UDC 004.048
IRSTI 28.23.25
https://doi.org/10.55452/1998-6688-2023-20-2-92-102

ALITURLIYEVA A.E
Kazakh-British Technical University, 050000, Almaty, Kazakhstan
E-mail: a_aliturliyeva@kbtu.kz

LINK PREDICTION USING TENSOR DECOMPOSITION

Abstract. In recent years, tensor decomposition has gained increasing interest in the field of link prediction, which aims to estimate the likelihood of new connections forming between nodes in a network. This study highlights the potential of the Canonical Polyadic tensor decomposition in enhancing link prediction in complex networks. It suggests effective tensor decomposition algorithms that not only take into account the structural characteristics of the network but also its temporal evolution. During the process of tensor decomposition, the initial tensor is decomposed into two-way tensors, also known as factor matrices, representing different modes of the data. These factor matrices capture the underlying patterns or relationships within the network, providing insights into the structure and dynamics of the network. For evaluation, we examine a dataset derived from the WSDM. After preprocessing, the data is represented as a multi-way tensor, with each mode representing different aspects such as users, items, and time. Our primary objective is to make precise predictions about the links between users and items within specific time periods. The experimental results demonstrate that our approach significantly improves prediction accuracy for evolving networks, as measured by the AUC.

Key words: link prediction, CP decomposition, Jennrich's algorithm, ALS algorithm, exponential smoothing, BiLSTM.

АЛИТУРЛИЕВА А.Е
Казақстан-Британ техникалық университет, 050000, Алматы қ., Қазақстан
E-mail: a_aliturliyeva@kbtu.kz

ТЕНЗОРЛЫҚ ЫДЫРАУ АРҚЫЛЫ БАЙЛАНЫСТЫ БОЛЖАУ

Аңдатпа. Қазіргі уақытта тензордың ыдырауы желідегі түйіндер арасында жаңа қосылыстардың пайда болу ықтималдығын бағалауға бағытталған байланыстарды болжау саласында қызығушылықты арттыруда. Бұл зерттеу күрделі желілердегі байланыстарды болжауды жақсарту үшін Канондық Полиадикалық тензор ыдырауының қолданысын көрсетеді. Сонымен қатар желінің құрылымдық сипаттамаларын ғана емес, оның уақытша эволюциясын да ескеретін тиімді тензорлық ыдырау алгоритмдері ұсынылған. Тензордың ыдырау процесі кезінде бастапқы тензор деректердің әртүрлі режимдерін білдіретін факторлық матрицалар деп те аталатын екі өлшемді тензорларға ыдырайды. Бұл фактор матрицалары желінің құрылымы мен динамикасы туралы түсініктерді қамтамасыз ете отырып, желі ішіндегі негізгі заңдылықтарды немесе қатынастарды көрсетеді. Модельді бағалау үшін біз WSDM-ден алынған мәліметтер жиынтығын қарастырдық. Алдын ала өңдеуден кейін деректер көп деңгейлі тензор ретінде ұсынылды, әр режим пайдаланушылар, элементтер және уақыт сияқты әртүрлі аспектілерді білдіреді. Біздің басты мақсатымыз-белгілі бір уақыт аралығында пайдаланушылар мен элементтер арасындағы байланыстарға қатысты нақты болжамдар жасау. Эксперименттік нәтижелер біздің тәсіліміздің AUC арқылы өлшенетін уақыт бойынша өзгеретін желілерді болжау дәлдігін айтарлықтай жақсартатынын көрсетеді.

Тірек сөздер: байланысты болжау, CP-декомпозиция, Генрих алгоритм, ALS алгоритм, экспоненциалды тегістеу, BiLSTM.

АЛИТУРЛИЕВА А.Е
Казахстанско-Британский технический университет, 050000, г. Алматы, Казахстан
E-mail: a_aliturliyeva@kbtu.kz

