Multi-Relational Classification via Bayesian Ranked Non-Linear Embeddings

Ahmed Rashed
Information Systems and Machine Learning Lab, University of Hildesheim, Germany
ahmedrashed@ismll.uni-hildesheim.de

Josif Grabocka
Information Systems and Machine Learning Lab, University of Hildesheim, Germany
josif@ismll.uni-hildesheim.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab, University of Hildesheim, Germany
schmidt-thieme@ismll.uni-hildesheim.de

ABSTRACT

The task of classifying multi-relational data spans a wide range of domains such as document classification in citation networks, classification of emails, and protein labeling in proteins interaction graphs. Current state-of-the-art classification models rely on learning per-entity latent representations by mining the whole structure of the relations’ graph, however, they still face two major problems. Firstly, it is very challenging to generate expressive latent representations in sparse multi-relational settings with implicit feedback relations as there is very little information per-entity. Secondly, for entities with structured properties such as titles and abstracts (text) in documents, models have to be modified ad-hoc. In this paper, we aim to overcome these two main drawbacks by proposing a flexible nonlinear latent embedding model (BRNLE) for the classification of multi-relational data. The proposed model can be applied to entities with structured properties such as text by utilizing the numerical vector representations of those properties. To address the sparsity problem of implicit feedback relations, the model is optimized via a sparsely-regularized multi-relational pair-wise Bayesian personalized ranking loss (BPR). Experiments on four different real-world datasets show that the proposed model significantly outperforms state-of-the-art models for multi-relational classification.

CCS CONCEPTS

• **Computing methodologies → Artificial intelligence**; *Learning latent representations*; *Multi-task learning*; • **Information systems → *Data mining***.

KEYWORDS

Multi-Relational Classification; Multi-Relational Learning; Documents Classification; Network Representation

ACM Reference Format:
Ahmed Rashed, Josif Grabocka, and Lars Schmidt-Thieme. 2019. Multi-Relational Classification via Bayesian Ranked Non-Linear Embeddings. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330863

KDD ’19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330863

1 INTRODUCTION

Multi-relational classification is a widely studied task in the domains of document classification[17], web-page classification [21], recommender systems [5, 8, 10], protein-label predictions [5], and network analysis.

There are multiple approaches for classifying entities within multi-relational settings. The typical approach is to extract a set of per-entity engineered features which are then fed directly to off-the-shelf classifiers. While certain accuracy can be achieved with engineered features, these approaches suffer from two main limitations. Firstly, they need expert domain knowledge to extract, preprocess and engineer features from raw data. Secondly, to extract expressive features, a fair amount of per-entity information is needed such as user profiles in social networks, however, this information might not be available in real-world scenarios.

On the other hand, there exist a set of models that are able to overcome such limitations by directly learning to extract per-entity latent features through mining the whole relations graph [2]. A well-known approach to extract such latent features is the multi-relational matrix factorization [7, 8, 16] which represents every relation in the graph as a matrix that can be factorized into two separate smaller matrices representing the latent features of the interacting entities. Although these approaches can be generalized to most multi-relational settings, they still face two significant challenges. The first challenge are multi-relational settings with implicit feedback relations. This kind of relations is usually very sparse and contains positive-only observed information while the unobserved information is a mixture of negative feedback (no relation exists) and missing values (relation might exists but is missing). Such relations are more common in real-world scenarios as they are easier to be extracted and tracked automatically such as citation references in citation networks or monitoring clicks in social networks. In such cases, treating the observed and unobserved edges as binary values will lead to poor latent features per-entity because the model will assume all unobserved relations to be negative information. The second challenge are relations with entities that have rich structured properties such as text in the case of documents. In such scenarios, models will need to be modified ad-hoc to utilize those structured properties while generating the latent features, otherwise, they risk losing important information that might affect the expressiveness of the output features.

In this paper, we address both challenges of multi-relational classification simultaneously by introducing (BRNLE), a flexible and scalable non-linear embedding model for classifying entities within diverse multi-relation settings. BRNLE is optimized via a multi-relational pair-wise Bayesian personalized ranking loss (BPR) with sparse regularization updates to overcome the drawbacks of sparse implicit relations. The proposed model can also be applied to entities with structured data modalities such as text by utilizing the numerical vector representations of those properties.

Our contributions can be summarized as follows :

- We introduce a simple, yet effective and flexible multi-relational classification model (BRNLE) that can be applied to diverse multi-relational settings where entities might have structured properties such as text and it has the potential to be extended for other complex structured properties.
- We utilize a sparsely regularized multi-relational Bayesian ranked loss for training the BRNLE to overcome the challenges of sparse implicit feedback relations.
- We conduct multiple experiments on four real-world datasets from diverse domains. The results show that the proposed BRNLE model outperforms state-of-the-art models in multi-relational classification and achieves improvements of up to 18.3% on accuracy and Micro F1. It also outperforms the best state-of-the-art models on two datasets with 40% less training data.

The rest of the paper is organized as follows. In Section 2, we summarize the related work. We discuss the problem formulation of the multi-relational classification task in Section 3. In Section 4, we present and discuss the technical details of the BRNLE model. We present the experimental results in Section 5. Finally, we conclude with discussing possible future work in Section 6.

2 RELATED WORK

Current approaches for multi-relational classification rely on learning per-entity latent features by mining the relations graph structure and analyzing entity interactions within the graph. An earlier approach proposed by Džeroski et al. [3] relied on relational decision trees and it was succeeded by more recent approaches that rely on learning entities' latent dimensions [18, 19]. These approaches produce a K-dimensional latent features for each entity by using either the first K-eigenvectors of a modularity matrix that was generated for the friendship relation [18] or a sparse k-means clustering of friendship edges [19]. After the extraction of the K-dimensional latent features, they are fed into a supervised SVM classifier for predicting their labels.

