number of head entities, number of tail entities, type[9], (log) determinant and trace of covariance matrix). We can draw the following conclusions: 1) the covariance of Gaussian embedding can effectively model the (un)certainty of a relation; 2) relations with complex semantic (e.g., many_to_one (m-1) and many_to_many (m-n) relations) have larger uncertainty, and 3) the more unbalanced the head and tail entities, the larger the uncertainty. For example, the *nationality* relation has the largest uncertainty, and the *parents* relation has the smallest uncertainty among these 13 relations.

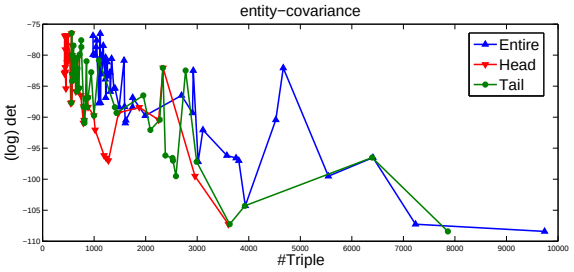

Figure 3: The relationships between the density (linked triplet number) of entity and the determinant of its corresponding covariance.

Relaion	#Triplet	#Head	#Tail	Type	(log) det	trace
nationality	4198	3755	100	m-1	-4.77	67.90
place_lived	3740	2441	784	m-n	-23.02	53.24
profession	11636	4145	152	m-n	-57.45	23.10
gender	3721	3721	2	m-1	-59.53	21.35
place_of_birth	2468	2468	685	m-1	-63.42	19.41
ethnicity	2030	1610	78	m-1	-69.95	15.00
major	260	217	60	m-1	-69.59	14.62
sibling	131	111	113	1-1	-72.98	14.29
religion	1086	963	45	m-1	-75.07	12.98
spouse	427	395	385	1-1	-77.77	12.24
children	77	69	71	1-1	-76.94	12.14
parents	83	74	76	1-1	-77.20	12.03

Table 3: The relationships between some relations and the determinants and traces of their corresponding covariances, sorted by the descending order of trace.

Next, we want to know the ability of Gaussian embedding to learn valid entity/relation representations. Tables 4 and 5 give the top 5 similarity entities/relations with regard to some sampling examples. The tables illustrate that the proposed method can learn a valid representation for modeling KGs.

4.3 Link Prediction

Following the usage in [5][4], link prediction aims to predict the missing h or t for a relation fact triplet(h, r, t). Instead of obtaining one best answer, this task puts more emphasis on ranking a set of candidate entities from the KG. Similar to the setting in [5][4][30][16], we conduct initial experiments using the datasets WN18 and FB15k.

Evaluation protocol. We follow the same protocol as in TransE [4] and its variants [30][16]: In the testing phrase, for each test triplet (h, r, t), we replace the tail entity by all entities e in the

[9]We follow the definition in [4] and measure used in [30].

GNU/Linux	ESPN
Unix	Philips
Solaris Operating System	CNN
Android	AOL
Windows Vista	Parlophone
Project management	64th Primetime Emmy Awards
University of Sydney	**Java (island)**
University of Leeds	Central Java
Seoul National University	East Java
Commonwealth of Nations	West Java
Princeton University	Bandung
Carleton University	Javanese people

Table 4: The top-5 similarity entities with regards to some examples.

people/person/nationality	location/location/contains
location/*division/country	*/country/*divisions
location/hud_county_place/place	location/location/containedby
*/educational/*institution	location/*division/country
base/*/bibs_location/country	biblioness/*location/state
music/artist/origin	*area/capital
film/film/produced_by	***/organization/founders**
film/film/directed_by	*/organizations_founded
film/*/executive_produced_by	people/person/place_of_birth
film/film/written_by	*/location/people_born_here
film/*/performance/actor	*/employer*/person
film/film/story_by	*/award_winner/*/ceremony

Table 5: The top-5 similarity relations with regards to some examples, using a wildcard $*$ to reduce space occupation without ambiguous expression.