ПРОГНОЗ СВЯЗИ С ИСПОЛЬЗОВАНИЕМ ТЕНЗОРНОЙ ДЕКОМПОЗИЦИИ

Аннотация. В последние годы тензорная декомпозиция вызывает все больший интерес в области прогнозирования связей, целью которого является оценка вероятности образования новых соединений между узлами в сети. Это исследование подчеркивает потенциал Канонической Полиадической тензорной декомпозиции для улучшения предсказания связей в сложных сетях. В процессе тензорной декомпозиции исходный тензор разлагается на двумерные тензоры, также известные как матрицы факторов, представляющие различные режимы данных. Эти факторные матрицы фиксируют базовые закономерности или отношения внутри сети, обеспечивая понимание структуры и динамики сети. В нем предлагаются эффективные алгоритмы тензорной декомпозиции, которые учитывают не только структурные характеристики сети, но и ее временную эволюцию. Для оценки мы изучаем набор данных, полученный на WSDM. После предварительной обработки данные представляются в виде многоуровневого тензора, причем каждый режим представляет различные аспекты, такие как пользователи, элементы и время. Наша основная цель – сделать точные прогнозы относительно связей между пользователями и товарами в течение определенных периодов времени. Экспериментальные результаты демонстрируют, что наш подход значительно повышает точность прогнозирования для развивающихся сетей, измеряемую AUC.

Ключевые слова: предсказание связи, CP-декомпозиция, алгоритм Генриха, алгоритм ALS, экспоненциальное сглаживание, BiLSTM.

Introduction

“Tensor” was first introduced in 1927 [1], and the idea of using more than two matrices in factor analysis has been widely accepted since the 1960s in various domains [2]. Complex interactions among input features can be captured using a tensor form, which is impossible with flattened data. However, any analysis on a full tensorial representation is often accompanied by a so-called curse-of-dimensionality challenge, with the complexity increasing exponentially with the tensor order. This is where tensor decompositions play a crucial role, allowing for lessening the data representation's complexity without significantly affecting its ability to capture correlations in the data. Similar to their matrix counterpart, tensor decompositions break down high-dimensional tensors into a sum of lower-dimensional factors. In addition to their direct use in processing multi-way input data, tensors are frequently used as a core component of machine learning models. In later years, tensor decomposition has gained increasing interest in various fields, including computer vision and social network analysis [3].

Some existing studies compared tensor decomposition-based link prediction methods with other popular graph-based link prediction methods in multi-relational data. For analysis of temporal multi-relational data, Bader et al. [7] employ a decomposition method called ASALSAN related to RESCAL. As noted by Nickel et al. [8], this decomposition method has shown suboptimal performance on previous benchmarks. Ma et al. [9] proposed another generalization of RESCAL called the ConT decomposition method for temporal link prediction. The core tensor indices are contracted in this method, lowering the computational complexity. Evrim et al. [10] explore various matrix and tensor decomposition methods for solving link prediction problems. They consider author-conference relationships in bibliometric data called DBLP and propose an extension of the matrix-based Katz method, which employs truncated SVD for approximation. However, the authors conclude that the tensor-based decomposition methods are much better than matrix-based decomposition methods. Because temporal latent trends are not entirely derived via matrix-based decomposition from evolving data.

Lin et al. [11] offer a decomposition approach for community extraction on multi-relational and multi-dimensional social data. Their coupled factorization method includes CANDECOMP and PARAFAC decomposition methods and divergence-based cost function. Furthermore, Narita et al. propose a joint factorization method based on Tucker and CP decomposition methods and utilize a Euclidian distance-based loss function. Finally, Liang et al. [12] implement the Bayesian Probabilistic Tensor Factorization (BPTF) algorithm for temporal relational data. BPTF can capture the overall evolution of latent features by imposing a smoothness constraint on those features and incorporating additional time features. Sheng et al. [13] proposed a new method called Link Pattern Prediction Tensor (LPPT) based on Tucker Decomposition, which captures interaction patterns in multi-relational networks. Chi and Kolda [14] presented the CP Alternating Poisson Regression (CP-APR), suitable for handling weighted time-evolving networks because it is made for sparse count data. The primary concept of the TensorCast method proposed by Araujo et al. [15] is using non-negative coupled tensor decomposition and standard models to forecast the time component.

Main provisions

Several popular tensor decomposition methods, including Canonical Polyadic (CP) and Tucker decomposition, decompose tensor-formed multidimensional data into low-order tensors and identify underlying trends of decomposed tensors. Tucker decomposition aims to decompose a tensor into a core tensor with the same order and low-order factor tensors [4]. In contrast, the CP decomposition represents an observed tensor as a sum of rank-one tensors. The CP decomposition methods first found applications in psychometrics [5] and linguistics [6], where they were referred to as Canonical Decomposition and Parallel Factor models, respectively. In this study, we use CP decomposition algorithms to recover the factor matrices model to make predictions about potential edge connections within a given network. Furthermore, our approach captures temporal trends within a tensor, where time is treated as a separate dimension.