Recently, unsupervised [5, 10, 21, 23, 23] and semi-supervised [6, 20, 22] approaches have been proposed for multi-relational classification. Most of these approaches are inspired by the recent breakthroughs in the domain of natural language processing for learning word latent representations using convolutional neural networks and the Skip-gram model [9]. The Skip-gram model was firstly used in DeepWalk [10] which was extended later to handle entities with text properties [21, 23] and was improved by Node2Vec [5] for better random walk generations. Recently, Qiu et al. proposed a unified matrix factorization approach called NetMF [11]. NetMF incorporates aspects from multiple Skip-gram models and was able to achieve the current state-of-the-art performance on graph embeddings, however, it cannot utilize any structured properties that may be embedded in the graph entities such as text features. Skip-gram approaches learn the latent features by casting the multi-relation classification problem as a words classification problem where relations are treated as documents and entities as a sequence of words. The Skip-gram model will then classify an entity based on its similarity to other labeled entities.

On the other hand, semi-supervised approaches such as GCN [20] and GraphSAGE [6] were proposed to tackle the problem by relying on customized Graph Convolutional Networks (GCN) and pooling layers for learning latent entity features by utilizing the structured properties of their neighboring entities in the relations graph. Although those two approaches achieve state-of-the-art performance, they cannot be applied to graphs where entities lack any structured properties which is their major limitation.

Although, these recent unsupervised and semi-supervised models achieve competitive accuracy, however, they still suffer from two major drawbacks, Firstly, they underperform when applied to sparse multi-relational settings. Secondly, some models cannot represent entities with structured properties which leads to poor classification accuracy while others mainly rely on entity properties which are not always available in real cases. Most graph embedding models are also naturally limited to two-relational settings (Entity-to-Entity and Entity-to-Labels) and cannot be extended for an arbitrary number of relations with an exception of multi-relational factorization models.

To overcome the sparsity problem and the limited number of supported relations, Krohn-Grimberghe et al. [8] proposed (MR-BPR) as a learning to rank approach for multi-relational classification by extending the original BPR model [13] for sparse multi-relational settings. This approach formulates the problem as a multi-relational matrix factorization trained to optimize the AUC measure using BPR loss. Each available relation is represented by a matrix and the relation between target entities and labels will be the target to be predicted. Although this model is suitable for any number of sparse relations, it still cannot represent entities with structured properties.

In this paper, we aim at addressing those two major drawbacks simultaneously by proposing the BRNLE model for multi-relational classification. BRNLE is capable of accommodating any number of relations by utilizing a separate embedding function for each available entity type, and a separate scoring function for each available relation. However, to compare the model performance against other baselines, we only used two-relations settings in this work. We also utilized a multi-relational pair-wise Bayesian personalized ranking loss with controlled sparse regularization updates to overcome the drawbacks of sparse relations. Regarding the entities with structured properties such as text, BRNLE can directly utilize the numerical vector representations of those properties without any extra ad-hoc modifications.

3 PROBLEM DEFINITION

In multi-relational settings [8], there exists a set of entity types $\mathcal{E} := \{E_1, E_2, ..., E_{|\mathcal{E}|}\}$, where each type contains a set of instances $E_k := \{e_k^{(1)}, e_k^{(2)}, ..., e_k^{(|E_k|)}\}$. An entity instance e_k is represented as a one-hot encoding vector of size $|E_k|$ and it might have some embedded structured properties s_{e_k} that can be represented as numerical vectors such as the case of text. There exists also a set of binary relations $\mathcal{R} := \{R_1, R_2, ..., R_{|\mathcal{R}|}\}$ where each relation R_r

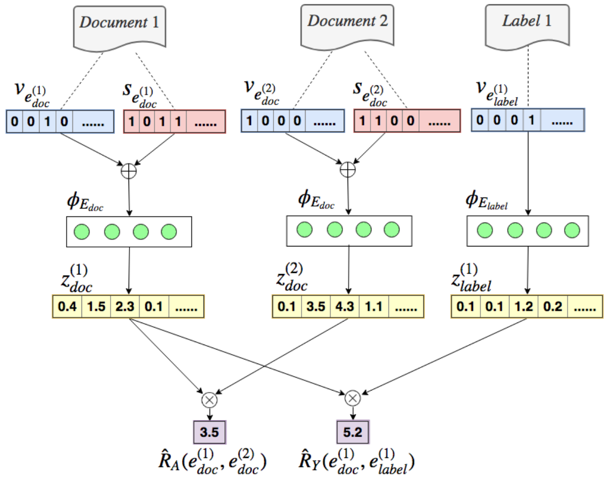


Figure 1: BRNLE architecture for documents citation graphs. Initially for all document entities e_{doc}, a one-hot encoding vector $v_{e_{doc}}$ and a numerical vector $s_{e_{doc}}$ representing its structured properties are generated. These two vectors are concatenated and fed to a per-entity type embedding function $\phi_{E_{doc}}$ which is learned to generate latent features z_{doc}. Same procedure is applied to the class label entities e_{label} but without concatenating any structured properties. Finally, two scoring functions $\hat{R}_Y$ and $\hat{R}_A$ are learned jointly to predict the relations scores for the target relation $Y \subseteq E_{doc} \times E_{label}$ and the auxiliary relation $A \subseteq E_{doc} \times E_{doc}$ by utilizing the learned entities latent features

represents the interaction between two entity types $E_{r,1}$ and $E_{r,2}$ such that $R_r \subseteq E_{r,1} \times E_{r,2}$.

