Enhancing Knowledge Graph Completion with Positive Unlabeled Learning

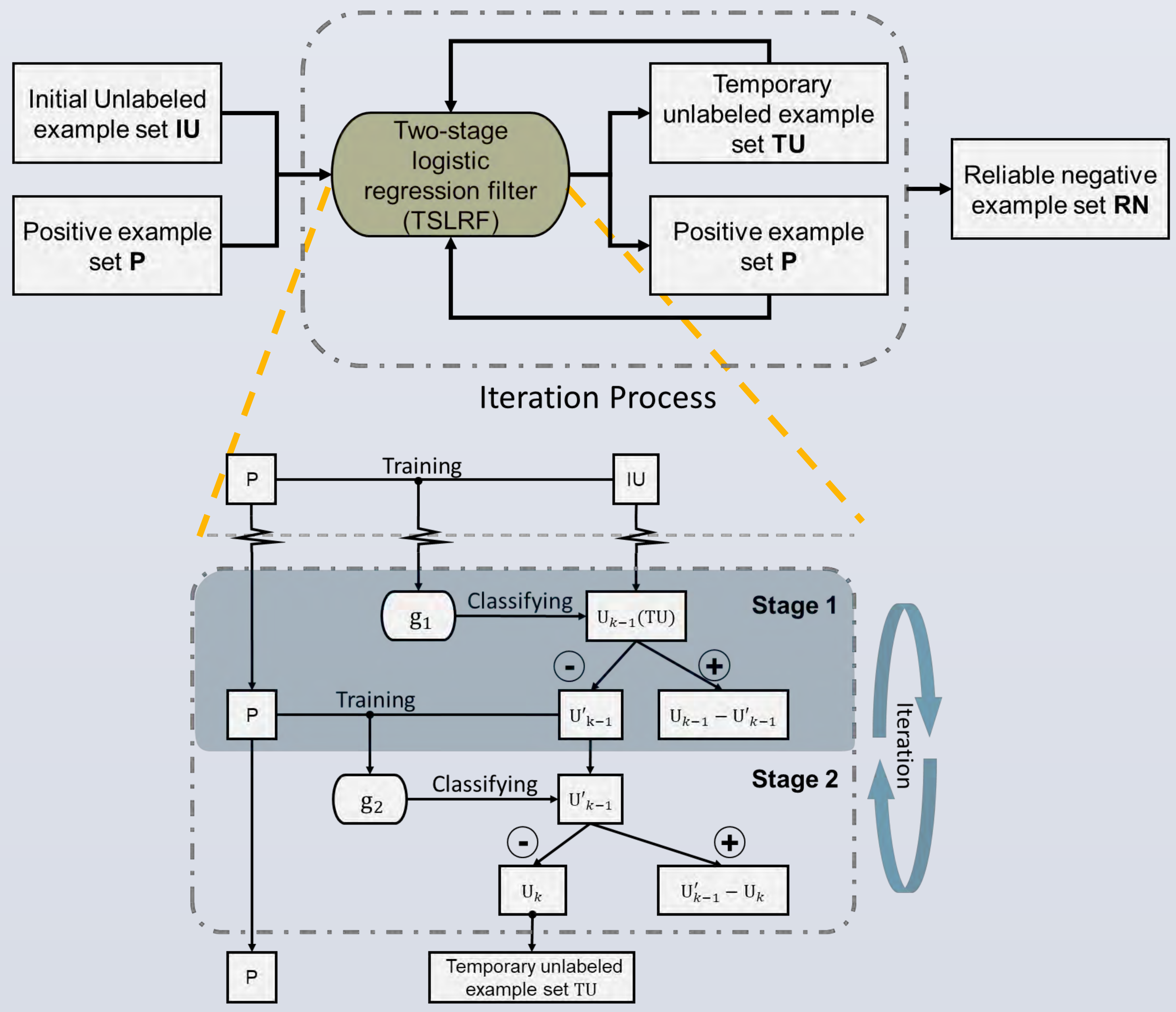
Jinghao Niu, Zhengya Sun, Wensheng Zhang
Institute of Automation, Chinese Academy of Sciences

Introduction

Motivation: Knowledge graphs have proven to be incredibly useful for many artificial intelligence applications. Although typical knowledge graphs may contain a huge amount of facts, they are far from being complete, which motivates an increasing research interest in learning statistical models for knowledge graph completion. Learning such models relies on sampling appropriate number of negative examples, as only the positive examples are contained in the data set. However, this would introduce errors or heuristic biases which restrict the sampler to visit other potentially reliable negative examples for better prediction models.

Proposed Method: In this paper, we propose a two stage logistic regression filter (TSLRF), i.e. a novel negative example generation approach based on the positive-unlabeled learning framework. Specifically, it extracts a set of reliable negative examples from the initial unlabeled data, which together with the available positive examples, are then used to train a binary classifier. It performs in an iterative manner and outputs the set of the low scoring negative candidates for the downstream training. We further devise a novel embedding-based model which works with cost-sensitive losses, by weighting the semantic differences between negative examples and particular positive ones. This weighting scheme reflects the importance of predicting the preferences between them correctly.

Generating Reliable Negative Examples



Architecture of Two-Stage Logistic Regression Filter (TSLRF).