Tensor decomposition has drawn more interest recently in link prediction, which estimates the possibility of new connections forming between network nodes. Numerous research has concentrated on the static features of single graph snapshots, which do not reveal the behavior of networks. Tracking patterns over time that are impacted by adding and deleting nodes to forecast links is essential. The topic of collaborative filtering, which aims to recommend new things to a user, is closely connected to link prediction. In this problem, the input is a partially observed matrix of (user, item) preference scores. In collaborative filtering, users and items are represented by nodes, and edges pairing nodes are weighted by the preference score. The dataset we examine is derived from Amazon, which was published in WSDM 2022 Cup. After preprocessing, it is represented as a multi-way tensor where each mode corresponds to different aspects of the data, such as users, items, and time. Our main objective is to make accurate predictions about the links between users and items in specific time periods. The results show that our approach yields significant improvements in prediction accuracy for evolutionary networks, as measured by AUC. Overall, this research demonstrates the potential of the CP decomposition method in improving link prediction in complex networks. This paper suggests effective tensor decomposition methods for temporal link prediction for large-scale, complicated networks considering temporal data.

Materials and Methods

A tensor is a generalization of multi-way arrays. The number of dimensions determines a tensor order. The order of a tensor is an important property, as it determines how it behaves under several types of transformations. For convenience, we use a three-dimensional tensor as an example throughout this paper, but the notation can extend to tensors of higher dimensions in most cases. The notation is primarily based on the reviews by Kolda et a,l. [3] In this paper higher-order tensors are denoted by underlined uppercase letters, e.g., $\underline{X} \in R^{(i_1 \times .. \times i_n)}, \mathrm{n} \geq 3$. For convenience, we use lower case letters to denote vectors $\mathrm{x} \in R^{i}$ and upper case to matrices $\mathrm{X} \in R^{(i \times j)}$. The шт stand for the number certain of elements in each dimension. To better understand the structure of tensors, we can look at their subfields such as fibers and slices. Fibers defined by fixing all but one index and given as $\underline{X}_{ij:}$, $\underline{X}_{i:k}$, $\underline{X}_{:jk}$

Vectorization is the process of transforming a given matrix into a vector by vertically stacking the columns of matrix $\mathrm{X} \in R^{(i \times j)}$. The final vector contains every component of the initial matrix; therefore, its dimension will be $(i \times j, 1)$. It can be helpful when we need to restructure the data for specific mathematical operations or algorithms that demand vector inputs. The vectorization of a matrix X is represented as *vec(X)*:

$$vec(X) = \left(x_{11}, x_{21}, \dots, x_{ij}\right)^{T} \tag{1}$$

Matricization is the process of rearranging an N-order tensor into a matrix. Analogous to vectorization, matricization is useful when working with algorithms that need matrix inputs. The mode-n matricization of a tensor, also known as unfolding or flattening, is indicated as $\underline{X}(n)$. In this process, the mode-n fibers of $\underline{X}$ are converted into the columns of $\underline{X}(n)$.

Rank – one tensor. When a higher-order tensor $\underline{X} \in R^{(i_1 \times .. \times i_n)}$ can be represented as outer product of N vectors, it means that it is a rank-one tensor. The 3-order $\underline{X} \in R^{(i \times j \times k)}$ rank-one tensor can be expressed as follows:

$$\underline{X} = u \circ v \circ w \tag{2}$$

In this context, the outer vector product is denoted as the symbol " $\circ$ ". Figure 1 shows a visual illustration of the rank-one idea. By extending this concept to the general n-order tensor:

$$\underline{X} = u^{(1)} \circ u^{(2)} \circ \dots \circ u^{(n)}, \text{with } x_{j_1 j_2 \dots j_n} = u_{j_1}^{(1)} u_{j_2}^{(2)} u_{j_3}^{(3)} \dots u_{j_n}^{(n)} \tag{3}$$

This represents that the corresponding elements from the related vectors are multiplied to create each tensor component.

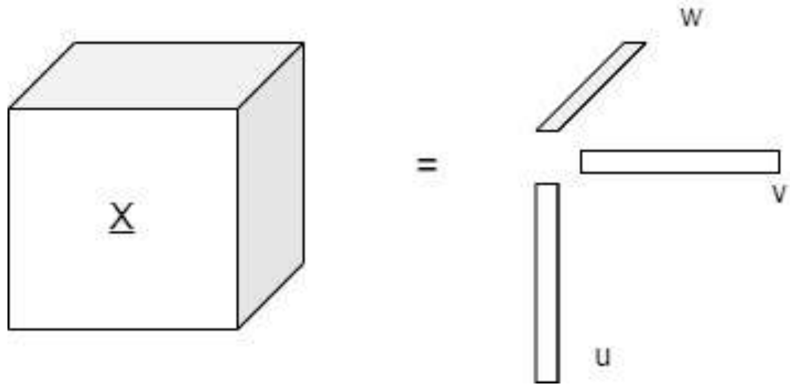


Figure 1 – Rank-one 3-order tensor.