The primary goal of multi-relational classification tasks is to predict missing edges in a target relation $Y \subseteq E_p \times E_{labels}$ that represents the relation between the entities to be classified E_p and the class labels E_{labels}. Other relations are considered auxiliary information $\mathcal{A} := \mathcal{R} \setminus \{Y\}$ that might be used in the prediction task.

Since each entity might have multi-class labels, we formulate the classification problem as a ranking problem by deriving a ranked list of labels for each entity sorted according to the likelihood that the entity belongs to each one of them. To achieve this, a scoring function $\hat{R} : E_p \times E_{labels} \rightarrow \mathbb{R}$ is learned to predict the scores of all labels with respect to the predictor entities.

4 PROPOSED MODEL

The proposed BRNLE model is composed of two main components. The first component contains the non-linear embeddings which are responsible for extracting latent features for every entity based on all of its observed relations and its embedded structured properties. The second component is the Bayesian personalized ranking function which is responsible for utilizing the generated entities latent features for deriving the ranked list of labels. Figure 1, illustrates the architecture of the BRNLE model and each component will be discussed in details in the following subsections.

4.1 Non-Linear Embeddings

Given the set of entity types $\mathcal{E}$, we define a set of latent entities embeddings $\mathcal{Z} := \{Z_1, Z_2, ..., Z_{|\mathcal{E}|}\}$ and a per-entity type latent embedding function ϕ_{E_k} to extract the latent features vectors $Z_k := \{z_k^{(1)}, z_k^{(2)}, ..., z_k^{(|E_k|)}\}$ from all instances that belong to the entity type E_k as follows:

$$z_k = \phi_{E_k}(x_{e_k}; \theta_{\phi_k}) \tag{1}$$

$$x_{e_k} = [v_{e_k}, s_{e_k}] \tag{2}$$

where z_k are the extracted latent feature vectors for entity instance e_k. ϕ_{E_k} is a series of non-linear fully connected layers with network parameters θ_{ϕ_k}. x_{e_k} is a concatenation between a one-hot encoded vector v_{e_k} of instance e_k and the numerical vector representing its structured properties s_{e_k}. s_{e_k} is a structure-dependent variable which can be omitted if no structured properties exist for the entity, e.g. in document classification v_{e_k} will be the document id and s_{e_k} the binary/tf-idf vector representation of the document text. In this paper's scope, we address only binary vector representations of words as structured properties, however, s_{e_k} can be extended to other types. We also utilize a single-layer embedding function in most of our comparative studies because its capacity was found to be expressive enough for the utilized datasets. Comparison between different layered architectures for ϕ is discussed in Section 5.2.1.

In order to extract expressive latent features, one needs to carefully select the activation functions for ϕ. In case of single-layer embedding function, the activation function will be applied directly to the output of that layer. After an experimental study with different non-linear activations, we select CReLU [15] activation which is activated in both positive and negative direction while maintaining the same degree of non-saturated non-linearity similar to ReLU activation. Results of this study are discussed in Section 5.1.

4.2 Bayesian Personalized Ranking

For all binary relations $R_r \subseteq E_{r,1} \times E_{r,2}$, we define a scoring function $\hat{R}_r \rightarrow \mathbb{R}$ that utilizes the latent features vectors pairs $z^{(i)}_{r,1}$, $z^{(j)}_{r,2}$ of all available instances pairs $e^{(i)}_{r,1}$ and $e^{(j)}_{r,2}$ respectively.

$$\hat{R}_r(e^{(i)}_{r,1}, e^{(j)}_{r,2}) := {z^{(i)}_{r,1}}^T z^{(j)}_{r,2} \tag{3}$$

$\hat{R}_r$ can either be a direct dot product between the latent features vectors or a parameterized function $g_{\hat{R}_r}(z^{(i)}_{r,1}, z^{(j)}_{r,2}; \theta_{g_{R_r}})$. We only utilized the dot product version in all comparative experiments of the BRNLE model because it achieved higher accuracy against multiple parameterized versions. Experiments with different choices of $\hat{R}_r$ are discussed in Section 5.2.2.

To learn the parameters of the embedding function ϕ and the scoring function g, we need a suitable loss function that is immune to the sparsity of implicit feedback relations. To do so, we utilized the Bayesian personalized ranking loss to train the BRNLE model which was inspired by [8, 13]. In a single relation setting of user-item interactions, BPR makes use of the assumption that, for a given user u, any item i he interacted with (observed), should be ranked higher than any item j he didn't interact with (unobserved). To achieve this, the BPR loss aims to maximize the difference $\hat{d}^{R_r}_{u,i,j}$ between the predicted ratings scores $\hat{R}_r(u, i)$ and $\hat{R}_r(u, j)$.

$$\hat{d}^{R_r}_{u,i,j} = \hat{R}_r(u,i) - \hat{R}_r(u,j) \tag{4}$$

To follow the same notations in multi-relational settings, for any relation R_r, the user u will represent an entity of type $E_{r,1}$, while i and j will represent two entities of type $E_{r,2}$. During the training phase, a set of triples $D_{R_r} := \{(u,i,j) | (u,i) \in R_r \wedge (u,j) \notin R_r\}$ will be sampled using bootstrap sampling with replacement.

BRNLE is then optimized by maximizing the overall objective function $J(\Theta)$ which is the sum of all differences between the predicted rating scores of the observed and unobserved edges for all relations $\mathcal{R}$. The maximization is done by applying a stochastic gradient ascent on Equations (5) and (6).

$$J(\Theta) = \sum_{R_r \in \mathcal{R}} \sum_{(u,i,j) \in R_r} J(R_r, u, i, j) \tag{5}$$

$$J(R_r, u, i, j) = \alpha_{R_r} \ln \sigma(\hat{d}^{R_r}_{u,i,j}) - \text{Reg}(E_{r,1}, E_{r,2}, u, i, j) \tag{6}$$

where σ is the logistic function, α_{R_r} is the weight of relation R_r, and Θ is a set of all model parameters. $\text{Reg}(E_{r,1}, E_{r,2}, u, i, j)$ is a sparse L2 regularization term which will be discussed in details in Section 4.3.

