TransMVG: Knowledge Graph Embedding Based on Multiple-Valued Gates

Xiaobo Guo[1,2], Neng Gao[1], Jun Yuan[3], Xin Wang[1(✉)], Lei Wang[1], and Di Kang[4]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{guoxiaobo,gaoneng,wangxin,wanglei}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] College of Traffic Engineering, Hunan University of Technology, Zhuzhou, China
yuanjun@iie.ac.cn
[4] National Secrecy Science and Technology Evaluation Center, Beijing, China
nsstec_kangd@163.com

Abstract. The essence of knowledge representation learning is to embed the knowledge graph into a low-dimensional vector space to make knowledge computable and inferable. Semantic discriminate models greatly improve the performance of knowledge embedding through increasingly complex feature engineering. For example, the projection calculation based on matrixes can achieve more detailed semantic interactions and higher accuracies. However, complex feature engineering results in high time complexity and discriminate parameters pressure, which make them difficult to effectively applied to large-scale knowledge graphs. TransGate is proposed to relieve the pressure of the huge number of parameters in semantic discriminate models and obtains better performance with much fewer parameters. We find that the gate filtering vector obtained by the traditional gate used by TransGate would rapidly fall in the state of a nearly boundary binary-valued distribution (most values are near 0 or near 1) after only a few hundred rounds of training. This means that most filtering gate values either allow the information element to pass completely or not at all, which can be called extreme filtering. We argue that this filtering pattern ignore the interaction between information elements. In this paper, TransMVG model is proposed to improve the traditional boundary binary-valued gate to a multiple-valued gate on the premise of ensuring the randomness. The experiments results show that TransMVG outperforms the state-of art baselines. This means it is feasible and necessary to multivalue the filter gate vectors in the process of knowledge representation learning based-on the gate structure.

Keywords: Knowledge representation learning · Boundary binary-valued gate · Multiple-valued gate.

Z. Huang et al. (Eds.): WISE 2020, LNCS 12342, pp. 286–301, 2020.
https://doi.org/10.1007/978-3-030-62005-9_21

1 Introduction

Nowadays, knowledge graph has become an important resource to support AI related applications, including relation extraction, question answering, semantic search and so on. Generally, a knowledge graph is a set of facts, usually represented as a triplet $(head, relation, tail)$, denoted as (h, r, t). Although the current knowledge graph, such as WordNet [14] and Freebase [1], has a large amount of data, it is far from perfect. For example, according to Google, 71% of the people in Freebase lack birthplace records and 75% lack nationality records. The lack of completeness of knowledge graph seriously affects the downstream applications.

Semantic indiscriminate models assume that the vector representation of entities and relations in any condition should be the same, regardless of the importance of semantic environments. Semantic discriminate models are proposed to distinguish multiple semantics. TransH [22] embeds entities in the relation hyperplane to distinguish relation-specific information. TransR [13] learns a mapping matrix for each relation and map each entity into the relation space respectively. TranSparse [11] replaces the mapping matrix in the TransR with two sparse matrices for each relation. TransD [10] dynamically constructs two mapping matrices for each triplet by setting projection vectors for each entity and relation. TransG [23] models the different semantics of the same relation into different Gaussian distributions and assumes that all semantics of a relation contribute to the fractional function of the fact triples. These models have greatly improved accuracies through increasingly complex feature engineering, but the problem of large number of parameters and high computation complexity also come. Despite their high accuracies, it is difficult to apply these models to large-scale real knowledge graphs.

The primary cause of the large number of parameters in the semantic discriminated models is that they do not pay attention to the intrinsic correlation between relations, and by default they assume relations are independent. As a result, these models have to learn a set of parameters for each relation for a relation-specific semantic discrimination. TransGate [25] is proposed to relieve the pressure of the huge number of parameters in semantic discriminate models with two fixed-size shared parameter gates based on the traditional Long Short Term Memory (LSTM) [26] gate by utilizing the inherent correlation between relations. It obtains better performance with much fewer parameters. Unlike relation-specific matrices in other models, the two learned global shared parameter gates of TransGate [25] will not grow with the expansion of the data set and makes it much easier to be applied to the large-scale real knowledge graphs.

We run TransGate [25] on FB15K [2] multiple times with different parameter configurations and find that the gate filtering vector obtained by the traditional gate rapidly falls in the state of a nearly boundary binary-valued distribution (most values are near 0 or near 1) after only a few hundred rounds of training. It means that most filtering gate values either allow the information element to pass completely or not at all. This extreme filtering pattern is shown in Fig. 1 (The parameter configuration is the same with Fig. 4). We argue that it ignores the interaction between information elements. We believe that the gate filtering

vector should choose how much information elements to pass through rather than whether to pass through. In different semantic environments, the proportion combinations of information elements allowed to pass by the gate filtering vector are different. This is the fundamental reason why the same vector can express multiple semantics. Inspired by [12], TransMVG proposed in this paper not only inherites the advantages of global parameter sharing of TransGate [25], but also makes the gate filtering vector have stronger information selection ability.