Tensor rank. Tensor rank is the least number of rank-one tensors needed to produce X through their summation, given as $r = rank(\underline{X})$. Therefore, a 3-order rank-r tensor can be written as:

$$\underline{X} = \sum_{i=1}^{r} \lambda_i u_i \circ v_i \circ w_i = [\lambda; U, V, W] \tag{4}$$

The general n-order form is provided as follows:

$$\underline{X} = \sum_{i=1}^{r} \lambda_i u_i^{(1)} \circ u_i^{(2)} \circ \dots \circ u_i^{(n)} = \left[\lambda; U^{(1)}, U^{(2)}, U^{(3)}, \dots, U^{(n)}\right] \tag{5}$$

The factor matrices in tensor decomposition are constructed by placing the combinations of vectors from the rank-one components as columns. Therefore, the factor matrices $U^j, j = 1, \dots, n$ takes the shape:

$$U = [u_1, u_2, u_3, \dots, u_r] \tag{6}$$

Matrix operations. To comprehend the ideas and calculations of tensor decomposition, it is essential to grasp these matrix operations:

1. *Kronecker product.* The Kronecker product expands the concept of the vector outer product to matrices. This operation between two matrices $U \in R^{(i*j)}$ and $V \in R^{(k*l)}$ can be described as follows:

$$\mathrm{U} \otimes \mathrm{V} = \begin{bmatrix} u_{11}V & u_{12}V & \cdots & u_{1J}V \\ u_{21}V & u_{22}V & \cdots & u_{2J}V \\ \vdots & \vdots & \ddots & \vdots \\ u_{I1}V & u_{I2}V & \cdots & u_{IJ}V \end{bmatrix}$$

$$= [u_1 \otimes v_1 \ u_1 \otimes v_2 \ \cdots \ u_J \otimes v_{L-1} \ u_J \otimes v_L] \qquad (7)$$

1. *Khatri-Rao product.* The outcome of the Khatri-Rao product of two matrices $U \in R^{(i*j)}$ and $V \in R^{(k*j)}$ is a matrix with the size $(i * k, j)$. It is defined by:
2.

$$U \odot V = [u_1 \otimes v_1 \ u_1 \otimes v_2 \ \cdots \ u_K \otimes v_K] \qquad (8)$$

3. *Hadamard product.* A elementwise product of two same-sized matrices is known as the Hadamard product. Given two matrices $U \in R^{(i*j)}$ and $V \in R^{(i*j)}$ is of size *(i, j)*, their Hadamard product is represented by *U*V.* The result is also matrix with the same size *(i, j)* and defined by:

$$U * V := \begin{bmatrix} u_{11}v_{11} & u_{12}v_{12} & \cdots & u_{1J}v_{1J} \\ u_{21}v_{21} & u_{22}v_{22} & \cdots & u_{2J}v_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ u_{I1}v_{I1} & u_{I2}v_{I2} & \cdots & u_{IJ}v_{IJ} \end{bmatrix} \qquad (9)$$

The approach used in this work significantly differs from standard link prediction methods that proceed without dimensionality reduction: we use tensor decomposition to map 3-order tensor to several 2-order tensors and then apply time-series forecasting methods to solve the task. The main idea of this approach is based on the work of Acar et al. [16] and was extended with CP decomposition algorithms. Firstly, we capture temporal trends present in the data using time factor W derived from CP decomposition. Two alternative algorithms are used for CP decomposition: Jennrich's algorithm and ALS. Then we employ time-series forecasting methods to the temporal factor matrix W to predict future points in time while node factors U and V remain unchanged. Time-series forecasting was done using exponential smoothing and Long Short-Term Memory (LSTM), which has been gaining popularity in making forecasts in recent years. Finally, we can reconstruct the tensor with predicted links in the following *T* time instants by extracted factors U, V, and extrapolated factor W. Figure 2 shows the link prediction proposed approach's block diagram.