Finally, in the inference phase, we only use the predicted rating scores of the target relation $Y \subseteq E_p \times E_{labels}$ as the classification output. The full pseudocode for BRNLE is described in Figure 2.

```
procedure LEARNBRNLE($\mathcal{D}, \mathcal{R}, \mathcal{E}$)
    Initialize $\Theta := \{\theta_{\phi_k}, \forall k \in \{1, ..., |\mathcal{E}|\}\}$
    repeat
        for $R_r \in \mathcal{R}$ do
            draw (u,i,j) from $D_{R_r} \in \mathcal{D}$
            $z^{(u)}_{r,1} \leftarrow \phi_{E_{r,1}}(x_u; \theta_{\phi_{r,1}})$                ▷ Eq.(1)
            $z^{(i)}_{r,2} \leftarrow \phi_{E_{r,2}}(x_i; \theta_{\phi_{r,2}})$
            $z^{(j)}_{r,2} \leftarrow \phi_{E_{r,2}}(x_j; \theta_{\phi_{r,2}})$
            $\hat{d}^{R_r}_{u,i,j} \leftarrow z^{(u)^T}_{r,1} z^{(i)}_{r,2} - z^{(u)^T}_{r,1} z^{(j)}_{r,2})$    ▷ Eq.(3&4)
            $\theta_{\phi_{r,1}} \leftarrow \theta_{\phi_{r,1}} + \mu \frac{\partial J(R_r,u,i,j))}{\partial \theta_{\phi_{r,1}}}$    ▷ Eq.(5&6)
            $\theta_{\phi_{r,2}} \leftarrow \theta_{\phi_{r,2}} + \mu \frac{\partial J(R_r,u,i,j))}{\partial \theta_{\phi_{r,2}}}$    ▷ Eq.(5&6)
        end for
    until convergence
return $\Theta$
end procedure
```

Figure 2: BRNLE pseudocode

4.3 Sparse Regularization Updates

In sparse multi-relation settings with implicit feedback relations, the input vector x_{e_k} of the embedding function ϕ will be very sparse which means most of the first-layer weights will be multiplied by zeros and they will not contribute to the resulting latent feature values. If we apply normal L2 regularization on all θ_{ϕ_k} weights, the gradient of the objective function will be zero for all first-layer weights that were multiplied by zero input. However, the gradient of the L2 regularization component will always be a positive value, hence all first-layers weights will be penalized by the gradient of the L2 regularization component, regardless of whether they contributed to the resulting latent feature or not. This means that weights will be repeatedly penalized over all iterations, although, in reality, they might have been used only a few times. To avoid such unnecessary regularization gradient updates to the weights θ_{ϕ_k}, we define a sparse L2 regularization term as follows:

$$\begin{aligned}\text{Reg}(E_{r,1}, E_{r,2}, u, i, j) =& \lambda_{E_{r,1}} \phi_{E_{r,1}}(u)^2 + \lambda_{E_{r,2}} \phi_{E_{r,2}}(i)^2 \\ &+ \lambda_{E_{r,2}} \phi_{E_{r,2}}(j)^2\end{aligned} \tag{7}$$

where $\text{Reg}(E_{r,1}, E_{r,2}, u, i, j)$ is a sparse L2 (activity) regularization term [4] applied to the latent features output of the embedding function ϕ instead of applying it on all weights θ_{ϕ_k}. $\lambda_{E_{r,1}}$ and $\lambda_{E_{r,2}}$ represent the sparse regularization weights for entity types $E_{r,1}$ and $E_{r,2}$. This term will allow the regularization gradients to propagate back only through the weights that were activated by the entity instance and contributed to the resulting latent feature.

5 EXPERIMENTS

In this section, multiple experiments were conducted to evaluate and find the best architecture for BRNLE. These experiments aim to answer the following research questions:

RQ1 Are non-linear activation functions helpful for learning rich latent representations?

RQ2 How many layers of hidden units are needed for BRNLE to learn and score entity relations?

RQ3 How well does BRNLE perform in comparison with other state-of-the-art models for multi-relational settings that contain rich entity features?

RQ4 How well does BRNLE perform in comparison with other state-of-the-art models for multi-relational settings that lack any entity features?

RQ5 What are the effects of BRNLE individual components on the classification accuracy?

Table 1 shows the detailed statistics of all datasets that were used in our experiments. In the following subsections, we answer the research questions by presenting the results of their related experiments.

5.1 Linear vs Non-linear Activations in Latent Embeddings (RQ1)

Figure 3 shows a performance comparison on Cora [14] using a 10% of labeled nodes for training. This comparison was done between different versions of the single-layer embedding function ϕ using Linear, CReLU [15] and ReLU activation functions. Results show that the CReLU and Linear versions have higher accuracies than ReLU which indicates that truncating the negative values has an adverse effect on the expensiveness of the output features. The results also show that the CReLU version has higher accuracy than its linear counterpart due to its non-linearity which allows it to extract more expressive features. Finally, in comparison to other well-known models, BRNLE with CReLU is able to outperform all of them, while the Linear counterpart is on par with the best

Table 1: Datasets Statistics

	Entity Features	Relations	Dimensions	# of observations	Sparsity
Email-Eu-core	-	Target	1005x42	1005	97.61%
		Auxiliary	1005x1005	25571	97.46%
PPI	-	Target	3890x50	6640	96.58%
		Auxiliary	3890x3890	76584	99.49%
Cora	1433	Target	2708x7	2708	85.71%
		Auxiliary	2708x2708	5429	99.92%
Citeseer	3703	Target	3312x6	3312	83.33%
		Auxiliary	3312x3312	4732	99.95%

GraphSAGE variant [6]. The BRNLE hyper-parameters used in this experiment are discussed in details in Section 5.6.