KG and rank these entities in descending order of similarity scores, measured by the energy function $\mathcal{E}(h, r, e)$. A similar process is performed for the head entity measure by $\mathcal{E}(e, r, t)$. Based on these entity ranking lists, we use two evaluation metrics by aggregation over all the testing triplets: 1) the average rank of correct entities (denoted as *Mean Rank*) and 2) the proportion of correct entities in the top 10 ranked entities (denoted as *Hits@10*). A good method should obtain lower *Mean Rank* or higher *Hits@10*. Considering the fact that a corrupted triplet for (h, r, t) also exists in a KG, such a prediction should also be deemed correct. However, the above evaluations do not consider the issue and may underestimate the metrics. To eliminate this factor, we remove those corrupted triplets that already appeared in training, valid or testing sets before obtaining the rank entity list of each testing triplet. We term the former evaluation setting as "Raw" and the latter setting as "Filter".

Implementation. Because the testing datasets are the same, we directly compare our models with several baselines reported in [4][30][16]. In learning KG2E, we select the learning rate α for SGD among {0.001, 0.01, 0.05}, the margin γ among {1, 2, 4}, the dimensions of entity and relation sharing embedding k among {20, 50, 100}, the batch size B among {20, 120, 1440, 2480}, and the pair of restriction values c_{min} and c_{max} for covariance among {(0.01, 1), (0.03, 3), (0.05, 5)}. The optimal configuration is determined by the *Hits@10* in the validation set. As the strategy of constructing negative labels can greatly influence the evaluations, we use different parameters for "unif" and "bern". We also use different parameters for KG2E_KL and KG2E_EL. The default configuration for all experiments is as follows: $\alpha = 0.001$, $\gamma = 1$, $k = 50$, $B = 120$, and (c_{min}, c_{max})= (0.05, 5). Below, we list only the non-default parameters. For KG2E_KL, under the "unif" setting, the optimal configuration is as follows: $\alpha = 0.01$, and $\gamma = 4$ on WN18 and $B = 1440$ on FB15k. Under the "bern" setting, the optimal configuration is as follows: $\alpha = 0.01$, $\gamma = 4$, $B = 20$, and (c_{min},c_{max})= (0.03,3) on WN18 and $B = 2480$ on FB15k. For KG2E_EL,under the "unif" setting, the optimal configuration is as follows: $\alpha = 0.01$, $\gamma = 4$, and $B = 20$ on WN18 and $\gamma = 2$ on FB15k. Under the "bern" setting, the optimal configuration is as follows: $\alpha = 0.01$, $\gamma = 4$, $B = 20$, and (c_{min},c_{max})= (0.03,3) on WN18 and $B = 2480$ on FB15k. For both datasets, we traverse all the training triplets for 500 rounds.

Results. The results are reported in Table 6. On WN18, TransE, TransH, TransR, KG2E and even the naive baseline unstructured models outperform other approaches in terms of the *Mean Rank* metric, but the majority of models are poor in terms of the *Hits@10* metric. KG2E with the KL energy function outperforms other baseline models, including TransE, TransH and TransR, in terms of the *Hits@10* metric but achieves a worse *Mean Rank*. One reason may be that WN18 simply contains a small number of relations, and thus, simple methods can judge the correct triplet but cannot rank it in the top position. Another reason is that the *Mean Rank* is easily reduced by an obstinate triplet with a low rank. On FB15k, KG2E_KL consistently outperforms the other baseline models in both *Mean Rank* and *Hits@10*. As the *density* diversity in FB15k is greater than that in WN18, we hypothesize that the improvements are because the *uncertainty* diversities in FB15k are greater than those in WN18, and thus, the *density-based* embedding methods can handle it better. From the observations, we can draw the following conclusions: 1) Gaussian embedding can learn valid representations of KGs for link prediction. 2) KG2E is superior to other baseline methods. 3) KG2E_KL performs better than KG2E_EL, which indicates that the asymmetric energy function is more suitable for learning the representation of KGs with Gaussian embedding. 4) The "bern" sampling strategy works well for most approaches, especially on FB15k, which has many more relation types.

Table 7 shows the evaluation results with separated types of relation properties. Following [4], we divide relations into four types: one-to-one, one-to-many, many-to-one and many-to-many, for which the proportions in FB15k (1345 relations in total) are 24%, 23%, 29% and 24%, respectively, based on the measure used in [30]. KG2E_KL and KG2E_EL significantly outperform TransE, TransH, TransR and other baseline methods in one-to-one, one-to-many, and many-to-one relations. For the difficult tasks of predicting tails in one-to-many relations and predicting heads in many-to-one relations, KG2E_KL obtains 29.3% and 29.9% improvements, respectively. However, the proposed method presents only a slight advantage for many-to-many relations, possibly because there are various fine-grained types within a many-to-many relation that cannot be effectively expressed by one Gaussian embedding. We believe CTransR, proposed by [16], is an effective way to handle this issue by adopting a clustering strategy to divide entity pairs into different sub-types. To better review the modeling improvement of *uncertainty* of KG2E over TransE and its variants, Table 8 shows the *Hits@10* results on some typical one-to-many, many-to-one, many-to-many and reflexive relations. We directly copy the experimental results of TransH from [30] for a fair comparison. We can observe that the asymmetric similarity measure can effectively handle reflexive relations.

