Artificial Intelligence Applications And Innovations Vol. 1 No. 1 (2024)

Article history:
Received 17 September 2023
Revised 29 November 2023
Accepted 05 December 2023
Published online 01 January 2024

Multi-view Knowledge Graph Embedding for Link Prediction by PLSA Method

Afrooz. Moradbeiky[1], Farzin. Yaghmaee[1*]

[1] Department of Electrical and Computer Engineering, Semnan University, Semnan, Iran

*** Corresponding author email address**: f_Yaghmaee@semnan.ac.ir

A r t i c l e I n f o

Article type:
Original Research

How to cite this article:
Moradbeiky, A., & Yaghmaee, F. (2024). Multi-view Knowledge Graph Embedding for Link Prediction by PLSA Method. Artificial Intelligence Applications and Innovations, *1*(1), 66-77
https://doi.org/10.61838/jaiai.1.1.5

A B S T R A C T

Reasoning refers to the extraction of information from a graph in an embedded space through link prediction. Link prediction involves identifying the missing edge between a pair of entities. While graph features are utilized to locate these missing edges, similarity-based methods often fail to consider all of the graph features. Multi-view methods leverage multiple perspectives, each offering a different aspect of data analysis. In this paper, we present a novel approach called the Multi-View Probabilistic Latent Semantic Analysis (PLSA) based PLSA algorithm. This algorithm calculates three distinct views, enabling a comprehensive analysis of the underlying data. The first one refers to the viewing probability of a node in the head, tail, or relation separately. The second view highlights the significance of a suggested tail in information representation, while the third view evaluates the quality of information flow within a set comprising the head, relation, and tail. These three views are combined to derive the optimal score function by PLSA method. Test results indicate that the proposed method ensures that the correct solution lies within an acceptable range, with a hit rate exceeding 40% in the Freebase dataset. Experimental results further demonstrate the effectiveness of the proposed algorithms compared to state-of-the-art methods.

Keywords: *knowledge graph embedding, multi view, page rank, information diffusion.*

1. Introduction

The knowledge graph (KG) serves as an abstract representation of stored data and encompasses knowledge about various aspects of the world, including regulations and relationships among entities [1]. The KG is structured as a directed graph, composed of triplets known as (Head-Relation-Tail) representations. While the KG may contain relation faults and not encompass all facts, artificial intelligence researchers have developed a strong interest in extracting and discovering new knowledge from the existing knowledge graph [2-5].

The knowledge extracting from an incomplete graph is link prediction (LP), enabling the retrieval of valuable information and insights from the graph [3]. The knowledge extraction is done in an embedded space. The feature vectors are converted into lower-dimensional representations known as embeddings. Several algorithms have been proposed for LP in the embedded knowledge graph [2-5].

The existing methods can be classified as the translation-based, bi-linear, and neural network-based categories [6].

The translation-based methods define a similarity relation in which the mapped related entities have the minimum distance in the embedded space. In the second category, called bi-linear, a score function is defined to assign a score to each triplet.The score indicates the proportionality of each triplet in the embedded space. This means that if the triplet is correct, its score will be high. The third category, neural network-based, primarily encompasses deep learning methods [2].

Most of the knowledge graph embedding methods utilize similarity or distance functions to compute the closeness between the Head, Relation, and Tail entities. However, relying solely on one type of closeness function may not be efficient enough to accurately determine the closeness of the embedded data in the mapped space. Furthermore, numerical methods would not be efficient in determining relations for textual data that can be semantically close but numerically distant [7].

To solve this problem, it is important to achieve the semantic relation between the entities.

Probabilistic models are very efficient in calculating the semantic relation of data. Probabilistic latent semantic analysis (PLSA), as a probabilistic model, provides high accuracy for semantic relation extraction [1, 8]. In this paper, we provide a new method for LP by adopting PLSA. The proposed method combines the learning of the geometrical structure of a graph with node similarity to obtain multiple perspectives on the data. Finally, we integrate the obtained views using PLSA [9, 10].

Our contributions are three folds:

- The correct triplet has the highest score and represents an acceptable concept in the embedded space.
- If a sufficient number of appropriate score functions are defined, a simple embedding will achieve a proper representation.
- The geometrical structure of a graph, combined with similarity measures, improves data representation in the embedded space by considering proximity between related data and the concept of correctness.

The structure of this paper is organized as follows; Section 2 presents an overview of the previous works on knowledge graph embedding methods; section 3 provides the proposed model. Experiment setup and experimental result are in section 4 and at the end section 5 is the conclusion of the paper

2. Previous Work

The LP models learn entities and their relations, which should be designed to preserve the local structure for the discovery of new facts.

The RESCAL method utilizes two similarity matrices: one between the head and relation, and the other between the tail and relation [6].

The proportional RESCAL of each entity involves considering a vector to preserve its latent meanings [8].

The matrix considered by RESCAL indicates relations between pairs of entities. The simplified DistMult can be seen as a variant of the RESCAL method, limited to a diagonal matrix. For each relation r, it is necessary to use an embedding vector and a matrix [9].

Inspired by the Word2Vec model, the TransE model was proposed in 2013 [10]. This model regards the relation r as the translation between the vector representations of two entities h and t. In other words, the triplet (h, r, t) is the embedding of an entity t near the translation of an entity h through the relation r, and the score function is defined as the distance between h and t.