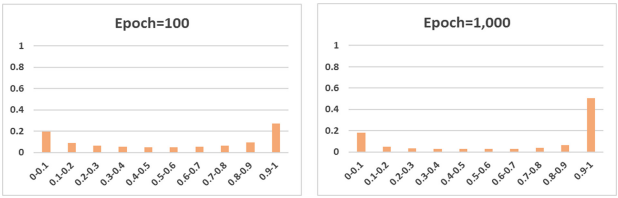

Fig. 1. The extreme filtering of TransGate

Our main contributions are as the following:

Contribution I: We find that extreme filtering phenomenon exists in the gate filtering vectors obtained by the traditional shared gate method. We propose that the gate filtering vector should choose how much information elements to pass through instead of whether to pass through, so as to obtain richer semantic interaction and more refined semantic expression.

Contribution II: The proposed TransMVG model improves the traditional boundary binary-valued gate to the multiple-valued gate, which means all values between 0 and 1 can be randomly selected as a filtering gate value, not mostly 0 and 1. In this way, more different proportion combinations of information elements allowed to pass by the gate filtering vector make the learned semantics more precise and clear.

Contribution III: Experiments show that TransMVG obtains some significant improvement compared to most state-of-art baselines with fewer parameters. This indicates that it is feasible and necessary to filter information elements precisely by using the multiple-valued gate.

2 Related Work

2.1 Semantic Indiscriminate Models

Indiscriminate models usually focus more on the scalability on real-world knowledge graphs. They assume that the vector representations of entities and relations are consistent in any semantic environment. As a result, they often have low accuracies:

TransE [2] represents a relation as a translation vector r indicating the semantic translation from the head entity h to the tail entity t, so that the pair of embedded entities in a triplet (h, r, t) can be connected by r with low error. The score function is $f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} + \mathbf{h} - \mathbf{t}\|_2^2$. It is very efficient, but only suitable for $1-to-1$ relations, and has flaws in dealing with $1-to-N$, $N-to-1$ and $N-to-N$ relations.

DistMult [24] has the same time and space complexity as TransE. It uses weighted element-wise dot product to define the score function $f_r(h,t) = \sum h_k r_k t_k$. Although DistMult has better overall performance than TransE, but it is unable to model asymmetric relations.

ComplEx [21] makes use of complex valued embeddings and Hermitian dot product to address the antisymmetric problem in DistMult. However, TransE and DistMult perform better on symmetry relations than ComplEx.

CombinE [19] considers triplets features from two aspects: relation $\mathbf{r}_p \approx \mathbf{h}_p + \mathbf{t}_p$ and entity $\mathbf{r}_m \approx \mathbf{h}_m - \mathbf{t}_m$. The score function is $f_r(h,t) = \|\mathbf{h}_p + \mathbf{t}_p - \mathbf{r}_p\|_{L1/L2}^2 + \|\mathbf{h}_m - \mathbf{t}_m - \mathbf{r}_m\|_{L1/L2}^2$. CombinE doubles the parameter size of TransE, but does not yield significant boost in performance.

2.2 Semantic Discriminate Models

Discriminate models focus more on precision. They assume that the vector representations should depend on the specific semantic environment. They usually contain two stages: relation-specific information discrimination and score computation.

TransH [22] is proposed to enable an entity to have distinct distributed representations when involved in different relations. TransH projects the entity embeddings to the hyperplane with a certain norm vector. By projecting entity embeddings into relation hyperplanes, it allows entities playing different roles for different relations.

TransR/CTransR [10] is proposed based on the idea that entities and relations should be considered in two different vector spaces. TransR set a mapping matrix for each relation to map entity embedding into a relation vector space. CTransR is a clustering-based extension of TransR, where diverse head-tail entity pairs are clustered into different groups and each group has only one relation vector for all pairs in the group.

TranSparse [11] consider the heterogeneity (some relations link many entity pairs and others do not) and the imbalance (the number of head entities and that of tail entities in a relation could be different) of knowledge graphs. It uses adaptive sparse matrices to replace transfer matrices, in which the sparse degrees are determined by the number of entity pairs linked by relations.

KG2E [10] uses Gaussian embedding to explicitly model the certainty of entities and relations. Each entity or relation is represented by a Gaussian distribution, where the mean denotes its position and the covariance can properly represent its certainty. It performs well on $1-to-N$ and $N-to-1$ relations.

TransG [23] can discover the latent semantics of a relation automatically through Chinese Restaurant Process, and leverages a mixture of multiple relation-specific components for translating entity pair to address new issues.

TransGate [25] is proposed to relieve the pressure of the huge number of parameters in semantic discriminate models. Across the whole knowledge graph, it establishes two fixed-size shared parameter gates based on the traditional LSTM [26] gate by utilizing the inherent correlation between relations. The shared parameter gate is used to filter entity vectors according to certain semantic environment, and the filtered entity vectors only represent the semantic in the current semantic environment.