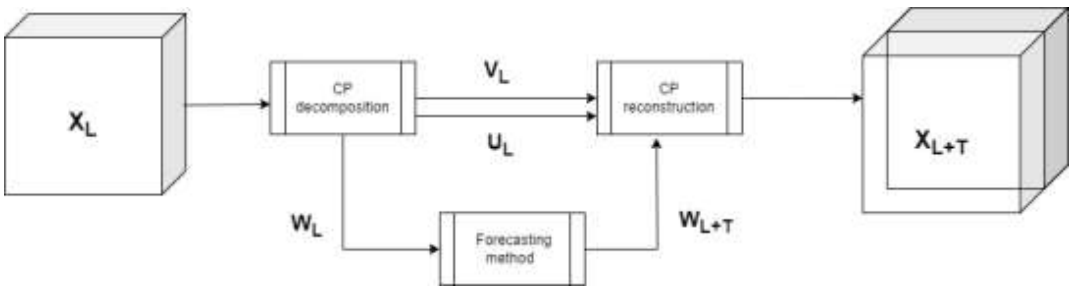


Figure 2 – The proposed approach's block diagram

Canonical Polyadic Decomposition. CP decomposition was first proposed by Hitchcock [1] in 1927. The fundamental idea of CP decomposition is to represent a tensor as a sum of rank-one tensors, where each rank-one tensor corresponds to a latent factor. The 3-order CP decomposition case is formalized as follows:

$\min \|\underline{X} - \underline{\hat{X}}\|$, where $\underline{X} \in R^{(i \times j \times k)}$ and

$$\underline{\hat{X}} = \sum_{i=1}^{r} u_i \circ v_i \circ w_i = [\![U, V, W]\!]$$

This concept is illustrated in Figure 4.

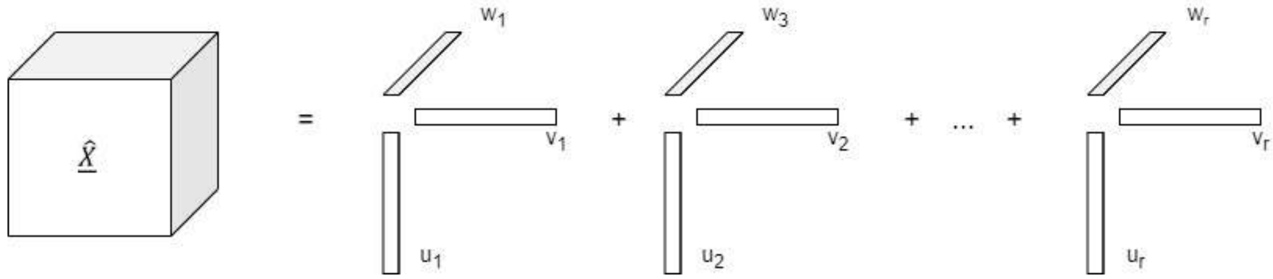


Figure 3 – Canonical Polyadic Decomposition (CP).

The factor matrices in CP decomposition are composed of the merged vectors derived from the rank-one tensors. The matricized versions of (10) are:

$$\hat{\underline{X}}_{(1)} = (W \odot V)U^T$$

$$\hat{\underline{X}}_{(2)} = (W \odot V)V^T$$

$$\hat{\underline{X}}_{(3)} = (W \odot V)W^T \quad (11)$$

In general case:

$$\hat{\underline{X}}_{(k)} = \Lambda\left(U^{(n)} \odot \dots \odot U^{(k+1)} \odot U^{(k-1)} \odot \dots \odot U^{(1)}\right)U^{(k)T} \quad (12)$$

where: $\Lambda = \mathrm{diag}(\lambda)$

The CP decomposition can be computed using a variety of algorithms. Here, we concentrate on the Alternating Least Squares (ALS) and Jennrich's algorithm.

Jennrich's Algorithm. Using Jennrich's algorithm, we can recover the factor matrices U, V, and W in (10). This straightforward approach was first disclosed in a paper by Harshman [6], with the author crediting Dr. Robert Jennrich. When the tensor components are guaranteed to be orthogonal, this algorithm stays effective.