4.4 Triplet Classification

This task seeks to judge whether a given triplet (h, r, t) is correct or not. That is, it is a binary classification task for fact triplets, which was first explored in [25] and then widely used to evaluate KGs embedding [30] [16]. In this task, we use three datasets: WN11, FB13 and FB15k.

Evaluation protocol. We follow the same protocol as in NTN [25]. The evaluation of binary classification requires negative triplets.

Data sets	WN18				FB15K			
Metric	Mean Rank		Hits@10		Mean Rank		Hits@10	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
Unstructured (Bordes et al. 2012)	315	304	35.3	38.2	1,074	979	4.5	6.3
RESCAL (Nickle, Tresp, and Kriegel 2011)	1,180	1,163	37.2	52.8	828	683	28.4	44.1
SE (Bordes et al. 2011)	1,011	985	68.5	80.5	273	162	28.8	39.8
SME (linear) (Bordes et al.2012)	545	533	65.1	74.1	274	154	30.7	40.8
SME (bilinear) (Bordes et al. 2012)	526	509	54.7	61.3	284	158	31.3	41.3
LFM (Jenatton et al. 2012)	469	456	71.4	81.6	283	164	26.0	33.1
TransE (Bordes et al. 2013)	263	251	75.4	89.2	243	125	34.9	47.1
TransH (unif) (Wang et al. 2014)	318	303	75.4	86.7	211	84	42.5	58.5
TransH (bern) (Wang et al. 2014)	401	388	73.0	82.3	212	87	45.7	64.4
TransR (unif) (Lin et al. 2015)	**232**	**219**	78.3	91.7	226	78	43.8	65.5
TransR (bern) (Lin et al. 2015)	238	225	79.8	92.0	198	77	48.2	68.7
CTransR (unif) (Lin et al. 2015)	243	230	78.9	92.3	233	82	44.0	66.3
CTransR (bern) (Lin et al. 2015)	**231**	**218**	79.4	92.3	199	75	**48.4**	70.2
KG2E_EL (unif)	381	369	74.8	87.8	217	94	38.5	58.6
KG2E_EL (bern)	385	373	74.1	85.0	219	112	39.4	56.7
KG2E_KL (unif)	362	348	**80.5**	**93.2**	**183**	**69**	47.5	**71.5**
KG2E_KL (bern)	342	331	**80.2**	**92.8**	**174**	**59**	**48.9**	**74.0**

Table 6: Experimental results on link prediction.

Tasks	Prediction Head (Hits@10)				Prediction Tail (Hits@10)			
Relation Category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
Unstructured (Bordes et al. 2012)	34.5	2.5	6.1	6.6	34.3	4.2	1.9	6.6
SE (Bordes et al. 2011)	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME (linear) (Bordes et al.2012)	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME (bilinear) (Bordes et al. 2012)	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE (Bordes et al. 2013)	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH (unif) (Wang et al. 2014)	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransH (bern) (Wang et al. 2014)	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR (unif) (Lin et al. 2015)	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
TransR (bern) (Lin et al. 2015)	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
CTransR (unif) (Lin et al. 2015)	78.6	77.8	36.4	68.0	77.4	37.8	78.0	70.3
CTransR (bern) (Lin et al. 2015)	81.5	89.0	34.7	**71.2**	80.8	38.6	90.1	**73.8**
KG2E_EL (unif)	49.2	87.9	55.2	53.1	47.8	54.9	88.4	56.2
KG2E_EL (bern)	51.8	89.4	47.9	51.6	52.0	43.6	89.3	55.0
KG2E_KL (unif)	**92.2**	**93.7**	**67.4**	69.1	**91.2**	**69.7**	**93.6**	71.2
KG2E_KL (bern)	**92.3**	**94.6**	**66.0**	**69.6**	**92.6**	**67.9**	**94.4**	**73.4**