2.3 Other Models

Many researches attempt to introduce some novel techniques of deep learning into knowledge graph embedding. KBGAN [4] introduces GAN (Generative Adversarial Networks) to boost several embedding models. ProjE [17] uses a learnable combination operator to combine embeddings and feeds combined embeddings in a multi-class classifier to handle complex relations. R-GCN [16] and ConvE [5] introduces multi-layer convolution network in knowledge graph embedding. ConvKB [15] employs a convolutional neural network to capture global relations and transitional characteristics between entities and relations .

TransMVG does not use many sets of semantic parameters specific to each relationship, nor does it use deep learning frameworks. It only uses a smaller set of global shared parameters based on two multiple-valued gates and a shallow learning framework to get a better performance on link prediction task and triplet classification task.

3 Embedding Based on Multiple-Valued Gate

In this paper, we propose the TransMVG model. Its gate filtering vectors of all dimensions can take the values between 0 and 1 with nearly the same proportions, ensuring that the values of each dimension obey different *Bernoulli* distribution respectively at the same time. In this section, we first introduce some background knowledge and then introduce the TransMVG model and its training methods. Finally, we perform a complexity analysis, comparing TransMVG with other baselines.

3.1 Background Knowledge

LSTM. LSTM [26] is a kind of recurrent neural network which overcomes long-term dependence. It consists of a memory cell and three gates, namely input gate, forget gate and output gate. The input gate selects which information elements are related to the existing state and can be added to the memory cell. The forget gate determines which information elements can be filtered out. The output gate determines which inputs can be entered to the next step based on the existing state.

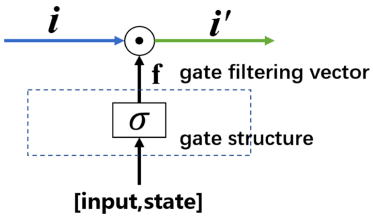

Fig. 2. Traditional gate

Gate is the core mechanism of LSTM, and its function is to let information pass selectively. A gate consists of a full connection layer and a sigmoid activation function. The gate vector and the vector to be filtered perform the Hadamard product operation to finish the information filtering. The feedforward form of gate is:

$$\mathbf{f} = \sigma\left(\mathbf{W}\left[\boldsymbol{input}, \boldsymbol{state}\right] + \mathbf{b}\right) \tag{1}$$

$$\boldsymbol{i}^{'} = \boldsymbol{i} \bigodot \mathbf{f} \tag{2}$$

$\forall x \in \mathbb{R}$, $\sigma(x) = \frac{1}{1+exp(-x)}$, $0 \leq \sigma(x) \leq 1$. That is, all the values of gate filtering vector $\mathbf{f}$ is between 0 and 1. As shown in Figure 2, ***input*** represents new information of the current gate, and ***state*** represents the memory sum of all previous input information. Based on the current ***state*** and the new ***input***, a gate filtering vector $\mathbf{f}$ is generated. Each value of $\mathbf{f}$ indicates how much of each information element in a vector should be allowed passing and how much should be forgotten. $\boldsymbol{i}$ is the vector to be filtered, $\boldsymbol{i}^{'}$ is the vector filtered through the gate structure, $\bigodot$ is the counterpoint multiplication.

Noise Theorem. The following theorem has already been proved in [12]. It can be used to extend one Bernoulli distribution to many independent Bernoulli distributions by adding random noise. The value of $G(\alpha, \tau)$ can be controlled by controlling the temperature parameter τ . τ is a parameter greater than zero, which controls the soft degree of *sigmoid*. The higher the temperature, the smoother the generated distribution. The lower the temperature, the closer the generated distribution is to the discrete one-hot distribution.

Theorem 1. *Assume $\sigma(\cdot)$ is the sigmoid function. Given $\alpha \in \mathbb{R}$ and temperature $\tau > 0$, we random variable $D_\alpha \sim B(\sigma(\alpha))$ where $B(\sigma(\alpha))$ is the Bernoulli distribution with parameter $\sigma(\alpha)$, and define*

$$G(\alpha, \tau) = \sigma\left(\frac{\alpha + logU - log(1-U)}{\tau}\right) \tag{3}$$

where $U \sim Uniform(0,1)$. Then the distribution of $G(\alpha, \tau)$ can be considered as an approximation of Bernoulli distribution $B(\sigma(\alpha))$.

3.2 TransMVG

A value of each dimension of entity vectors and relation vectors corresponds to an information element. In different semantic environment, the information elements of each dimension pass through the gate in different proportions. The essence of multiple semantics is the combination of information elements with different proportions. Our main contribution is eliminating the extreme filtering of traditional gates and making the information elements get more accurate combinations.