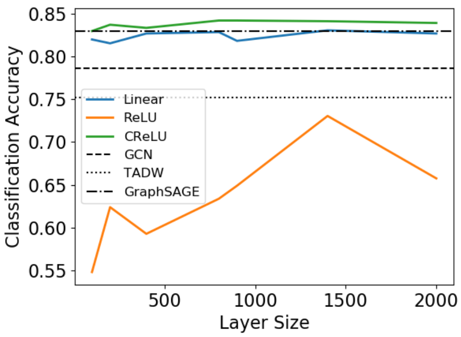
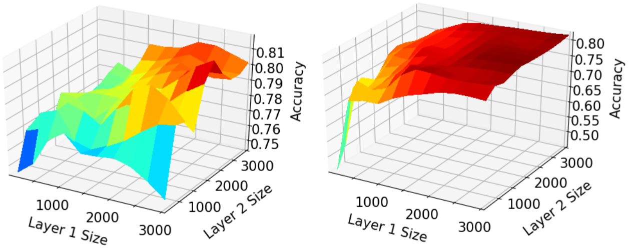


Figure 3: Performance comparison on the Cora dataset with 10% labeled nodes between Linear, CReLU and ReLU activations for the embedding function ϕ with a single layer.

5.2 Model Structure and Complexity (RQ2)

In order to find the best architecture for the embedding and scoring functions, two experiments were conducted on Cora dataset [14] using 10% labeled nodes for training.

5.2.1 Embedding Function. In order to find the best number of layers for the embedding function ϕ, we studied the performances of different combinations of two-layered and three-layered architectures with different sizes of hidden units per-layer ranging from 100 to 3000 units. Performances of different two-layered versions are as shown in Figures 4(a) and 4(b). We also compared the training objective value and accuracy of the best two-layered version (3000 hidden units per-layer) against the best single-layered version (900 hidden units) and three-layered version (2000 hidden units per-layer) as shown in Figures 5(a) and 5(b).

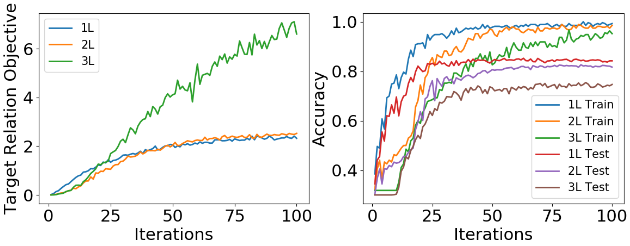


(a) Without L2 regularization **(b) With L2 regularization**

Figure 4: Comparison between several two-layered architectures of the embedding function ϕ. Figure (a) shows the results without sparse L2 regularization while (b) show results with sparse L2 regularization

Figure 4(b) shows that with controlled sparse regularization updates, increasing the size of the first layer is sufficient to capture most important information in the given dataset and adding an extra layer provides no significant lift. On the other hand, Figures 5(a) and 5(b), show that all models are capable of achieving the same training accuracy with a fixed regularization weights, however, the two-layered and three-layered versions start overfitting after 50 and 25 iterations. The results also show that the single-layered version has sufficient capacity to express the entities in the given dataset and no further layers are required.

According to those findings, we employed a single-layered embedding function ϕ in all of our comparative experiments.



(a) Train Objective **(b) Train and Test Accuracy**

Figure 5: Performance comparison between the best two-layered and three-layered versions of the embedding function ϕ against the best single-layered one.

5.2.2 Scoring Function. To select the best scoring function $\hat{R}_r$ for the BRNLE model, we compared the performance of the dot product version against two parametric versions as shown in Table 2.

The first version relies on concatenating the entity features before feeding them to two fully connected layers that predict the final relation score. The second version relies on multiplying the entity features element-wise to maintain their spatial alignment before feeding them to the fully connected layers. Best results were achieved by using 900-dimensional latent embeddings for the dot product versions, 400 for the multiplication version and 1000 for the concatenation version. We also tried different sizes for the first fully connected layer of the scoring function. The final fully connected layer has only one output linear unit to predict the relation score. Table 2 shows that the dot product version outperforms various parametric versions with different settings.

Table 2: Comparison between different choices for the scoring function on Cora dataset with a 10% labeled nodes for training. There are two ways to align the entities features which are either by concatenating them or multiplying them element-wise before they are fed to the scoring function.

Function Type	Feature Alignment	Layer Size	Accuracy
Dot Product	-	-	**83.79 %**
Parametric	Concatenated	1000	64.52 %
		800	66.32 %
		200	65.66 %
Parametric	Multiplied	1000	71.41 %
		800	70.79 %
		200	70.01 %

5.3 Comparison with state-of-the-art models for multi-relational classification with entity features (RQ3)

5.3.1 Datasets. In order to compare the performance of BRNLE against other state-of-the-art models that utilize entity features, it was applied on the following two well-known multi-relational classification datasets that contain entity features.

(1) Cora [14]: A citation network where each document has 1433 binary features representing word occurrences. It has two relations, a target relation which represents the class label of a document, and an auxiliary relation that represents citation links between documents.
(2) Citeseer [14]: Another citation network where each document has 3703 binary features representing word occurrences. It has two relations, a target relation which represents the class label of a document, and an auxiliary relation that represents citation links between documents.