1. $\underline{X} \in R^{m \times n \times p}$, choose a random unit-length (or Gaussian) $a, b \in R^p$ and get $\underline{X}_a, \underline{X}_b$:

$$\underline{X}_a = \sum_{i=1}^{p} a_i \underline{X}(:,:,i) \quad (14)$$

Where $\underline{X}_a$ and $\underline{X}_b$ can be formulated as follows:

$$\underline{X}_a = \sum_{i=1}^{r} (u_i \circ v_i)\langle w_i, a\rangle \quad (15)$$

$$\underline{X}_b = \sum_{i=1}^{r} (u_i \circ v_i)\langle w_i, b\rangle \quad (16)$$

1. Compute the eigen-decomposition of $\underline{X}_a(\underline{X}_b)^{\dagger}$ and $\underline{X}_b(\underline{X}_a)^{\dagger}$. Where $\underline{X}_a = UD_aV^T$ and $\underline{X}_b = UD_bV^T$, $D_a = diag(\{\langle w_i, a\rangle\}_i)$, $D_b = diag(\{\langle w_i, b\rangle\}_i)$ and we can get:

$$\underline{X}_a(\underline{X}_b)^{\dagger} = UD_aV^T(V^T)^{\dagger}D_b{}^{\dagger}U^{\dagger} = UD_aD_b{}^{\dagger} \quad (17)$$

$$\underline{X}_b(\underline{X}_a)^{\dagger} = V^T D_b U U^{\dagger} D_a{}^{\dagger} (V^T)^{\dagger} = V^T D_b D_a{}^{\dagger} (V^T)^{\dagger} \quad (18)$$

where the columns of U and V are u_i and v_i respectively.

2. Given u_i and v_i, we can solve the linear system of equations to find w_i and finally get the tensor factor matrices U, V, W.

Alternating Least Squares Algorithm. The ALS algorithm is an efficient approach to computing CP decomposition. The main idea of this algorithm is to fix all factor matrices aside from one and then optimize the non-fixed factor matrix. Each factor matrix goes through this process repeatedly until a stopping criterion is satisfied, signifying convergence or obtaining the required level of approximation. The steps of the ALS algorithm for 3-order tensor:

$$U \leftarrow arg\min_{U} \left\| \underline{X}_{(1)} - (W \odot V) U^T \right\|$$

$$V \leftarrow arg\min_{V} \left\| \underline{X}_{(2)} - (W \odot U) V^T \right\|$$

$$W \leftarrow arg\min_{W} \left\| \underline{X}_{(3)} - (V \odot U) W^T \right\| \quad (19)$$

The optimal solution for the minimization is obtained by:

$$\hat{U} = \underline{X}_{(1)}[(W \odot V)^T]^{\dagger} = \underline{X}_{(1)}(W \odot V)(W^T W * V^T V)^{\dagger}$$

$$\hat{V} = \underline{X}_{(2)}[(W \odot U)^T]^{\dagger} = \underline{X}_{(2)}(W \odot U)(W^T W * U^T U)^{\dagger}$$

$$\hat{W} = \underline{X}_{(3)}[(V \odot U)^T]^{\dagger} = \underline{X}_{(3)}(V \odot U)(V^T V * U^T U)^{\dagger} \quad (20)$$

Exponential smoothing. An exponential smoothing time series forecasting method uses weighted averages of previous observations. Recent data points are given more weight while the significance of earlier observations is gradually reduced. An exponential decay factor is used to produce this weighting technique, assigning more weight to recent observations. The formula for exponential smoothing is as follows:

$$s_t = \beta x_t + (1 - \alpha) s_{t-1} \quad (21)$$

where β - smoothing parameter, s_t – smoothed value, x_t - observed value

In our approach for link prediction, we use simple exponential smoothing to predict future values of the temporal factor matrix W of decomposed tensor. The formula of exponential smoothing for factor matrix W can be expressed as:

$$W_{t+1,i} = \beta W_{t,i} + \beta(1-\beta) W_{t-1,i} + \beta(1-\beta)^2 W_{t-2,i} + \cdots + \beta(1-\beta)^3 W_{t-3,i} + \ldots + (1-\beta)^i W_{1,i} \quad (22)$$

Where: $i = 1, \ldots, R$ and $0 < \beta < 1, t = 1, .., L + T$, T – time period to predict

By utilizing the extrapolated factor matrix W, we can reconstruct the observed tensor by estimating the potential links that may be formed within L timestamps, as indicated by formula (23):

$$\hat{\underline{X}}_{L+T} = \sum_{i=1}^{r} u_i \circ v_i \circ w_i = [\![U_L, V_L, W_{L+T}]\!] \quad (23)$$