The TransH model was proposed in 2014 [11]. In this model, an entity involved in different relations is allowed to have different representations. In fact, TransH selects a page

for embedding connections by mapping every relation r as a vector r onto the embedding page.

The TransR model, proposed in 2015, maps entities and relations onto two separate spaces denoted as R^k and R^d respectively [12]. This method employs a mapping matrix that maps the entity space onto the relational space r. The matrix first maps entities onto the relational space.

In the name embedding process, a translation method like TransE is utilized for word embedding and relation embedding. All embeddings are mapped onto a single space to aggregate the views, or the views can be weighted. Alternatively, the allocation matrices of each row for each view can also be used. However, these methods have certain disadvantages, including the mapping onto the same space, the challenge of identifying important views, or the increased computational complexity due to the model. In this study, the implemented feature extraction is solely based on the features of entities. Nevertheless, the empirical results indicate that [13]. TimE decomposes every relationship into entities and a diagonal mapping matrix [14], whereas CoNE was designed through the geometrical structure of a graph based on adjacency to calculate the features of entities with respect to their local adjacencies [15]. Cascade learning is a supervised learning method for neural networks. It employs the sequence in the objective function optimization, a process which is observed and controlled in every step. Most of the methods based on neural networks transform the initial data space into a new data space. Conceptual features and the graph of features are extracted from knowledge embedding and graph embedding, respectively [16]. These two sets are then employed to obtain a relationship between two features in cascade learning through three steps, each of which is calculated through a specific likelihood function.

These methods were able to solve one to many and many to many states gradually and these approaches reduced the number of parameters but the accuracy decreased. Extracted from the public knowledge bases, textual information can be employed to confirm the relation between two entities in the embedding process [15].

Random walk algorithms are employed to calculate the probability of paths between nodes in the graph, helping capture the influence and relationships between different entities in the network [7, 17, 18]. The random path selection approach may overlook certain potential nodes, leading to the neglect of their influence. The problem at hand is to identify which tails are most influenced by each pair of head-relation.

Most of the knowledge graph embedding (KGE) methods face four important constraints. First, the previous methods were semantically weak. In fact, the vector representation is only a point in the high-dimensional geometrical space [19]. Second, the entities have many features, but not all of these features are utilized in previous methods. Third, it is assumed that the roots of words are available as training data in large numbers for similarity functions; however, the number of training data is always limited. The embedding methods with clustering are not suitable due to the challenge of drawing numerical comparisons. Moreover, these methods often utilize Latent Dirichlet Allocation as a supervised technique to determine the dependence of an entity on a cluster. However, clustering improvement does not directly relate to views during the clustering process. Another problem is the retention of both the global structure and the local structure when dealing with high-dimensional data, which often leads to the curse of dimensional.

Extracted from the public knowledge bases, textual information can be employed to confirm the relation between two entities in the embedding process [20].

Additionally, embedding methods with clustering are not suitable due to their reliance on non-numerical comparisons. Furthermore, the use of LDA in these methods to determine the dependence of an entity on a cluster introduces a supervised aspect. Another challenge is the retention of both the global and local structure, especially in high-dimensional data, which leads to the curse of dimensionality.

The multiple features of the graph are taken into account through a multi-view policy. The proposed method will be described in the upcoming section.

Knowledge graph embedding models aim to learn entities and relations in order to improve a global loss function. It is crucial to consider this in a way that preserves the local structure, enabling the discovery of new facts.

3. Model Description

The proposed architecture is based on multiple views, including the position of a node, its adjacency with other nodes, the affected node, the information representation, and the independent presence of two entities and a relation.The features can be divided into multiple subsets called views completing each other [8].

The empirical results indicate that the latent features of a graph should also be taken into consideration, as they

complement each other [8]. Each view in the proposed method can have either a global or local feature graph. The global graph structure is used to capture semantic communication. However, unlike previous methods, the proposed method does not rely solely on node similarity. Instead, it takes into account the quality of communication for each node. This allows for the learning of structural communication, where the communications are not purely numerical.

We are searching for score functions (f) that can be learned through the PLSA process, as depicted in Figure 1. As illustrated in Figure 1, the input to the score function, denoted as x, is computed using the state pattern matrix. The score function f(x) is then calculated using Multi-view PLSA.

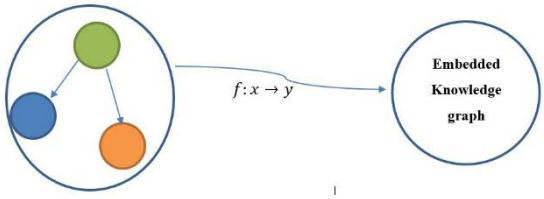

Figure 1

The mapping function

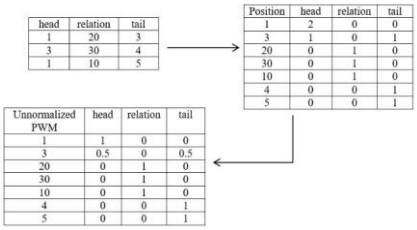

Figure 2

An example of normalized PWM matrix computation

3.1. State Pattern Matrix

As depicted in Figure 5, X represents the input, while Y represents a probability function. In Figure 1, the state pattern matrix is utilized as the input X. Additionally, the normalized PWM matrix is generated, as illustrated in **Error! Reference source not found.**.