Generating RN: our proposed prediction framework first generates the initial unlabeled example set IU and positive example set P depending on known triples in the KGs. For every positive example (h, r, t), we generate two types of unlabeled examples (h', r, t) and (h, r, t') (unknown triples in the training set) by random sampling under local closed world assumption. Every triple will correspondingly generate 10 examples for both types, which make up the initial unlabeled example set. The initial example set IU and P are then fed into TSLRF. We use TSLRF in an iterative manner until the convergence of the unlabeled example set or achieving the max iteration number. The output of TSLRF is the reliable negative example set RN.

Loss Function of TSLRF:

$$L_s = -\sum_{i=1}^{(N^+ + N^-)} \{\alpha \cdot y_i \cdot ln(\sigma(\boldsymbol{w}\mathbf{x}^T + w_0)) + (1 - y_i) \cdot ln(1 - \sigma(\boldsymbol{w}\mathbf{x}^T + w_0))\} + \beta \cdot \|\boldsymbol{w}\|_1$$

where $\alpha > 1$ is used to control the importance ratio of positive and negative examples β is the regularization parameter; σ denotes the logistic function; N^+ and N^- denote the number of positive and negative examples respectively.

Latent and Observed Features of Triples

Latent Feature: In latent feature models, each triple is represented as a score function or certain combination operator that depends only on learned embedding vectors of the entities and relations, and possibly additional global parameters.
Observed Feature: Observed feature models directly construct interpretable features for each triple, which together with their weights, are used to define the score of a triple.

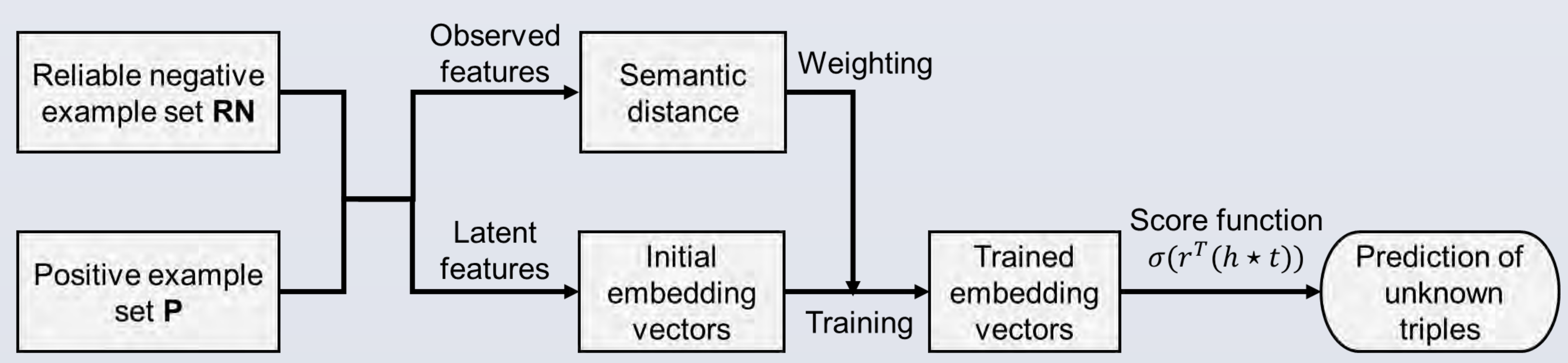
In this work, we extract six types of observed features for every candidate triple (e_i, r_k, e_j). To be specific, we employ four types of observed features (feature 1-4) introduced by Toutanova et al. and further define another two types of new observed features (feature 5 and 6):

$$\text{Feature 5:} \quad \mathbf{1}(e_i = s) = \begin{cases} 1 & (e_i, r, e) \in TrainingSet \\ 0 & otherwise \end{cases}$$

$$\text{Feature 6:} \quad \mathbf{1}(e_j = o) = \begin{cases} 1 & (e, r, e_j) \in TrainingSet \\ 0 & otherwise \end{cases}$$

where e, r denote any possible entity and relation, and r' denotes the relation that is different from r_k

Semantically Weighted Prediction Model



Semantically Weighted Loss Function:

$$L = \sum_{s \in P} \sum_{s' \in RN} \nu(s, s') \cdot [f_r(s') + \gamma - f_r(s)]_+$$

where P denotes the positive example set, RN represents the reliable negative example set, γ is the margin value, $[x]_+ = \max\{0, x\}$; f_r is the score function, the weight function $\nu(s, s')$ quantifies the semantic distance between any two triples of our concern. We estimate the contributions of different example pairs through a weight function:

$$\nu(s_1, s_2) = 1 - \frac{\sum_{z \in Z_1} \frac{\cos_z(s_1, s_2)}{|Z_1|} + \delta \cdot \sum_{z \in Z_2} \frac{\cos_z(s_1, s_2)}{|Z_2|}}{1 + \delta}$$

where δ is a weight parameter chosen through cross validation, $\cos_z(s1, s2)$ denotes the cosine distance between s_1 and s_2 in the space of feature z. All six types of features are divided into two sets: Z_1 denotes the feature set comprising feature 3 and 4, Z_2 denotes the feature set comprising feature 1, 2, 5 and 6.

Conclusion

In this paper, we first propose a PU-learning framework to iteratively improve the negative candidate pools for training triple prediction models. The experimental results on *FB15k* and *WN18* validate the effectiveness of introduced negative selection scheme for both the latent feature models and observed feature models. Then, we devise a semantic distance weighting scheme to better the pairwise loss function, which could be widely used in many triple prediction models. This weighting strategy effectively exploits additional observed features to improve the latent feature model further. Besides, experimental results also show that the combination of the two proposed schemes brings about substantial improvements over state-of-the-art methods.