Bidirectional Long Short-Term Memory. LSTM is an effective and versatile tool for learning activity patterns. It has three layers: an input layer, a hidden layer, and an output layer, just like other neural network designs. In our study, we use Bidirectional LSTM(BiLSTM) as a forecasting method for temporal factor matrix W retrieved from CP decomposition. BiLSTM analyzes the historical context of the time series by performing a forward pass on the historical observations. By examining the future observations made through the backward pass, it also takes the future context into account. BiLSTM can offer a more thorough insight into the temporal patterns and trends in the data by merging these two information streams. To maximize prediction accuracy by reducing the difference between the expected values and the actual observations, the model learns to modify its parameters throughout training. The backward pass aids in learning long-term dependencies and identifying future trends while also providing helpful information for gradient computing. The employed architecture has two hidden BiLSTM layers, and the number of epochs is 300 with the ADAM optimizer. The algorithm of temporal link prediction via tensor decomposition with BiLSTM:

1. CP decomposition of observed data with rank-R
2. Get factor matrices $[\![U_L, V_L, W_L]\!]$
3. Train each column of the temporal factor W_L with BiLSTM and predict W_{L+t}, where $t=1,..,T$
4. Concatenate W_L and W_{L+t}
5. Reconstruct CP decomposition with formula (23)

Results and Discussion

The dataset we examine for temporal link prediction is downloaded from WSDM 2022 Cup. This dataset represents a user-item time-evolving network of Amazon. After the preprocessing stage, the data is represented as a 3-order tensor. The first dimension (i) corresponds to the user node, the second dimension (j) represents the item node, and the third dimension (k) captures the date of interaction. For evaluation, we split the data into a training set and a test set, with 80% of the data allocated for training and 20% for testing.

We use the area under the receiver operating characteristic curve (AUC) as a metric to evaluate the performance of our methods in temporal link prediction. AUC is selected due to its robustness in handling imbalanced data, which is crucial in our case as the training dataset contains a small fraction (less than 0.5%) of actual links compared to all possible links. Firstly, in order to assess the performance of tensor decomposition and its reconstruction, only the CP decomposition part of the model was evaluated.

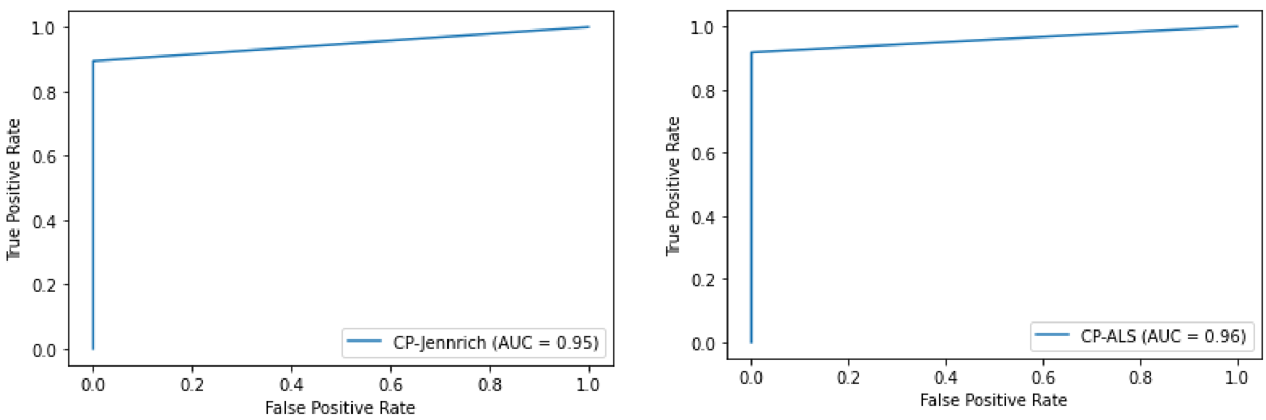


Figure 4 – AUC and ROC of the CP decomposition algorithms

Figure 4 displays the ROC curves, which provide a comprehensive view of the tensor decomposition performance. It can be seen that Jennrich's algorithm demonstrates slightly lower performance than the ALS algorithm for 0.01 (0.95 vs. 0.96).

In Figure 5, the bar chart provides valuable insights into the link prediction performance using AUC as the evaluation metric. Among all the methods, Jennrich's CP decomposition algorithm with BiLSTM algorithm achieves the highest AUC score (0.95). However, the ALS algorithm with the same forecasting model yields the lowest AUC score (0.83). The ROC of these models is presented in Figure 8. But with exponential smoothing in the prediction part, the AUC score of the method increases by 0.5 (0.88). With two alternative forecasting methods in the prediction part, Jennrich's algorithm performs exceptionally well in link prediction with the time-evolving dataset. It can be assumed that.