The PWM (Prior Position Probability Matrix) is a matrix that represents the probability of a state for each triplet, as mentioned before [21]. It is commonly used in the detection of duplicated patterns in genes [22, 23]. To calculate the PWM matrix, the occurrence probability of each entity in a pattern needs to be determined. In the pattern matrix, the first column represents all the present entities and relations, while each subsequent column indicates a specific pattern. The rows of this matrix correspond to the number of entity representations in each pattern. The values of the unnormalized PWM (Position Weight Matrix) matrix are computed using the equation 1.

$$PWM_{i,j} = \frac{Position[i,j]}{\sum Position[i,:]} \qquad (1)$$

Which, i, and j refer to the rows and columns of the normalized PWM table, respectively. In the proposed architecture shown in Figure 5, the normalized PWM table is referred to as the pattern position and is used as X in Figure 1. The pattern position represents the probability of an entity being a head, a tail, a relation, or all three (head-relation-tail) together.

3.2. PLSA theory

The probabilistic latent semantic analysis (PLSA) is a generative learning model commonly used for classification tasks [2, 24] and is used for analyzing co-occurrence data. The proposed method utilizes PLSA to analyze the co-occurrence of (head-relation-tail) from a multi-view aspect. The effectiveness of the proposed method is attributed to the generative and multi-view learning capabilities of PLSA.

Assume that there are N input samples $D = \{d_0, d_1, \dots, d_N\}$, in which each d_i includes a triplet $d_i = (w_{hi}, w_{ri}, w_{ti})$ existing in $W = \{w_{h0}, \dots, w_{r0}, \dots, w_{t0}, \dots, w_{tN}\}$ where, It calculated by PWM matrix. The state equals one of the three members in $Z = \{head, relation, tail\}$.

The state model is calculated in the following way:

$$P(D|W) = \prod_{d,w} P(d,w) = \prod_{d,w} P(d)P(w|d)$$
$$where, p(w|d) = \sum P(w|z)P(z|d)$$
$$So, P(D,w) = \sum P(z)P(d|z)P(w|z) \qquad (2)$$

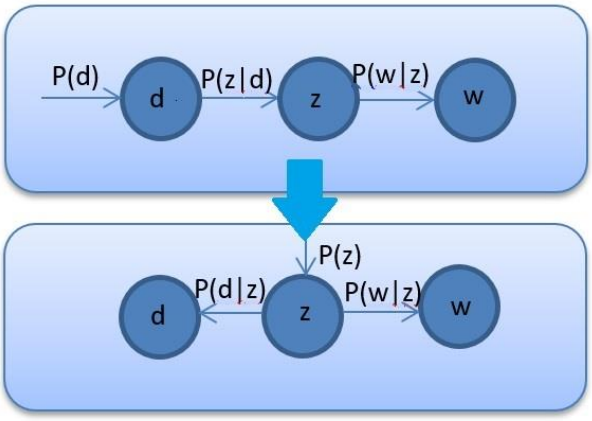

Figure 3

PLSA schema

PLSA decomposes into three probabilities as views. Equation 2 calculates the occurrence probability of each state independently, denoted as P(z). The probability of each triplet occurring together is represented by $P(d \mid z)$. Additionally, $P(d \mid z)$ indicates the occurrence probability.

Each PWM entry corresponds to a specific state. Therefore, PLSA can integrate multiple views if each view is represented as a probability. In equation 2, the conditional probability can be transformed into a joint probability. The specific method for this transformation is described in section 3.3 of the paper.

3.3. Proposed Multi-view PLSA

Assuming the information provided in Figure 3, the probability of each PWM entry is stored in $P(W \mid Z)$, where Z represents the (head, relation or a tail) and W represents a PWM entry [22].

The $P(W \mid Z)$ will be performed according to equation 3:

$$P(W|Z) = P(Words \mid Head, Relation, Tail) = \cdots$$
$$\ldots P(Words) \prod P(head, relation, tail) \quad (3)$$

Since each W can belong to the tail, relation, or head type, the conditional probability in equation 3 is converted into equation 4. Then:

$$P(words) \prod P(head, relation, tail) \Rightarrow^{\ln}$$
$$\ln(P(words \mid head, relation, tail)) =$$
$$\ln P(words) + \ln(\sum P(head, relation, tail)) \quad (4)$$

There is a latent set of y in PLSA for two sets of d and f:

$$P(f, d) = \sum_{y} P(f, d, y) = \sum_{y} P(f \mid y)P(y \mid d)P(d) \quad (5)$$

According to equation 5, if the relation is decomposed, then:

$$\ln(P(words \mid head, relation, tail)) = \cdots$$
$$\ldots \ln(p(words)) + \ln(\sum_{i} P(head, relation, tail)) \quad (6)$$

Thus, equation 3 will lead to the following output:

$$\ln(P(words \mid head, relation, tail)) =$$
$$\ln(p(words)) + \ln(\sum_{i} P(head, tail))$$
$$P(W \mid Z) = P(words \mid head, relation, tail)$$
$$= P(word) * P(head, tail) \quad (7)$$

So, the conditional probability is transformed into the joint probability of head and tail. Here, $P(word)$ represents the probability of each word occurring as tail, relation, or head. Additionally, $P(head, tail)$ denotes the probability of the (head-tail) combination, which can be obtained using equation 8. Therefore, the multi-view PLSA can be expressed as follows:

$$p(W \mid Z) = P(words \mid head, relation, tail) = \cdots$$
$$\ldots \alpha * P(head, tail) = \alpha * \prod_{i \in views} P_i(head, tail) \quad (8)$$

According to equation 6, the conditional probability in Figure 3 is transformed into the calculation of joint probability as shown in equation 8. In this context, $p(W|Z)$ represents a probability function for each triplet, while $P(head, tail)$ denotes the probability of each head and tail occurring together and P(head or relation or tail) denotes the probability of each head-relation or relation together.