Model. Shown in Fig. 3, TransMVG uses three real number vectors $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t} \in \mathbb{R}^m$ in a same vector space to represent each triplet in a knowledge graph. Two full connection layers $\boldsymbol{W}_h \cdot [\boldsymbol{h}, \boldsymbol{r}] + \boldsymbol{b}_h$ and $\boldsymbol{W}_t \cdot [\boldsymbol{t}, \boldsymbol{r}] + \boldsymbol{b}_t$ are set for learning the global shared parameter gates of head entity vectors and tail entity vectors respectively. Gate filtering vectors $\mathbf{f}_h$ and $\mathbf{f}_t$ generated by full connection layers after a *sigmoid* operation will carry on Hadamard multiplication with head entity vectors and tail entity vectors respectively to select how much the information element of each dimension should pass.

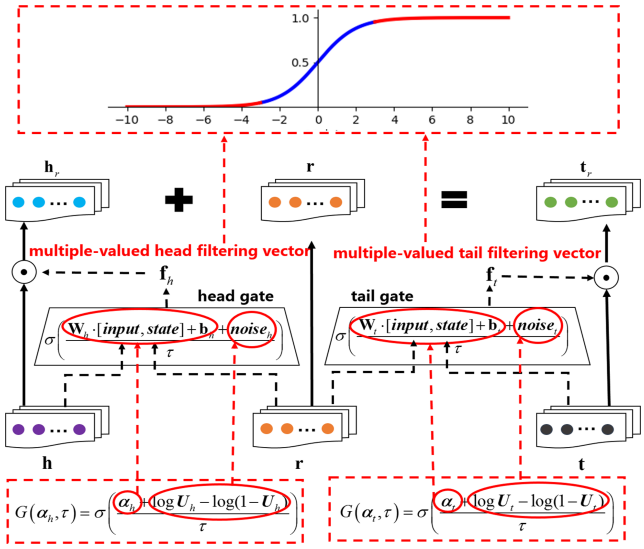

Fig. 3. TransMVG

The relation vector (semantic environment) in a triplet can be regarded as the ***state*** in the aforementioned LSTM gate. Taking the head gate as an example, the working object of the full connection weight in the traditional LSTM gate is the concatenation of new input and the current state. The new input can be understood as the entity vector to be filtered $\boldsymbol{h}$, and the current state can be regarded as the semantic environment, namely the relation vector $\boldsymbol{r}$. Therefore,

the object that the full connection layer will work on is the concatenating of the head entity vector $\boldsymbol{h}$ and the relation vector $\boldsymbol{r}$. The weight matrix of full connection layer is $\boldsymbol{W}_h \in \mathbb{R}^{m\times 2m}$, the bias is $\boldsymbol{b}_h \in \mathbb{R}^m$. The output of the head entity gate is the filtering vector $\mathbf{f}_h$ of the head entity vector $\boldsymbol{h}$. A Hadamard multiplication is performed between $\mathbf{f}_h$ and $\boldsymbol{h}$ to realize the specific semantic representation of entity $\boldsymbol{h}$ in the sematic environment $\boldsymbol{r}$. In a similar way, the specific semantic representation of entity $\boldsymbol{t}$ in the sematic environment $\boldsymbol{r}$ can be obtained.

The values gate filtering vectors generated by traditional gates are mostly in a boundary binary-valued state(0 or 1). We use the ***Theorem*** 1 to change the extreme filtering pattern to a general one. α in Eq. 3 can be regarded as any value of the vector generated by the full connection layer, $(\boldsymbol{noise}_h)_i$ and $(\boldsymbol{noise}_t)_i$ can be regarded as the noise for each dimension. According to Eq. 3, we can get Eqs. 6 and 7 as following:

$$(\boldsymbol{W}_h \cdot [\boldsymbol{h}, \boldsymbol{r}] + \boldsymbol{b}_h)_i = (\alpha_h)_i \tag{4}$$

$$(\boldsymbol{W}_t \cdot [\boldsymbol{t}, \boldsymbol{r}] + \boldsymbol{b}_t)_i = (\alpha_t)_i \tag{5}$$

$$(\boldsymbol{noise}_h)_i = log\,(\boldsymbol{u}_h)_i - log(1 - (\boldsymbol{u}_h)_i) \tag{6}$$

$$(\boldsymbol{noise}_t)_i = log\,(\boldsymbol{u}_t)_i - log(1 - (\boldsymbol{u}_t)_i) \tag{7}$$

$i = 1, ..., k$,k is the number of the vector dimension. Then, according to ***Theorem*** 1, values of the i-th dimension of all head filtering gates and all tail filtering gates nearly obey Bernoulli distribution respectively, as Eqs. 8 and 9.

$$\sigma\left(\frac{(\boldsymbol{W}_h \cdot [\boldsymbol{h}, \boldsymbol{r}] + \boldsymbol{b}_h)_i + (\mathbf{noise}_h)_i}{\tau}\right) \sim Bernoulli \tag{8}$$

$$\sigma\left(\frac{(\boldsymbol{W}_t \cdot [\boldsymbol{t}, \boldsymbol{r}] + \boldsymbol{b}_t)_i + (\mathbf{noise}_t)_i}{\tau}\right) \sim Bernoulli \tag{9}$$

