### Thanks for all reviewers for providing useful comments.

### For # Review #1

### Details of generating structured sparse matrix.

### Assume that the sparse degree is \theta, the size of transfer matrix is m x n, and the number of non-zero elements is d. Then d = [ m x n x (1-\theta) ], where the operation [a] returns the maximal integer that no more than a. We let the number of non-zero elements of structured sparse matrix is the closest to d (maybe not d) under the condition that all the non-zero entries are symmetrical along the diagonal.

### How the structured and unstructured sparse matrix make differences in model performance?

### Besides the speed difference, unstructured pattern can get better prediction accuracy than structured pattern in most cases.

### For # Review #2

### why using adaptive sparse matrix can achieve a better results than TransR?

### TransR seems to be a more general form, but it is easy to happen overfitting on simple relations or underfitting on complex relations because all relations have the same parameters to learn. To solve the issues, we use less parameters for simple relations and more parameters for complex relations than TransR. Therefore, in experiments, our model obtains better performance than TransR both on simple and complex relations.

1. Why we use adaptive sparse matrix rather than other method such as Low Rank matrix?

Based on our motivations, we need to use matrices with higher and lower degrees of freedom to learn complex and simple relations, respectively. The degree of freedom of a weight matrix refers to the number of variates which are mutual independence and can be assigned values freely. For a weight matrix M, the low-rank and sparsity both can reduce the degree of freedom, they are both constrains enforced on the matrix M. Specifically, low-rank enforces some variables to satisfy specific constraints so that the variables in M can’t be assigned freely. Thus, the degree of freedom is reduced. For sparse matrices, we let some elements are zeros and don’t change their values during training, and the other entries are freedom variates which can be learned by training. The degree of freedom is the number of variates learned by training.

But the sparse matrix is more suitable for our tasks for the two reasons:

1. Sparse matrix is more flexible than low-rank matrix. We assume that the weight matrix M has size m x n (without loss of generality, m<=n). So the maximum rank of M is m. If we use the low-rank to control the degree of freedom, we can only obtain m low-rank matrices for the relations in KBs because the rank can determine the degree of freedom of the matrix M, which is supported by the Proposition:

A n\_1 x n\_2 rank-k matrix has k(n\_1 + n\_2) – k^2 degrees of freedom. (The proves can be found in “Low-Rank Matrix Completion, Ryan Kennedy, 2013, page 6, Proposition 1.)

However, if we use the sparsity to control the degree of freedom, we can obtain m x n matrices for the relations because the matrix M contains m x n elements.

Therefore, for the dataset which has many relations, such as FB15k (1,345 relations), sparsity is more flexible than low-rank.

1. Sparse matrix is more efficient than low-rank matrix. In sparse matrix, only the non-zero entries involves in computing, which can reduce the calculation significantly. But, the low-rank matrix doesn’t have the advantage. Therefore, with sparse matrix, our model can extend on large-scale kBs easily.

### About the hyper-parameter \theta.

### This may be caused by the two datasets’ properties. As shown in Table 2, FB15k contains 1,345 relations and 483,142 triplets, but FB13 has only 13 relations and 316,232 triplets. Therefore, FB15k is a very sparse graph, and FB13 is very dense graph. The hyper-parameter can be effect by datasets.

### 4.  In the experiments, the results of "US" are average results of 5 times. It seems to be unfair to the results of "S".

### In unstructured pattern, the non-zero elements locate with an uniform random distribution, but in structured pattern, their locations are fixed. Therefore, we do experiments 5 times for “US” to ensure the results are believable, and 1 time for “S”.

### # Review #3

### About the definition of complexity of relations.

### We argue that the complexity of a relation is proportional to the number of triplets (or entities) linked by it. Because the more data linked by the relation, the more knowledge is related to it. In the view of intuition, if a relation links much knowledge, the relation is complex.

### The relation “gender” links 59,423 (18.79%) triplets in FB13 train data, its head links 59,394 person names and its tail only links male, female. The complexity of relation “gender” focus on the head because the names come from 186 nations and each nation has its own naming habits. Furthermore, another trouble is that the relation “gender” is difficult to reason. Maybe we can infer a person’s gender according to the name (Socher et al., 2013, it shows that we can infer the “gender” from the words of a person’s name). However, in our approach, we encode each entity into a vector, not the components (words) of an entity. Thus, the name information can’t be used to infer the relation “gender”. In fact, the prediction accuracy of “gender” in classification task is lower than that of other relations (only higher than that of relation “cause\_of\_death”). The prediction accuracies of different relations in FB13 are shown in the following.

### cause\_of\_death 0.831

### gender 0.851

### profession 0.879

### religion 0.890

### nationality 0.946

### institution 0.869

### ethnicity 0.893

### Therefore, the relation “gender” is not so simple as it seems.

### And, the paper “Effective Blending of Two and Three-way Interactions for Modeling Multi-relational Data” also proposed

### Why use sparse matrix rather than low-rank matrix?

### This has been answered in the question 2 of #Review#2.

### In their formulation, the authors define the induced matrix as M(\theta) to denote a matrix M with \theta “sparse degrees”, fraction of number of zero entries in the matrix. The objective parameters are learning w, a decomposition of M(\tehta), [among others]. I am not aware of a technique that achieves this. I strongly urge the authors to explicitly write the objective function and the parameters that it is optimized over, in particular how theta is formulated in the objective function.

### I’m sorry that I can’t understand the question clearly, especially the sentence “The objective parameters are learning w, a decomposition of M(\tehta), [among others]”. But I can describe the objective and parameters that it is optimized, and the process of learning.

### First, I explain how \theta is formulated. \theta\_{min} is a hyper-parameter which need to be assigned value by human, the other \theta(s) are determined by equation (4) or (6). Please note that, \theta(s) are not formulated in the training process, we obtain them by (4) or (6) before training (All the values of notations in (4) and (6) can be obtained by dataset statistics). We first calculate the \theta(s) for all transfer matrices, then we compute the number of nonzero elements and determine the locations of them(nonzero elements) according to section 3.1. During training，we only calculate the partial derivatives of the nonzero elements and update them, all the zero elements don’t involve in computing.

### All the parameters contain entity vectors, relation vectors and the nonzero elements in transfer matrices. The objective is the ranking-loss function describe by (9), which is similar to TransR. We would show the details of the learning algorithm in our paper.