Table 7: Experimental results on FB15k by mapping properties of relations. (%)

Relation	Hits@10 (TransH / KG2E_KL)	
	Predict Head	Predict Tail
football_position/players*	100 / 100	22.2 / **100**
production_company/films*	85.6 / **97.3**	16.0 / **29.4**
director/film*	89.6 / **93.4**	80.2 / **85.8**
disease/treatments†	66.6 / 66.6	100 / 100
person/place_of_birth†	**37.5** / 34.1	**87.6** / 84.6
film/production_companies†	21.0 / **44.2**	87.8 / **97.8**
field_of_study/students_majoring‡	66.0 / **86.8**	62.3 / **81.1**
award_winner/awards_won‡	87.5 / **88.4**	86.6 / **89.2**
sports_position/players‡	100 / 100	86.2 / **100**
person/sibling§	63.2 / **89.5**	36.8 / **94.7**
person/spouse§	35.2 / **77.8**	42.6 / **85.2**

Table 8: Hits@10 (Filter) of KG2E_KL and TransH on some examples of one-to-many*, many-to-one†, many-to-many‡, and reflexive relations§.

WN11 and FB13 released by NTN [25] already contain negative triplets, which are built by corrupting the corresponding positive (observed) triplets. As FB15k has not released negative triplets in previous works, we construct negative triplets following the same procedure used in [25] for FB13. The setting for triplet classification is very simple: for each triplet (h, r, t), if the dissimilarity score obtained by the energy function $\mathcal{E}(h, r, t)$ is below a relation-specific threshold δ_r, then the triplet will be classified as positive. Otherwise, it will be classified as negative. The relation-specific threshold δ_r is optimized by maximizing the classification accuracy on the validation set.

Implementation. Considering that the same datasets (negative triplets) are used in WN11 and FB13, we directly compare our models with the baseline methods reported in [16]. For evaluation on FB15k, we use the code released by Lin[10] [16] (running TransE, TransH and TransR) and Socher[11] [25] (running NTN). For TransE, TransH and TransR, we select the learning rate α for SGD among {0.001, 0.01, 0.05}, the margin γ among {0.5, 1, 2}, the dimensions of entity and relation sharing embedding k among {20, 50, 100}, and the batch size B among {20, 120, 1440, 2480}. Other parameters follow the default configuration in the shared codes. For the NTN, we did not change the settings: dimension $k = 100$, and the number of slices equals 3. The optimal configurations are as follows: $\alpha = 0.001$, $\gamma = 1$, and $B = 4800$ for TransE (bern); $\alpha = 0.001$, $\gamma = 2$, and $B = 120$ for TransE (unif); $\alpha = 0.001$, $\gamma = 0.5$, and $B = 4800$ for TransH (bern) $\alpha = 0.01$, $\gamma = 0.5$, and $B = 4800$ for TransH (unif); $\alpha = 0.001$, $\gamma = 1$, and $B = 4800$ for TransR (bern); and $\alpha = 0.001$, $\gamma = 1$, and $B = 120$ for TransR (unif). The dimension $k = 100$ for all the above configurations.

In learning KG2E, we select the learning rate α for SGD among {0.001, 0.01, 0.05}, the margin γ among {1, 1.5, 2}, the dimen-

[10] https://github.com/mrlyk423/relation_extraction

[11] www.socher.org

sions of entity and relation sharing embedding k among {20, 50, 100}, the bach size B among {20, 120, 1440, 2480}, and the pair of restriction values c_{min} and c_{max} for covariance among {(0.01, 1), (0.03, 3), (0.05, 5)}. The optimal configuration is determined by the classification accuracy in the validation set. For all three datasets, we traverse all the training triplets for 1000 rounds. We also use different parameters for KG2E_KL and KG2E_EL. The default configuration for all experiments are as follows: $\alpha = 0.001$, $\gamma = 1$, $k = 50$, $B = 120$, and (c_{min}, c_{max})= (0.05, 5). Below, we list only the non-default parameters. For KG2E_KL, under the "unif" setting, the optimal configuration is as follows: $k = 20$, $\gamma = 2$, and $B = 120$ on WN11; $k = 100$, and $B = 1440$ on FB13. Under the "bern" setting, the optimal configuration is as follows: $k = 20$, and $\gamma = 2$ on WN11 and $k = 100$, and $B = 1440$ on FB13. For KG2E_EL, under the "unif" setting, the optimal configuration is as follows: $k = 20$, $\gamma = 2$, and $B = 120$ on WN11; $k = 100$, and $B = 120$ on FB13 and $\gamma = 1.5$, on FB15k; Under the "bern" setting, the optimal configuration is as follows: $k = 20$, $\gamma = 2$, and $B = 120$ on WN11; $k = 100$ on FB13 and $B = 2480$ on FB15k.