The final matrix is calculated as depicted in Figure 4.

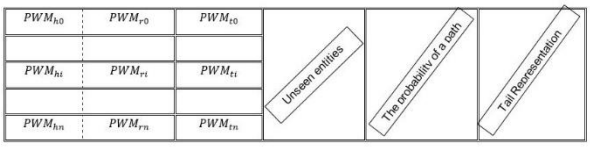

Figure 4

The final matrix

3.4. *The Proposed views*

The attributed network embedding framework is utilized as views, incorporating attributes that capture the geometrical learning of the graph and are commonly used in social network analysis. In accordance with Equation 8 and Figure 1, the input of the score function is the normalized PWM matrix, and the output is the probability $P(head, tail)$ in each view.

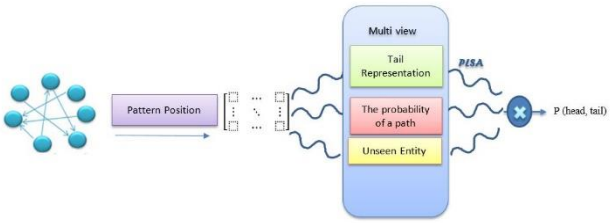

Figure 5

Proposed architecture

3.4.1. *The First and second view: the ability and the effectiveness of head relation pairs in tail representation Each triplet of head-relation-tail plays a role in determining the final score probability and data representation.*

This process is similar to detecting influential nodes in information diffusion on social networks, with the goal of determining the nodes that have the greatest impact on the spread of information or influence within the network [25, 26]. Indeed, the nodes that have a significant impact on social networks are those that can attract more viewers when they share a post. The probability of a path is determined based on the importance of a head-tail relation in representing information. In simpler terms, if there are fewer steps between a head-relation pair and a tail, the head-relation pair is considered more effective in representing the tail. Consequently, the priority of representing the tail by the head-relation pair increases. On the contrary, the probability of tail representation decreases as the number of steps increases. The selection of an effective triplet in representing information is not solely based on the similarity to other adjacent nodes [27, 28].

3.4.2. *The probability of a path*

The probability of a path is typically determined using random walk algorithms, which are commonly used for graph geometry learning [29, 30]. However, in this paper, a different method was employed due to certain limitations of the random walk approach. When searching for a path from one node to another, the directness of the graph should be taken into consideration. Simply having a path between two nodes does not guarantee the directness of the graph unless the resulting path matrix is asymmetric and only one target node is considered.

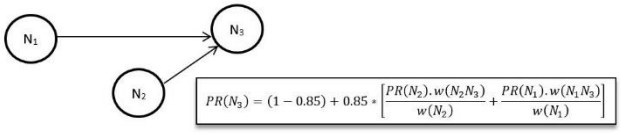

Figure 6

Formal Page Rank calculation example, W means weight and can be related to question

The problem definition

If the input graph is denoted as G= (V, E), where V represents the vertices and E represents the edges, the edges can indicate the degree of importance when the graph is based on the effectiveness of each parent node in representing child nodes. In other words, each head-relation pair will have a specific degree of importance in representing a tail. Since the representation of a tail depends on a head-relation pair, the directness of the graph will be taken into consideration.

The Page Rank algorithm

PageRank is an iterative algorithm commonly used to analyze and rank web pages [27, 28]. This algorithm can also be utilized to recommend tail nodes based on the effectiveness rate of each head-relation-tail in representing a concept. In this context, each edge between two nodes represents the rank and importance of a tail node.

Unlike the random walk algorithm, where selections are completely random, the proposed algorithm selects paths based on the degree of each node and similarity of a random node. The increased degree of a node indicates its effectiveness in describing information. The input of this algorithm is a graph, and its output is the importance of each node. Hence, increasing this value helps prevent the loss of additional information during the embedding process. The proposed rank of each tail is obtained from equation 9.

$$PR(tail) = \left((1-d) + d \sum_{v \in B(tail)} \frac{PR(head).sim(head-relation)}{\sum_v sim(head-relation)}\right) * \mathrm{P(head-tail)} \quad (9)$$

In equation 9, $P(head - tail)$ and $P(tail)$ represent the probability of a head-tail connection and the PageRank of a tail, respectively. Specifically, $P(head - tail)$ denotes the probability of a head-tail connection with each relation, while $B(tail)$ represents the set of nodes that have edges to the specific tail. Additionally, $w(head - relation)$ represents the edge weight, which is equal to the similarity between the head and relation. The edge weight, calculated using cosine similarity, is determined with a default damping factor of 0.85. According to equation 9, the importance of a head-relation pair is calculated in the representation of a tail. An example of this calculation is illustrated in Figure 6.

In our proposed method, we utilize a weighted transmission matrix as the initial value of $PR(head)$ [7, 31].