For the whole vector generated by the full connection layer, the above operation is performed for each dimension. Then we can get general vectors by adding noise to each dimension respectively, as Eqs. 10-13:

$$\mathbf{f}_h = \sigma\left(\frac{\boldsymbol{W}_h \cdot [\boldsymbol{h}, \boldsymbol{r}] + \boldsymbol{b}_h + \mathbf{noise}_h}{\tau}\right) \tag{10}$$

$$\mathbf{f}_t = \sigma\left(\frac{\boldsymbol{W}_t \cdot [\boldsymbol{t}, \boldsymbol{r}] + \boldsymbol{b}_t + \mathbf{noise}_t}{\tau}\right) \tag{11}$$

$$\boldsymbol{noise}_h = log\boldsymbol{U}_h - log\,(\mathbf{1} - \boldsymbol{U}_h) \tag{12}$$

$$\boldsymbol{noise}_t = log\boldsymbol{U}_t - log\,(\mathbf{1} - \boldsymbol{U}_t) \tag{13}$$

The temperature τ is a hyper-parameter and the gate value of the gate filtering vectors can be approximated to the none-flat region (The blue part of the *sigmoid* function in Fig. 3) of the *sigmoid* function by adjusting τ . The elements in $\boldsymbol{U}_h$ and $\boldsymbol{U}_t$ are independent with each other and are sampled from

a uniform distribution (0,1) respectively. Since the noise added for the values of each dimension is independent with each other, the gate values obtained for each dimension will conform to the Bernoulli distribution with different independent distributions respectively, that is, all values between 0 and 1 can be possible to be taken with similar possibilities.

After generating the gate filtering vector with precise filtering function, the entity can achieve the specific semantic representation in a certain semantic environment as shown in Eqs. 14 and 15.

$$\boldsymbol{h}_r = \boldsymbol{h} \bigodot \mathbf{f}_h \tag{14}$$

$$\boldsymbol{t}_r = \boldsymbol{t} \bigodot \mathbf{f}_t \tag{15}$$

After the semantic representation filtering, we make a translation in a specific semantic environment to obtain the distance function:

$$d_r\left(h,t\right) = \left\|\boldsymbol{h}_r + \boldsymbol{r} - \boldsymbol{t}_r\right\|_{L1/L2} \tag{16}$$

The smaller the distance, the better for the correct triplets, and the larger the distance, the better for the wrong triplets.

Training. In the training process, the maximum interval method is used to optimize the objective function to enhance the distinguishing ability of knowledge representation. For each (h,r,t) and its negative sample (h',r,t'), TransMVG aims to minimize the hinge-loss as following:

$$L = max\left(\gamma + d_r\left(h,t\right) - d_r\left(h',t'\right),0\right) \tag{17}$$

where γ is a margin hyper-parameter and (h',r,t') is a negative sample from the negative sampling set. A negative sample can be obtained by randomly replacing the head or the tail of a correct triplet with an entity from the entity list. The loss function 17 is used to encourage discrimination between training triplets and corrupted triplets by favoring lower scores for correct triplets than for corrupted ones. The training process of TransMVG is carried out using Adam optimizer with constant learning rate.

Complexity Analyisis. As shown in Table 1, we compare TransMVG with famous semantic discriminate models and some classical indiscriminate models. The statistics of parameters of these models are from [25]. N_e is the number of entities, m is the dimension of entities, N_r is the number of relations, n is the dimension of relations. k is the number of hidden nodes of a neural network and s is the number of slice of a tensor. $\widehat{\Theta}$ denotes the average sparse degree of all transfer matrices.

From the table, we can see TransMVG enables semantic discrimination with fewer parameters and lower complexity, and requires no pretraining. Compared with TransGate, TransMVG adds only one hyper parameter τ to get noise and temperature control for all values of the fully connected layers.

4 Experiments

In this section, we empirically evaluate TransMVG on two key tasks: link prediction and triplet classification. We demonstrate that TransMVG outperforms most state-of-art baselines on multiple benchmark datasets.

Table 1. Complexity analyisis

Models	Embedding parameters	Discriminate parameters	Hyper parameters	Time complexity	Pre-training
TransE	$O(N_e m + N_r n), (m = n)$	None	2	$O(m)$	None
TransH	$O(N_e m + N_r n), (m = n)$	$O(N_r n)$	4	$O(m)$	TransE
DistMult	$O(N_e m + N_r n), (m = n)$	None	2	$O(m)$	None
TransR	$O(N_e m + N_r n)$	$O(N_r mn)$	3	$O(mn)$	TransE
CTransR	$O(N_e m + N_r n)$	$O(N_r mn)$	4	$O(mn)$	TransR
TransD	$O(N_e m + N_r n)$	$O(N_e m + N_r n)$	3	$O(m)$	TransE
TranSparse	$O(N_e m + N_r n)$	$O(2N_r(1 - \hat{\theta})mn), (0 \leq \hat{\theta} \leq 1)$	5	$O(2(1 - \hat{\theta})mn), (0 \leq \hat{\theta} \leq 1)$	TransE
ComplEx	$O(2N_e m + 2N_r n), (m = n)$	None	2	$O(m)$	None
CompbinE	$O(2N_e m + 2N_r n), (m = n)$	None	2	$O(2m)$	None
ProjE	$O(N_e m + N_r n + 5m), (m = n)$	None	2	$O(N_e m + 2m)$	None
TransGate	$O(N_e m + N_r m + 5m)$	$O(4m^2 + 2m)$	2	$O(m^2)$	None
TransMVG	$O(N_e m + N_r n + 5m)$	$O(4m^2 + 2m)$	3	$O(m^2)$	None