Results. The accuracy of triplet classification on the three datasets is shown in Table 9. On WN11, KG2E_KL and TransR outperform all the other models. The NTN, the powerful model with the most parameters, outperforms the other approaches on FB13, but it performs poorly on FB15k, which contains many more relations. In contrast, on a more practical KG with large-scale relations (such as Freebase), the proposed method KG2E_KL performs much better than the other baseline models, and even the KG2E_EL is also a competitive model. We can draw the following conclusions from the observations: 1) KG2E_KL achieves superior performance compared to other baseline methods for a multi-relational KG, which indicates that Gaussian embedding can effectively model the enormous diversity of uncertainty in a KG. 2) KG2E_KL performs better than KG2E_EL, which is consistent with the results of link prediction. 3) The "bern" sampling strategy outperforms the majority of the approaches (including TransE, TransH, TransR and our proposed models KG2E_EL and KG2E_KG) on all three datasets.

We also compare with CTransR, another model proposed by Lin [16] to handle many-to-many relations with entity-pairs clustering. However, in the NTN [25], another set of results combined with word embedding [17] is reported. There are different ways to improve KG embedding between the aforementioned method and our method, and for fairness, we have not compared their results.

As shown in [30] and [16], the training times of TransE, TransH and TransR are approximately 5 minutes, 30 minutes and 3 hours, respectively. The computational complexities of our proposed methods are lower than that of TransR but higher than those of both TransE and TransH: KG2E_KL and KG2E_EL take approximately 80 and 75 minutes for training, respectively.

Data sets	WN11	FB13	FB15k
SE (Bordes et al. 2011)	53.0	75.2	-
SME (bilinear) (Bordes et al.2012)	70.0	63.7	-
SLM (Socher et al. 2013)	69.9	**85.3**	-
LFM (Jenatton et al. 2012)	73.8	84.3	-
NTN (Socher et al. 2013)	70.4	**87.1**	68.2
TransE (unif) (Bordes et al. 2013)	75.9	70.9	79.2
TransE (bern) (Bordes et al. 2013)	75.9	81.5	81.4
TransH (unif) (Wang et al. 2014)	77.7	76.5	85.4
TransH (bern) (Wang et al. 2014)	78.8	83.3	85.8
TransR (unif) (Lin et al. 2015)	85.5	74.7	83.7
TransR (bern) (Lin et al. 2015)	**85.9**	82.5	86.5
CTransR (bern) (Lin et al. 2015)	**85.7**	-	87.4
KG2E_EL (unif)	73.8	76.3	83.9
KG2E_EL (bern)	75.2	85.2	84.9
KG2E_KL (unif)	83.6	76.4	**88.6**
KG2E_KL (bern)	85.4	**85.3**	**89.3**

Table 9: Experimental results of Triplet Classification (%).

5. CONCLUSIONS

In this paper we propose KG2E, a new method for learning representations of entities and relations in KGs with Gaussian embedding. Each entity and relation is represented by a Gaussian distribution with a mean vector and a covariance matrix (currently with diagonal covariance for computational efficiency), which aims to model the uncertainty of entities and relations in a KG. The (un)certainties vary considerably for different entities and relations: for example, popular entities with fewer uncertainty, which contain more relations and facts than unpopular ones, high-frequency relations with more uncertainty, which link more entity pairs than low frequency ones, and, different parts of relations can contain a very different numbers of entities that sharpen the uncertainties of relations. We use two energy functions (symmetric and asymmetric) to compute the score of a triplet fact. Extensive experiments on link prediction and triplet classification with multiple benchmark datasets (including WordNet and Freebase) demonstrate that the proposed method significantly outperforms state-of-the-art methods.

In the future, we plan to address the following limitations that still exist in the methods:

- Existing methods do not explicitly consider the relationships between relations. For example, triplets with the relations *children* and *parents* (normally, *(X, children, Y)* and *(Y, parents, X)* are mutual implications) affect and restrict with each other. We will explore mixing logical rules into the Gaussian embedding. The same concepts have been successfully used in relation extraction [22].