5.3.2 Baselines.

(1) MR-BPR [8]: A multi-relational matrix factorization model optimized using the BPR loss for sparse multi-relational classification. This model is considered equivalent to a basic version of BRNLE without the non-linearity and the support for entity features.
(2) MMDW [21]: State-of-art model for multi-relational classification. It is also considered an extended version of DeepWalk using max-margin SVM classifier for entities with text structured properties.
(3) TADW [23]: Well-known model for document classification in citation networks. It is considered an extended version of DeepWalk using low-rank matrix factorization for entities with text structured properties.
(4) GCN [20]: State-of-art model for document classification by learning graph representations. It utilizes a multi-layered graph convolutional neural network for learning vertex representations with text features.
(5) GraphSAGE [6]: Current state-of-the-art for graph embedding that leverages node feature information (e.g., text attributes) to efficiently generate node embeddings. We used the GCN and mean aggregated variants of GraphSAGE for comparison with BRNLE.

In order to measure the performance of MR-BPR on datasets with text features, we had to use a workaround by defining the relation between documents and their binary words vectors as a separate third relation.

5.3.3 Experimental Protocol. For our experimental protocol, we followed a similar scale-up evaluation approach as [5, 8, 10]. We applied a 5-fold cross-validation using different percentages of labeled nodes for training which are 10%, 50% and 90% respectively. In each experiment, we only utilized the defined percentage of labeled nodes for training along with all the auxiliary relations, while the remaining percent of nodes were used for testing. To follow the same evaluation metrics as the original papers, we used the accuracy measure for Cora and Citeseer.

We used the same hyper-parameters that were used in the original baselines' papers, and grid-search was used to find the best hyper-parameters if none were mentioned. The optimal hyper-parameters for BRNLE have been estimated via grid search on the 10% split of each respective dataset. Hyper-parameter details for all baselines and BRNLE are discussed in Section 5.6.

Table 3: Accuracy(%) of document classification on Cora.

%Labeled Nodes	10%	50%	90%
MR-BPR [8]	75.03	78.76	81.66
MMDW [21]	74.94	84.71	88.19
TADW [23]	75.24	85.99	85.60
GCN [20]	78.37	86.53	86.39
GraphSAGE-GCN [6]	83.31	86.79	86.93
GraphSAGE-mean [6]	82.23	86.98	87.89
BRNLE (Our model)	**83.79**	**87.35**	**90.54**

5.3.4 Results. Tables 3 and 4 show the single-label classification accuracies with different training split ratios on Cora and Citeseer datasets.

Results show that BRNLE consistently and significantly outperforms state-of-the-art models that rely on entity features for multi-relational classification. All BRNLE improvements over the

Table 4: Accuracy(%) of document classification on Citeseer.

%Labeled Nodes	10%	50%	90%
MR-BPR [8]	43.91	45.36	48.70
MMDW [21]	55.60	66.93	70.95
TADW [23]	67.48	72.90	70.80
GCN [20]	71.59	76.14	77.42
GraphSAGE-GCN [6]	70.66	71.59	75.34
GraphSAGE-mean [6]	71.82	73.87	76.61
BRNLE (Our model)	**73.58**	**77.53**	**80.01**

baseline models are statistically significant with a p-value less than 0.01 using a paired t-test except only for the gain of BRNLE over the GraphSAGE variants on the 10% and 50% splits of the Cora dataset where the p values are between 0.3 and 0.1. BRNLE achieves around 2.5%, 3.5% improvements over the best baseline on Citeseer when ratios are 10% and 90%; It is worth to note that BRNLE outperforms the best baseline with 40% less data on Citeseer. On Cora, GraphSAGE had a close competitive performance to BRNLE but the proposed model significantly outperformed it after increasing the number of training instances to 90%.

5.4 Comparison with state-of-the-art models for multi-relational classification without entity features (RQ4)

5.4.1 Datasets. In order to compare the performance of BRNLE against other state-of-the-art models that don't utilize any entity features, it was applied on the following two well-known multi-relational classification datasets from two different domains.

(1) Email-Eu-core [24]: A network of emails from a large European research institution. It has two relations, a target one which represents the relation between persons and the departments of the research institute, and an auxiliary relation that represents email communications between institution members. Entities in this dataset have no structured properties.
(2) Protein-Protein Interactions (PPI) [1]: Protein-protein interactions graph for homo sapiens. It has two relations, a target one which represents the relation between protein-labels and proteins, and an auxiliary relation that represents the interactions between proteins. This dataset has no structured properties.

5.4.2 Baselines.

(1) MR-BPR [8]: A multi-relational matrix factorization model optimized by using BPR loss for sparse multi-relational classification. This model was also used in Section 5.3.
(2) DeepWalk [10]: Well-known model for multi-relational classification and learning network representations. It utilizes random walks and Skip-Gram models to learn the vertex latent features.
(3) Node2Vec [5]: Well-known unsupervised graph embedding model for multi-relational classification and usually considered as an improved version of DeepWalk with two guiding directional parameters p and q.
(4) SDNE [22]: Well-known model for multi-relational classification that utilizes deep networks to extract nonlinear entity information from the graph structure.
(5) NetMF [11]: Current state-of-the-art unified matrix factorization approach for multi-relational classification.

5.4.3 Experimental Protocol. We followed the same experimental protocol discussed in Section 5.3. To follow the same evaluation metrics as the original papers, we used Micro-F1 and Macro-F1 measures for performance evaluation on PPI as it has multi-labeled entities and we used the accuracy measure for Email-Eu-core.

Table 5: Accuracy(%) of multi-relational classification on Email-Eu-core.