4.1 Datasets

Link prediction and triplets classification are implemented on two large-scale knowledge bases: WordNet [14] and Freebase [1]. Twodata sets are employed from WordNet. Among them, WN11 [2] is for link prediction, and WN18RR [5] is for triplet classification. Three datasets from Freebase are used. Among them, FB15K [2] and FB15K-237 [20] are for link prediction, and FB13 [18] is for triplet classification. The details of these datasets are in Table 2.

4.2 Link Prediction

Link prediction aims to predict the missing h or t for a triplet (h, r, t). i.e., predict t given (h, r) or predict h given (r, t). Instead of giving one best answer,

Table 2. Statistics of datasets

Datasets	Rel	Ent	Train	Valid	Test
WN11 [2]	11	38,696	112,581	2,609	10,544
WN18RR [5]	11	40,943	86,835	3,034	3,134
FB13 [18]	13	75,043	316,232	5,908	23,733
FB15K [2]	1,345	14,951	483,142	50,000	59,071
FB15K-237 [20]	237	14,541	272,115	17,535	20,466

this task ranks a set of candidate entities from the knowledge graph. For each testing triplet (h, r, t), we corrupt it by replacing the tail t with every entity e in the knowledge graph and calculate all distance scores. Then we rank the scores in ascending order, and get the rank of the original. In fact, a corrupted triplet may also exist in the knowledge graph, which should be also considered as correct.

We filter out the correct triplets from corrupted triplets that have already existed in the knowledge graph to get the filtered results. With use of these filtered ranks, we get three commonly used metrics for evaluation: the average rank of all correct entities (Mean Rank), the mean reciprocal rank of all correct entities (MRR), and the proportion of correct entities ranked in top k(Hits@k). A good link prediction result expects a lower Mean Rank, a higher MRR, and a higher (Hits@k).

In this task, we use three datasets: WN18RR [5], FB15K [2] and FB15K-237 [20]. For three datasets, we search the learning rate α for Adam among $\{0.001, 0.01, 0.1\}$, the temperature τ among $\{100, 200, 500\}$ the margin γ among $\{2, 4, 6, 8, 10\}$, the embedding dimension m among $\{50, 100, 200\}$, and the batch size B among $\{1440, 2880, 5760\}$. The optimal configurations are as follow: on WN18RR, $\gamma = 8$, $\alpha = 0.1$, $\tau = 200$, $m = 200$, $B = 2880$ and taking $L1$ distance; on FB15K, $\gamma = 4$, $\alpha = 0.1$,$\tau = 200$, $m = 200$, $B = 5760$ and taking $L1$ distance; on FB15K-237, $\gamma = 4$, $\alpha = 0.1$, $\tau = 200$, $m = 200$, $B = 5760$ and taking $L1$ distance.

In Table 3, the best scores are in bold, while the second best scores are in underline. In the table, we observe that: (1) On FB15K-237, TransMVG outperforms all baselines at MRR, Hits@10 and Hits@1 metrics, improved by 1.4%, 6.0% and 20.7% respectively compared with the underlined second ranks. (2) On FB15K, TransMVG outperforms all baselines at Mean Rand metric and get a second good rank at Hits@10 metric. (3) On WN18RR, TransMVG outperforms all baselines at Hits@10 metric, improved by 2.7% compared with the underlined second ranks. This indicates the great ability of TransMVG on precise link prediction.

As the reversing relations have been removed in WN18RR, the semantic hierarchy of the relations in the database is no longer complete. As a result, the TransMVG model does not have a complete corpus to adequately learn