- Various KGs can supply and verify with each other. For example, NELL contains more *machine learning* related relations (e.g., *mlauthor*) than Freebase but fewer person related relations (e.g., *spouse*), and NELL may extract a false triplet fact (e.g., *(Erma Bombeck, was born in, Dayton)*) with high confidence that can be rectified by Freebase. We will explore the fusion of multiple KGs, learning the representations in a joint model with or without entity linking.

- Compared with other types of relations, KG2E does not significantly outperform previous methods in many-to-many relations, possibly because current entity embedding does not consider the types of entity. For example, even the relation *contains* incorporates many fine-grained relations, such as, *country_contain_university*, *country_contain_city* and *continent_contain_country*. We can ascertain the semantics of relations for the head entity of type *country* and the tail entity of type *city*.

Acknowledgments

The authors are grateful to anonymous reviewers for their constructive comments. This work was supported by the National High Technology Development 863 Program of China (No. 2015AA015405) and the National Natural Science Foundation of China (No. 61272332 and No. 61202329).

6. REFERENCES

[1] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[2] A. Bordes, S. Chopra, and J. Weston. Question answering with subgraph embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 615–620, 2014.

[3] A. Bordes, X. Glorot, J. Weston, and Y. Bengio. A semantic matching energy function for learning with multi-relational data. volume 94, pages 233–259. Springer, 2012.

[4] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

[5] A. Bordes, J. Weston, R. Collobert, Y. Bengio, et al. Learning structured embeddings of knowledge bases. In *AAAI*, 2011.

[6] R. Brachman and H. Levesque. *Knowledge representation and reasoning*. Elsevier, 2004.

[7] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, 2010.

[8] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov. Clustering with multi-layer graphs: A spectral perspective. *Signal Processing, IEEE Transactions on*, 60(11):5820–5831, 2012.

[9] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[10] F. Hayes-Roth, D. Waterman, and D. Lenat. Building expert systems. 1984.

[11] S. He, K. Liu, Y. Zhang, L. Xu, and J. Zhao. Question answering over linked data using first-order logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1092–1103, 2014.

[12] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *The Journal of Machine Learning Research*, 5:819–844, 2004.

[13] R. Jenatton, N. L. Roux, A. Bordes, and G. R. Obozinski. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems*, pages 3167–3175, 2012.

[14] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[15] N. Lao, T. Mitchell, and W. W. Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539. Association for Computational Linguistics, 2011.

[16] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2015.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[18] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[19] M. Nickel, V. Tresp, and H.-P. Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 809–816, 2011.

[20] A. Paccanaro and G. E. Hinton. Learning distributed representations of concepts using linear relational embedding. volume 13, pages 232–244. IEEE, 2001.

[21] K. B. Petersen, M. S. Pedersen, et al. *The matrix cookbook*, volume 450. 2008.

[22] T. Rocktäschel, S. Singh, and S. Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the 2015 Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 2015.

[23] H. Schuetze and C. Scheible. Two svds produce more focal deep learning representations. *arXiv preprint arXiv:1301.3627*, 2013.

[24] W. Shen, J. Wang, P. Luo, and M. Wang. Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 68–76. ACM, 2013.

[25] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.

[26] I. Sutskever, J. B. Tenenbaum, and R. R. Salakhutdinov. Modelling relational data using bayesian clustered tensor factorization. In *Advances in neural information processing systems*, pages 1821–1828, 2009.

[27] S. Szumlanski and F. Gomez. Automatically acquiring a semantic network of related concepts. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 19–28. ACM, 2010.

[28] L. Vilnis and A. McCallum. Word representations via gaussian embedding. In *Proceedings of the International Conference on Learning Representations (ICLR 2015)*, 2015.

[29] W. Y. Wang, K. Mazaitis, N. Lao, and W. W. Cohen. Efficient inference and learning in a large knowledge base. *Machine Learning*, pages 1–26, 2015.

[30] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119, 2014.

[31] M. Yahya, K. Berberich, S. Elbassuoni, and G. Weikum. Robust question answering over the web of linked data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1107–1116. ACM, 2013.

[32] L. Yao, S. Riedel, and A. McCallum. Probabilistic databases of universal schema. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 116–121. Association for Computational Linguistics, 2012.