%Labeled Nodes	10%	50%	90%
MR-BPR [8]	60.15	74.93	74.80
DeepWalk [10]	57.31	71.53	73.27
Node2Vec [5]	57.81	72.49	77.22
SDNE [22]	51.56	64.53	70.70
NetMF [11]	45.59	60.00	63.17
BRNLE (Our model)	**65.76**	**76.94**	**82.20**

Table 6: Micro and Macro F1 scores of multi-relational classification on PPI.

	%Lable Nodes	10%	50%	90%
Micro-F1(%)	MR-BPR [8]	17.11	22.61	23.44
	DeepWalk [10]	15.89	20.85	24.11
	Node2Vec [5]	15.09	20.38	22.65
	SDNE [22]	16.11	20.56	23.23
	NetMF [11]	17.93	23.23	23.26
	BRNLE (Our model)	**19.63**	**25.20**	**28.51**
Macro-F1(%)	MR-BPR [8]	12.88	18.81	19.48
	DeepWalk [10]	12.73	18.50	19.15
	Node2Vec [5]	12.17	18.01	18.76
	SDNE [22]	12.67	16.88	19.20
	NetMF [11]	13.97	20.21	19.98
	BRNLE (Our model)	**15.27**	**21.22**	**23.86**

5.4.4 Results. Tables 5 and 6 show the single-label and multi-label classification accuracy using different training split ratios on Email-Eu-core and PPI datasets.

Results show that BRNLE significantly outperforms state-of-the-art models that don't rely on entity features for multi-relational classification. All BRNLE improvements over the baseline models are also statistically significant with a p-value less than 0.01 using a paired t-test. BRNLE achieves around 9.4%, 6.5% improvements on Email-Eu-core when ratios are 10% and 90%; and around 9.6%, 8.5%, and 18.3% improvements over the best baseline on PPI when ratios are 10%, 50% and 90%. It is worth to note that BRNLE also outperforms the best baseline with 40% less data on PPI similar to the Citeseer dataset.

Another interesting finding was that the MR-BPR model introduced in 2012, has a very competitive performance with the most recent baselines on PPI and Email-Eu-core. However, it was never mentioned in the more recent papers.

5.5 Ablation Study (RQ5)

In this section, different configurations of BRNLE were compared to study the effects of each individual model component on the classification accuracy. This experiment was applied on Citeseer dataset using 10% of labeled nodes. The following BRNLE configurations were tested:

(1) With linear activation
(2) With linear activation and sparse L2 regularization
(3) With linear activation and entity features
(4) With linear activation, sparse L2 regularization and entity features
(5) With CRelu activation
(6) With CRelu activation and sparse L2 regularization
(7) With CRelu activation and entity features
(8) With CRelu activation, sparse L2 regularization and entity features

It is worth mentioning that BRNLE configuration (1) is equivalent to the basic version of matrix factorization model MR-BPR [8].

Table 7: Accuracy(%) of different BRNLE configuration on Citeseer.

%Labeled Nodes	10%
(1) Linear Act.	43.72
(2) Linear Act. + L2 Reg.	46.30
(3) Linear Act. + Entity Features	70.70
(4) Linear Act. + L2 Reg. + Entity Features	72.30
(5) CRelu Act.	47.18
(6) CRelu Act. + L2 Reg.	48.45
(7) CRelu Act. + Entity Features	71.51
(8) CRelu Act. + L2 Reg. + Entity Features	73.58

The results in Tables 7 show that the L2 component alone improves the accuracy by 1% to 2.5% regardless of the activation function. The improvements induced by the non-linearity alone are ranging from 1% to 3% without any regularization. Combining the non-linear activation function and L2 regularization in configuration (6) achieves an improvement of 4.7% over the basic configuration (1). Entity features on the other hand, have the highest effect on the accuracy which was an expected finding and the achieved improvements ranging from 51% to 65% Finally, the best configuration (8) was found to achieve an improvement of 68% over the basic configuration (1).

5.6 Reproducibility of the Experiments

For BRNLE, we tested the embedding dimensions ranging from 100 to 1000, the learning rate of [0.1, 0.05, 0.02, 0.01, 0.005, 0.002, 0.001, 0.0005], the regularization weights of [0.1, 0.05, 0.015, 0.0125, 0.01, 0.005, 0.0005, 0.0001] and the number of iterations ranging from 100 to 1000. We fixed the relations' weight α to 0.5 for all datasets to allow equal contribution. The best BRNLE hyper-parameters are shown in Table 8 and the following hyper-parameter settings were used for the baseline models.

Table 8: BRNLE hyper-parameters settings

Dataset	Embedding size	μ	λ_{entity}	λ_{label}	Iterations
Email	700	0.05	0.005	0.0005	1000
PPI	500	0.02	0.0125	0.0005	500
Cora	900	0.005	0.05	0.005	200
Citeseer	1000	0.005	0.005	0.0005	100