Table 3. Evaluation results on link prediction

Datasets	WN18RR				FB15K				FB15K-237			
Metrics	MRR	MR	Hits@10	Hits@1	MRR	MR	Hits@10	Hits@1	MRR	MR	Hits@10	Hits@1
TransE [2]	0.226	3384	50.1	-	0.220	125	47.1	23.1	0.294	347	46.5	14.7
DistMult [24]	0.43	5110	49.0	39	0.654	97	82.4	54.6	0.241	254	41.9	15.5
TransD [10]	-	-	42.8	-	0.252	67	77.3	23.4	-	-	45.3	-
CombineE [19]	-	-	-	-	0.283	-	85.2	55.4	-	-	-	-
ComplEX [21]	**0.44**	5261	51.0	**41.0**	0.692	-	84.0	59.9	0.247	339	42.8	15.8
KB-LRN [6]	-	-	-	-	0.794	44	87.5	74.8	0.309	209	49.3	21.9
NLFeat [20]	-	-	-	-	0.822	-	87.0	-	0.249	-	41.7	-
RUGE [7]	-	-	-	-	0.768	-	86.5	70.3	-	-	-	-
KBGAN [4]	0.213	-	48.1	-	-	-	-	-	0.278	-	45.8	-
R-GCN [16]	-	-	-	-	0.696	-	84.2	60.1	0.248	-	41.7	15.3
TransG [23]	-	-	-	-	0.657	51	83.1	55.8	-	-	-	-
ConvE [5]	0.43	4187	52.0	40.0	0.657	51	83.1	55.8	0.325	244	50.1	23.7
ConvKB[15]	0.248	**2554**	52.5	-	0.768	-	-	-	0.396	257	51.7	-
TransGate [25]	0.409	3420	51.0	39.1	**0.832**	33	**91.4**	**75.5**	0.404	**177**	58.1	25.1
TransMVG	0.253	4391	**53.9**	3.9	0.630	**31**	88	46.5	**0.410**	223	**61.2**	**30.3**

the information element interactions of various relations. Thus, TransMVG only perform best on one metirc on WN18RR.

FB15K contains a number of redundant relations. This may inhibit fine semantic recognition ability of TransMVG to some extent. Therefore, on FB15K, TransMVG only performed best in one metric. Fb15K-237 is obtained by removing the redundant relations in FB15K. TransMVG has a good performance almost beyond various baselines at all metrics. The result also shows that the multi-valued gate in TransMVG have more powerful multi-semantic learning ability than the boundary binary-valued gate in TransGate.

4.3 Triplet Classification

Triplet classification aims to judge whether a given triplet (h, r, t) is correct or not. It is first used in [18] to evaluate knowledge graph embeddings learned by NTN model. In this paper we use WN11 [2] and FB13 [18] as the benchmark datasets for this task. These two datasets contain positive and negative triplets. For each triplet (h, r, t), if the value calculated by the distance score Eq. 16 is above a relation-specific threshold Δ, then the triplet will be classified as positive, otherwise it will be classified as negative.

For WN11 and FB13, we compare TransMVG with baselines reported in [25]. In training, for the two datasets, we search the learning rate α among $\{0.001, 0.01, 0.1\}$, the temperature τ among $\{100, 200, 500\}$, the margin γ among $\{2, 4, 6, 8, 10\}$, the embedding dimension m among $\{50, 100, 200\}$, the batch size B among $\{1440, 2880, 5760\}$. The optimal configurations are as follow: On WN11, $\gamma = 10$, $\alpha = 0.01$, $\tau = 100$, $m = 100$, $B = 2880$ and taking L1 distance. On FB13, the best configurations are: $\gamma = 6$, $\alpha = 0.001$, $\tau = 100$, $m = 100$,

Table 4. Evaluation results on triplet classification

Datasets	WN11	FB13
SE [3]	53.0	75.2
SME [3]	70.0	63.7
LFM [9]	73.8	84.4
SLM [18]	69.9	85.3
NTN [18]	70.4	87.1
TransE [2]	75.9	70.9
TransH [22]	77.7	76.5
Trans R[13]	85.5	74.7
CTransR [13]	85.7	-
KG2E [8]	85.4	85.3
TransD [10]	85.6	**89.1**
TransSparse [11]	86.8	86.5
TransG [23]	87.4	87.3
TransGate [25]	87.3	88.8
TransMVG	**89.5**	84

$B = 2880$ and taking L1 distance. Table 4 shows the detailed evaluation results of triplets classification. From the table, we observe that: (1) On WN11, our method obtains the accuracy of 89.5% and outperforms all baseline models. (2) On FB13, the accuracy of our method is only in the middle of the rank list.

The WN11 dataset includes $1-1$, $1-N$, $N-1$ three relation types. There are 113,000 triples in its training set. The FB13 dataset includes $N-1$, $N-N$ two relation types. There are 316,000 triplets in its training set. This indicates that FB13 is more dense than WN11 both in relation nature and the number of entity pairs connected by each relation. So we guess if the dimension of TransMVG can go beyond the current maximum set 200, it will achieve better performance under richer proportion combination of more information elements.

4.4 Distribution Visualization

As shown in Fig. 5, we run TransGate and TransMVG on FB15K with the same settings, the initial values of the two models conform to both nearly binary-valued distributions. TransGate has an obvious trend of boundary binary-valued trend when it is run 100 times. TransMVG, on the other hand, obtains multiple-valued gates in a gentle manner. In fact, in both the TransGate and TransMVG models, the value of any dimension of the gate filtering vector conforms to a binary distribution, and each dimension distribution is independent with each other. The binary distribution of the gate values of each dimension in TransGate tends to be a boundary binary-valued distribution, while that of the gate values in TransMVG tends to be many independent different distributions due to the addition of noise, resulting in a mutiple-valued gate.