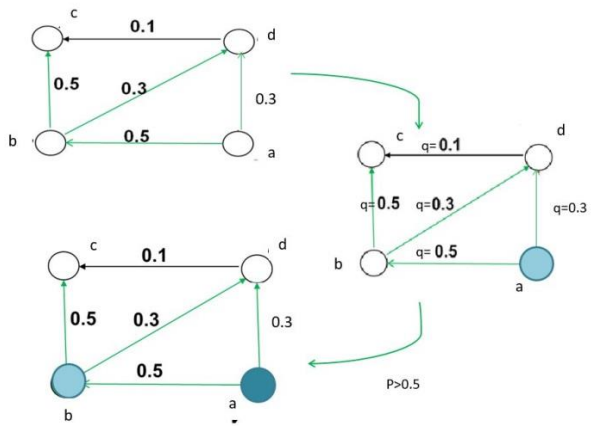

Figure 7

An example for Information Diffusion calculations

3.4.3. *Effective Pair in the Tail Representation*

Information diffusion refers to the process of spreading information across a network through its backbone or underlying structure [32]. It involves the transmission and dissemination of information from one node to another, leading to its propagation throughout the network. In information diffusion, certain nodes that exhibit slight similarities can play a significant role due to the higher density of edges among them. These nodes form the backbone of the graph, and their strong interconnections make them influential representatives of information. These nodes are commonly referred to as opinion leaders [24]. They have the ability to shape and influence the spread of information within a network. Using similarity-based embedding alone may result in the loss of some information, as effective nodes can be merged into other nodes. However, by incorporating additional parameters such as node degrees in the calculations, the loss of information can be mitigated, resulting in a more efficient representation. In this context, the tail node can be considered an opinion leader. One method of information diffusion is the cascade model of diffusion [32], which describes the spread of information through a network in a sequential manner, similar to a cascading effect.

In this model, a graph is constructed to represent social networks based on the presence of relations between individuals [33].

The nodes in the diffusion graph are categorized into active and inactive types, and the weights assigned to the nodes indicate the degree of success in activating the target node. These weights reflect the influence or impact that each node has on the activation process.

Figure 7 illustrates a case of information diffusion [7]. After the graph is constructed, steps are taken from one node to the next based on a probability value, and the success of each attempt is determined using different criteria. In this paper, the success of an attempt is measured by the head-relation similarity of a proposed tail, which is calculated using the equation 1.

$$P_{ij} = \left(\frac{d_i}{n} + \frac{CN_{ij}}{n}\right) * \sum \frac{1}{H(Head, nodes)} \qquad (10)$$

In equation 10, d_i represents the degree of node i, and CN_{ij} denotes the number of common neighbors between nodes i and j. Additionally, $H(Head, nodes)$ indicates the inverted summation of distances between two head nodes and the other nodes.

3.4.4. *Third view: each of head and tail can change its role*

The probability of being displayed for each entity in the triplet is calculated. This view implies that every unseen entity in the random position of a head or a tail can be considered independently.

3.4.5. *Unseen entities*

The position or state of a node is determined to calculate the occurrence probability of every head-relation-tail pattern. This probability reflects the effectiveness of a pattern in representing the data.

If the position probability of each entity is dependent on each position, then:

$$P(head, tail) = P(head) * P(relation) * P(tail) \quad (11)$$

The probability score function aims to determine the occurrence probability of possible entities in response to a head-relation or relation-tail pair. This probability can be considered equal to the occurrence probability of a tail in the head state and vice versa with each relation.

4. Experiment

In this section, the quality of the proposed algorithm is analyzed in the context of the link prediction problem using three public datasets. The performance and effectiveness of the algorithm are evaluated and compared against the existing methods using these datasets.

4.1. Experiment setup

Environment: All of our experiments have been performed on a server environment using 16 CPUs Intel Core(TM) i7-3820 at 3.60GH, 64GB RAM.

Indeed, link prediction refers to the task of predicting the missing tail node in a triplet (head, relation, tail), where one of the nodes is unknown or missing. The goal is to accurately propose the correct tail node that should be associated with the given head and relation, in order to represent the intended concept or relationship accurately.

Train Setup: According to **Error! Reference source not found.**, the normalized PWM is calculated using the training data to determine the prior value for each data point. Subsequently, each training data point is transformed into its corresponding prior value, which is then used to create the embedding matrix or map it onto the numerical space. In the next step, the three designed views, each estimating a value for equation (insert equation number here), are utilized through the support vector regression kernel learning algorithm based on PLSA. This allows for the estimation and learning of the probability $P(head, tail)$.

Test Setup: In papers on artificial intelligence, the evaluation parameters are typically calculated as the mean of various tests. However, in this paper, a different testing process was designed. For each triplet in the test data, a set of M correct data and M-1 random data points were created. The evaluation parameters, including equation 8, were then measured for each member of the set. The correct answer is expected to have the maximum probability among the options. This approach allows for a more comprehensive evaluation of the proposed method.

The final value of the evaluation parameter is calculated as the mean value. During the testing phase, the data is first transformed into prior values using the pattern matrix. Then, the tails are randomly proposed. The learned views are utilized to validate the proposed tails, and the tail with the highest probability is selected as the final solution. The value of M, which represents the number of correct data points in the set, is set to 10, 20, and 50.

Evaluation Parameter: The evaluation parameters MR and @hitK are commonly used to assess the performance of a model in link prediction tasks.