- MR-BPR [8]: We tested embedding dimensions ranging from 100 to 1000, learn rate of [0.1, 0.05, 0.04, 0.03, 0.02, 0.01, 0.005, 0.001] and regularization weights of [0.1, 0.05, 0.015, 0.0125, 0.01, 0.005, 0.0005, 0.0001, 0.00005]. The best hyper-parameters were $k = 500$, $\mu = 0.01$, $\lambda_{protein} = 0.0125$, $\lambda_{label} = 0.0005$, 400 iterations and $\alpha = 0.5$ for PPI; $k = 300$, $\mu = 0.05$, $\lambda_{user} = 0.005$, $\lambda_{label} = 0.0005$, 400 iterations and $\alpha = 0.5$ for Email-Eu-core; $k = 900$, $\mu = 0.03$, $\lambda_{document} = 0.005$, $\lambda_{label} = 0.0001$, $\lambda_{words} = 0.0001$, 1400 iterations and $\alpha = 0.33$ for Cora; and $k = 1000$, $\mu = 0.05$, $\lambda_{document} = 0.05$, $\lambda_{label} = 0.005$, $\lambda_{words} = 0.005$, 1000 iterations and $\alpha = 0.33$ for Citeseer.
- DeepWalk [10]: The used hyper-parameters are $d = 128$, $r = 10$, $l = 80$ and $k = 10$ for PPI and Email-Eu-core similar to the original paper.
- Node2Vec [5]: For p and q, we tested different values of [0.25, 1, 2, 4], while for the rest of the hyper-parameters, we used the original paper values. The best found parameters were $d = 128$, $r = 10$, $l = 80$, $k = 10$, $p = 4$ and $q = 1$ for PPI and Email-Eu-core.
- SDNE [22]: We tested layer sizes of [32, 64, 128, 256, 512, 1024] and learn rate of [0.1, 0.05, 0.01, 0.005, 0.001]. The best hyper-parameters we found were layer sizes = [256, 256] and the learn rate = 0.005 for PPI; layer sizes= [128, 128] and the learn rate = 0.01 for Email-Eu-core.
- NetMF [11]: We used the original paper hyber-parameters for PPI which are $h = 256$ and embedding dimension of size 128. For Email-Eu-core, we select 128 embedding dimension and $h = 256$ as well after testing different dimensions of [32, 64, 128, 256, 512].
- MMDW [21]: In this paper, we directly report the original paper results on Cora and Citeseer using their best version ($\eta = 10^{-2}$).
- TADW [23]:The same hyper-parameters from the original paper were used which are $k = 80$ and $\lambda = 0.2$ for Cora and Citeseer.
- GCN [20]: Same default hyper-parameters from the original paper were used which are dropout rate = 0.5, L2 regularization = 5.10^{-4}, 16 (number of hidden units), 200 epochs for Cora and Citeseer.
- GraphSAGE [6]: Hyper-parameters where set using grid search as follows, $S1 = 5$, $S2 = 5$, learn rate= 0.7, and embedding

size = 128 for Cora and Citeseer. We tested $S1$ and $S2$ values of [1, 5, 10, 15], Learn rate of [0.9, 0.7, 0.5, 0.2, 0.005] and embedding sizes of [64, 128, 256, 512].

The code for MR-BPR is available at the author’s home page[1]. The code for DeepWalk, Node2Vec, SDNE, GCN, TADW is available at the open source toolkit for network embedding (OpenNE)[2]. The code for NetMF is available at the authors’ GitHub repository[3]. The code for GraphSAGE-GCN and GraphSAGE-mean is available at the authors’ GitHub repository[4].

6 CONCLUSION

In this paper, we propose Bayesian ranked non-linear embeddings (BRNLE), a salable multi-relational classification model that can utilize entities’ structured properties such as text. BRNLE is optimized via a multi-relational pair-wise Bayesian personalized ranking loss (BPR) with sparse regularization updates to extract rich latent features that are capable of capturing entities’ interactions within sparse relations graph. Experimental results on four real-world datasets show that BRNLE outperforms state-of-the-art models in multi-relational classification with or without the existence of text structured properties.

In future works, we plan to explore further architectures for the parametric scoring and embedding functions, especially in the case of more complex structured properties such as images. We also plan to explore different approaches to improve the BRNLE model such as by using non-uniform samplers for the learning procedure [12] and adaptive sparse regularization.

ACKNOWLEDGMENTS

This work is co-funded by the industry project "Data-driven Mobility Services" of ISMLL and Volkswagen Financial Services.(https://www.ismll.uni-hildesheim.de/projekte/dna_en.html)

REFERENCES

[1] Bobby-Joe Breitkreutz, Chris Stark, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, Michael Livstone, Rose Oughtred, Daniel H Lackner, Jürg Bähler, Valerie Wood, et al. 2007. The BioGRID interaction database: 2008 update. *Nucleic acids research* 36, suppl_1 (2007), D637–D640.

[2] Hongyun Cai, Vincent W Zheng, and Kevin Chang. 2018. A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Transactions on Knowledge and Data Engineering* (2018).

[3] Sašo Džeroski. 2003. Multi-relational data mining: an introduction. *ACM SIGKDD Explorations Newsletter* 5, 1 (2003), 1–16.

[4] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research)*, Geoffrey Gordon, David Dunson, and Miroslav DudÃŋk (Eds.), Vol. 15. PMLR, Fort Lauderdale, FL, USA, 315–323. http://proceedings.mlr.press/v15/glorot11a.html

[5] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[6] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.

[7] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 135–142.

[8] Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2012. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 173–182.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[10] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[11] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 459–467.

[12] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 273–282.

[13] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[14] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93.

[15] Wenling Shang, Kihyuk Sohn, Diogo Almeida, and Honglak Lee. 2016. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *International Conference on Machine Learning*. 2217–2225.

[16] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 650–658.

[17] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1067–1077.

[18] Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 817–826.

[19] Lei Tang and Huan Liu. 2009. Scalable learning of collective behavior based on sparse social dimensions. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 1107–1116.

[20] N Kipf Thomas and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*.

[21] Cunchao Tu, Weicheng Zhang, Zhiyuan Liu, and Maosong Sun. 2016. Max-Margin DeepWalk: Discriminative Learning of Network Representation.. In *IJCAI*. 3889–3895.

[22] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1225–1234.

[23] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. 2015. Network representation learning with rich text information.. In *IJCAI*. 2111–2117.

[24] Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. 2017. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 555–564.

[1] https://www.ismll.uni-hildesheim.de/mymedialite/index.html

[2] https://github.com/thunlp/OpenNE

[3] https://github.com/xptree/NetMF

[4] https://github.com/williamleif/graphsage-simple