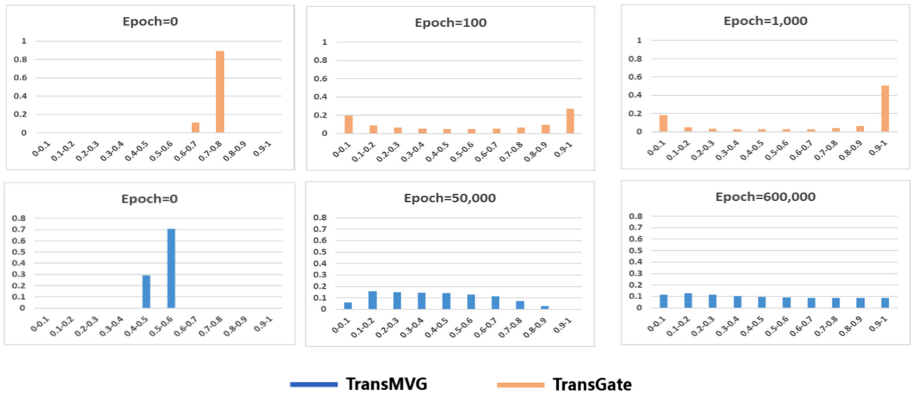

Fig. 4. The value distributions of the gate filtering vectors in TransGate and TransMVG

5 Conclusion

In this paper, we focus on embedding the knowledge graph into a low-dimension vector space for knowledge graph completion. We find that extreme filtering problem exsits in the traditional method based on shared parameter gate, and the main reason is the boundary binary-valued distribution of its gate filter values. We propose an information element interaction mechanism to explain the multi-semantic representation of the same vector in different semantic environments. Our TransMVG model refined the interaction of information elements by adding independently distributed noise to the full connection layer of the shared parameter gate and pushing the gate values to be multi-valued. We have conduct a number of experiments on link prediction task and triplet classification task. The experiments results show that TransMVG almost outperforms state-of-the-art baselines. This means it is feasible and necessary to multivalue the filter gate vectors in the process of knowledge representation learning.

In TransMVG, only the multi-semantics of entities have been taken into consideration. In the future, we will try to deal with the multi-semantics of both entities and relations at the same time.

References

1. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of data, pp. 1247–1250 (2008)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, pp. 2787–2795 (2013)

3. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Twenty-Fifth AAAI Conference on Artificial Intelligence, pp. 301–306 (2011)
4. Cai, L., Wang, W.Y.: Kbgan: Adversarial learning for knowledge graph embeddings. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1470–1480 (2018)
5. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: The Thirty-Second AAAI Conference on Artificial Intelligence, pp. 1811–1818 (2018)
6. Garcia-Duran, A., Niepert, M.: Kblrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features. In: Proceedings of UAI (2017)
7. Guo, S., Wang, Q., Wang, L., Wang, B., Guo, L.: Knowledge graph embedding with iterative guidance from soft rules. In: Thirty-Second AAAI Conference on Artificial Intelligence, pp. 4816–4823 (2018)
8. He, S., Liu, K., Ji, G., Zhao, J.: Learning to represent knowledge graphs with gaussian embedding. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 623–632 (2015)
9. Jenatton, R., Roux, N.L., Bordes, A., Obozinski, G.: A latent factor model for highly multi-relational data. Adv. Neural Inf. Process. Syst. **4**, 3167–3175 (2012)
10. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, pp. 687–696 (2015)
11. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: Thirtieth AAAI Conference on Artificial Intelligence, pp. 985–991 (2016)
12. Li, Z., et al.: Towards binary-valued gates for robust LSTM training. In: Proceedings of the 35th International Conference on Machine Learning (2018)
13. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-ninth AAAI Conference on Artificial Intelligence, pp. 2181–2187 (2015)
14. Miller, G.: Wordnet: a lexical database for English. Commun. ACM. **38**, 39–41 (1995)
15. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: A novel embedding model for knowledge base completion based on convolutional neural network. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 327–333 (2017)
16. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference, pp. 593–607 (2018)
17. Shi, B., Weninger, T.: Proje: Embedding projection for knowledge graph completion. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 1236–1242 (2017)
18. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems, pp. 926–934 (2013)
19. Tan, Z., Zhao, X., Wang, W.: Representation learning of large-scale knowledge graphs via entity feature combinations. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1777–1786 (2017)

20. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, pp. 57–66 (2015)
21. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33rd International Conference on Machine Learning, pp. 2071–2080 (2016)
22. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 1112–1119 (2014)
23. Xiao, H., Huang, M., Zhu, X.: Transg: A generative model for knowledge graph embedding. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 2316–2325 (2016)
24. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (2014)
25. Yuan, J., Gao, N., Xiang, J.: TransGate: Knowledge graph embedding with shared gate structure. Proc. AAAI Conf. Artif. Intell. **33**, 3100–3107 (2019)
26. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)