MR (Mean Rank) measures the average rank of the correct solution among the proposed data. It provides an indication of how well the model is able to rank the correct answer compared to other options. A lower MR value indicates better performance.

@hitK measures the number of times the correct solution is included within the top K proposed options. It assesses the model's ability to identify the correct answer within a given range. A higher @hitK value indicates better performance.

These parameters are typically calculated based on a test dataset with a total of |T| test instances.

$$hit@k = \frac{1}{T}\sum_{t<T} I[rank_t < K]$$

$$MR = \frac{1}{T}\sum rank \quad (12)$$

The test was conducted on three databases: FB15k237, FB15k, and WN18RR. Table 1 provides information on the number of entities and relations in each database. The hyperparameter values were initialized using the validation set, and the designated kernel for the support vector regression algorithm is the Radial Basis Function (RBF) kernel.

Table 1

The LP datasets included in our comparative analysis

Data set	Entities	Relations	Train	Valid	Test
WN18RR	40943	11	86000	3034	3134
FB15k-237	14541	237	272115	17535	20466
FB15k	14951	1345	400000	50000	30000

The test is conducted using both raw and filtered methods. In the raw test, the rank of a correct solution is calculated without excluding any correct probability solutions. In the filtered method, the ranks are calculated while considering the filters of the other correct solutions. This paper specifically employs the filtered method because

the viewing probability of correct solutions is minimal in the created test set.

Table 2 and Table 3 illustrate the differences between the truth tail score function and the best calculated value for M values of 10, 20, and 50. The x-axis represents the designated test sample number, while the y-axis represents the difference in values between the two calculated values based on equation 12.

Based on the conducted test, it was observed that the calculated probability of the correct solution often had the highest value. The resulting values of @Hit and MRR were calculated as the mean of the tests. Table 2 and Table 3 present the resulting values of @Hit and MR for different values of M and k in the FB15k-237 and FB15k datasets.

Table 2

The LP datasets included in our comparative analysis

Test	Samples in each test	@Hit1	MR
20466	10	49.88	5.35903
	20	48.64	10.3218
	50	48.03	21.4871

Table 3

The LP datasets included in our comparative analysis

Test	Samples in each test	@Hit1	MR
30000	10	21.22	4.277
	30	10.94	12.659
	40	11.65	16.7

Figure 8 and Figure 9 illustrate the changes in the values of @Hit and MR. The x-axis represents the increase in the number of samples or the value of M, while the y-axis represents the values of @Hit and MR. It is evident that as the number of test data increases, both parameters will eventually reach a constant value. The final values of @Hit and MR can be determined by identifying the value at which the changes become constant in Figure 8 and Figure 9.

As the number of samples increases, the values of @Hit and MR reach a constant level. This indicates that the proposed method ensures that the correct solution is included among the initial guesses. Additionally, @Hit1 represents the probability that the initial guess is correct, while MR represents the range of guesses within which the correct solution exists.

In the WN18RR dataset, there is limited structural information available about the knowledge graph. To compensate for this, a word2vec model with a size of 2 is employed for each relation. These two word2vec vectors are utilized during the training step to enhance the representation and understanding of the relations in the dataset.

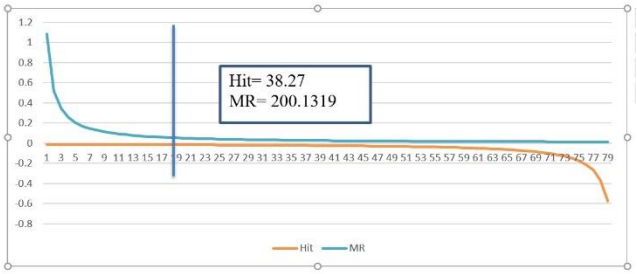

Figure 8

@Hit and MR values reach a constant value after increasing the number of test samples. The top view is on the FB15k-237 data base.

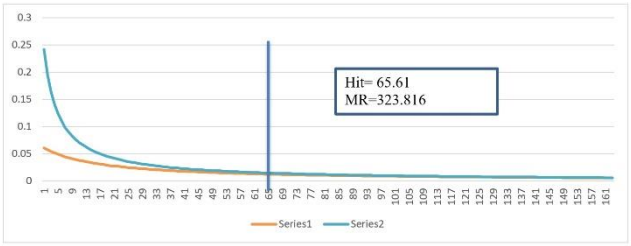

Figure 9

@Hit and MR values reach a constant value after increasing the number of test samples. The top view is on the FB15k data base.

Error! Reference source not found. presents the values of @Hit and MR for different numbers of tests. Figure 10 illustrates the changes in these two parameters as the number of test data increases. It can be observed that both @Hit and MR reach nearly constant values as the number of test data increases.

Figure 10

@Hit and MR values reach a constant value after increasing the number of test samples. The top view is on the WN18RR base.

Table 4

Result of each metric in WN18RR

Test	Samples in each test	@Hit1	MR
3134	10	18.54	0
	50	94.7	1.735
	100	92.69	2.075
	200	19.449	2.2125

Error! Reference source not found. provides a comparison between the proposed algorithm and other methods. The focus of the proposed method is on enhancing performance in datasets such as FB15k and FB15k-237, where there are more relations than entities. It is observed that most of the existing methods did not perform well on these datasets. The proposed method stands out as a more efficient learner due to its integration of the geometric structure of the graph with the similarity of nodes. Additionally, the complex structure of the knowledge graph is taken into account by incorporating word2vec vectors during the training step, which helps capture more information and improve the learning process.

According to **Error! Reference source not found.**, the proposed method demonstrated superior performance compared to other techniques. As shown in Figure 8 and Figure 10, the MR value reached a constant level. This indicates that the correct guess is within a specific range, which has the minimum value based on the evaluation results presented in **Error! Reference source not found.**.

Although the value of Hit1 may not be optimal for WN18RR, the value of MR is still suitable. This suggests that while the first guess may not always be correct, it is likely to be among the top 100 solutions. As shown in Figure 9 and Figure 10, the MR value reaches a constant level, indicating that the correct guess is within a specific range that has the minimum value according to the results presented in **Error! Reference source not found.**.

Table 5 provides a comparison between the proposed algorithm and state-of-the-art algorithms, including deep learning approaches. As shown, the proposed algorithm achieves optimal results compared to the other methods. The Hit parameter indicates the probability that the correct answer is among the top ten answers, while the computed Hit parameter specifically refers to the top five answers. The proposed algorithm demonstrates superior performance in terms of both accuracy and efficiency.

As evident from Table 5 and Table 1, the proposed algorithm has successfully performed structure learning even in cases where there is an imbalance between relations and entities. It is crucial to utilize appropriate structural features based on the dataset's characteristics to achieve optimal results. By incorporating relevant structural information, the proposed algorithm can effectively capture the underlying patterns and relationships within the data, leading to improved performance.

Table 5

Result of each metric in WN18RR

algorithm	FB15k-237		WN18RR		FB15k	
	@Hit1	MR	@Hit1	MR	@Hit1	MR
DistMult	22.44	199	39.68	5913	73.61	173
ComplEx	25.72	202	42.55	4907	81.56	34
ANALOGY	12.59	476	35.82	9266	65.59	126
SimplE	10.03	651	38.27	8764	66.13	138
HolE	21.37	186	40.28	8401	75.82	211
TransE	21.72	209	2.79	3936	49.36	45
STransE	22.48	357	10.13	5172	39.77	69
CrossE	21.21	227	38.07	5212	60.08	136
TorusE	19.62	211	42.68	4873	68.85	143
RotatE	23.83	178	42.60	3318	73.93	42
ConvE	21.90	281	38.99	4944	59.46	51
ConvKB	13.98	309	5.63	3429	11.44	324
ConvR	25.56	251	43.73	5646	70.57	70
Proposed Algorithm	65.61	323.816	81.56783	3.1455	38.27	200.1319

5. Conclusion

This paper presents a method for knowledge graph embedding in link prediction tasks, which integrates the geometric structure of the graph with the similarity of nodes. Additionally, the proposed method can be seen as a Markov model, as it measures the probability of a relation based on a few prior nodes.

The proposed method is designed to consider factors such as similarity, the importance of each node, and the overall

graph structure in each view. By incorporating these elements, the method aims to capture the underlying patterns and relationships within the knowledge graph, leading to improved link prediction performance.

The Freebase database has a significant difference in the number of relations and entities, as indicated in Table 5. The proposed method achieved notable improvements in the Freebase database. This suggests that emphasizing the geometric structure of the graph can enhance the performance of link prediction.

Authors' Contributions

All authors equally contributed to this study.

Declaration

In order to correct and improve the academic writing of our paper, we have used the language model ChatGPT.

Transparency Statement

Data are available for research purposes upon reasonable request to the corresponding author.

Acknowledgments

We would like to express our gratitude to all individuals helped us to do the project.

Declaration of Interest

The authors report no conflict of interest.

Funding

According to the authors, this article has no financial support.

Ethical Considerations

Not applicable.

References

[1] B. Nie and S. Sun, "Knowledge graph embedding via reasoning over entities, relations, and text," *Future Generation Computer Systems*, vol. 91, pp. 426-433, 2019, doi: 10.1016/j.future.2018.09.040.

[2] B. Jagvaral, W. K. Lee, J. S. Roh, M. S. Kim, and Y. T. Park, "Path-based reasoning approach for knowledge graph completion using CNN-BiLSTM with attention mechanism," *Expert Systems with Applications*, vol. 142, p. 112960, 2020, doi: 10.1016/j.eswa.2019.112960.

[3] Q. Zhang, R. Wang, J. Yang, and L. Xue, "Knowledge graph embedding by translating in time domain space for link prediction," *Knowledge-Based Systems*, vol. 212, p. 106564, 2021, doi: 10.1016/j.knosys.2020.106564.

[4] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Expert Systems with Applications*, vol. 141, p. 112948, 2020, doi: 10.1016/j.eswa.2019.112948.

[5] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494-514, 2021, doi: 10.1109/TNNLS.2021.3070843.

[6] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," *Advances in Neural Information Processing Systems*, vol. 26, 2013. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/b337e84de8752b27eda3a12363109e80-Paper.pdf.

[7] F. Xia, J. Liu, H. Nie, Y. Fu, L. Wan, and X. Kong, "Random walks: A review of algorithms and applications," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 2, pp. 95-107, 2019, doi: 10.1109/TETCI.2019.2952908.

[8] H. Xiao, Y. Chen, and X. Shi, "Knowledge graph embedding based on multi-view clustering framework," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 585-596, 2019, doi: 10.1109/TKDE.2019.2931548.

[9] X. Bai, L. Zhu, C. Liang, J. Li, X. Nie, and X. Chang, "Multi-view feature selection via nonnegative structured graph learning," *Neurocomputing*, vol. 387, pp. 110-122, 2020, doi: 10.1016/j.neucom.2020.01.044.

[10] B. Zhang, Q. Qiang, F. Wang, and F. Nie, "Fast multi-view semi-supervised learning with learned graph," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 286-299, 2020, doi: 10.1109/TKDE.2020.2978844.

[11] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," presented at the Proceedings of the AAAI Conference on Artificial Intelligence, 2016.

[12] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in Neural Information Processing Systems*, vol. 26, 2013. [Online]. Available: https://papers.nips.cc/paper_files/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

[13] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," presented at the Proceedings of the AAAI Conference on Artificial Intelligence, 2015.

[14] P. Kumar and M. Vardhan, "Aspect-Based Sentiment Analysis of Tweets Using Independent Component Analysis (ICA) and Probabilistic Latent Semantic Analysis (pLSA)," in *Advances in Data and Information Sciences: Proceedings of ICDIS 2017, Volume 2*: Springer Singapore, 2019, pp. 3-13.

[15] N. Guan, D. Song, and L. Liao, "Knowledge graph embedding with concepts," *Knowledge-Based Systems*, vol. 164, pp. 38-44, 2019, doi: 10.1016/j.knosys.2018.10.008.

[16] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," presented at the Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015. [Online]. Available: https://aclanthology.org/P15-1067/.

[17] A. Jabri, A. Owens, and A. Efros, "Space-time correspondence as a contrastive random walk," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19545-19560, 2020. [Online]. Available: https://arxiv.org/abs/2006.14613.

[18] Y. He, Y. Song, J. Li, C. Ji, J. Peng, and H. Peng, "Hetespaceywalk: A heterogeneous spacey random walk for heterogeneous information network embedding," presented at the Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019.

[19] S. Rani and M. Kumar, "Topic modeling and its applications in materials science and engineering," *Materials Today: Proceedings*, vol. 45, pp. 5591-5596, 2021, doi: 10.1016/j.matpr.2021.02.313.

[20] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724-2743, 2017, doi: 10.1109/TKDE.2017.2754499.

[21] J. Fostier, "BLAMM: BLAS-based algorithm for finding position weight matrix occurrences in DNA sequences on CPUs and GPUs," *BMC Bioinformatics*, vol. 21, pp. 1-13, 2020, doi: 10.1186/s12859-020-3348-6.

[22] F. Albalawi, S. Alshehri, A. Chahid, and T. M. Laleg-Kirati, "Voxel Weight Matrix-Based Feature Extraction for Biomedical Applications," *IEEE Access*, vol. 8, pp. 121451-121459, 2020, doi: 10.1109/ACCESS.2020.3006521.

[23] S. Tsioutsiouliklis, E. Pitoura, P. Tsaparas, I. Kleftakis, and N. Mamoulis, "Fairness-aware pagerank," presented at the Proceedings of the Web Conference 2021, 2021.

[24] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," presented at the Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part IV 9, 2006.

[25] Y. Lin, H. Zhang, R. P. Zhang, and Z. J. M. Shen, "Nonprogressive diffusion on social networks: Approximation and applications," *SSRN*, 2022. [Online]. Available: https://dx.doi.org/10.2139/ssrn.4232670.

[26] L. A. Dennis, Y. Fu, and M. Slavkovik, "Markov chain model representation of information diffusion in social networks," *Journal of Logic and Computation*, vol. 32, no. 6, pp. 1195-1211, 2022, doi: 10.1093/logcom/exac018.

[27] W. C. Yeh, W. Zhu, C. L. Huang, T. Y. Hsu, Z. Liu, and S. Y. Tan, "A new BAT and PageRank algorithm for propagation probability in social networks," *Applied Sciences*, vol. 12, no. 14, p. 6858, 2022, doi: 10.3390/app12146858.

[28] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-Pour, "MGFS: A multi-label graph-based feature selection algorithm via PageRank centrality," *Expert Systems with Applications*, vol. 142, p. 113024, 2020, doi: 10.1016/j.eswa.2019.113024.

[29] E. Nasiri, K. Berahmand, M. Rostami, and M. Dabiri, "A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding," *Computers in Biology and Medicine*, vol. 137, p. 104772, 2021, doi: 10.1016/j.compbiomed.2021.104772.

[30] A. B. Asl, J. Raghothama, A. Darwich, and S. Meijer, "Using pagerank and social network analysis to specify mental health factors," presented at the Proceedings of the Design Society, 2021.

[31] C. Coquidé, J. Queiros, and F. Queyroi, "Pagerank computation for higher-order networks," presented at the Complex Networks & Their Applications X: Volume 1, Proceedings of the Tenth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2021, 2022.

[32] A. Foroozani and M. Ebrahimi, "Anomalous information diffusion in social networks: Twitter and Digg," *Expert Systems with Applications*, vol. 134, pp. 249-266, 2019, doi: 10.1016/j.eswa.2019.05.047.

[33] S. N. Firdaus, C. Ding, and A. Sadeghian, "Retweet: A popular information diffusion mechanism–A survey paper," *Online Social Networks and Media*, vol. 6, pp. 26-40, 2018, doi: 10.1016/j.osnem.2018.04.001.