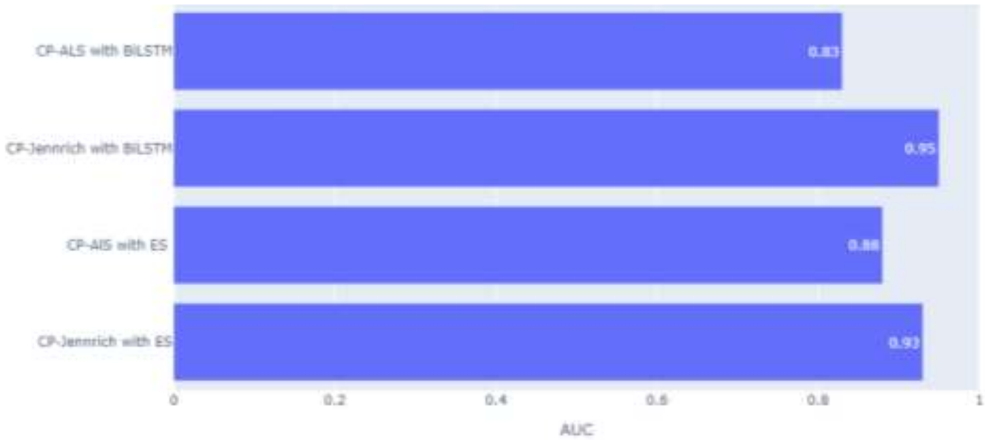
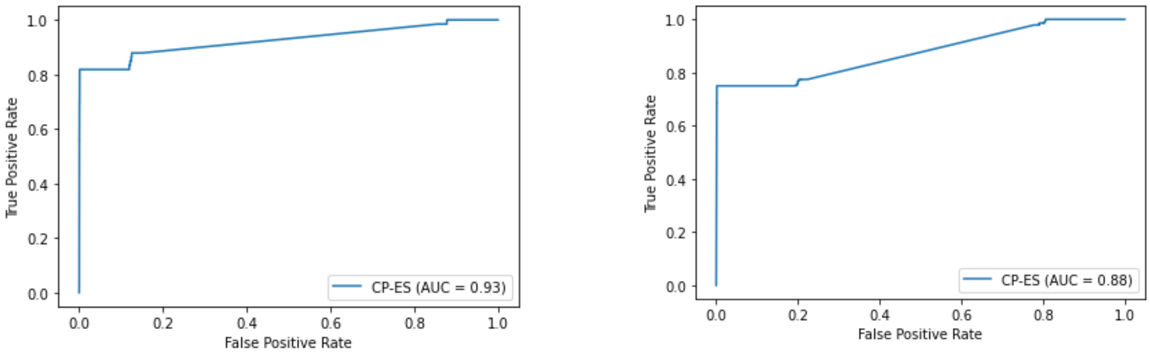


Figure 5 – AUC of link prediction models

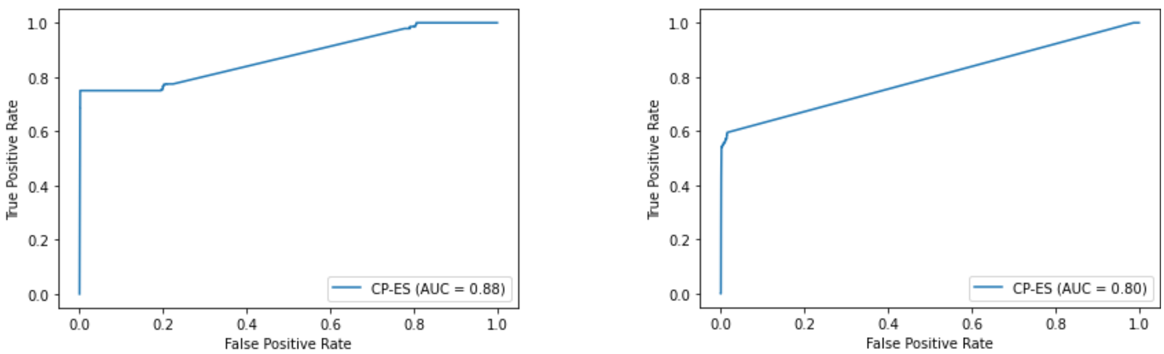
Jennrich's CP decomposition effectively retrieves latent temporal trends from an observed tensor to the temporal factor matrix of the CP decomposition where the prediction part is held.



(a) $T = 1$ (b) $T = 30$

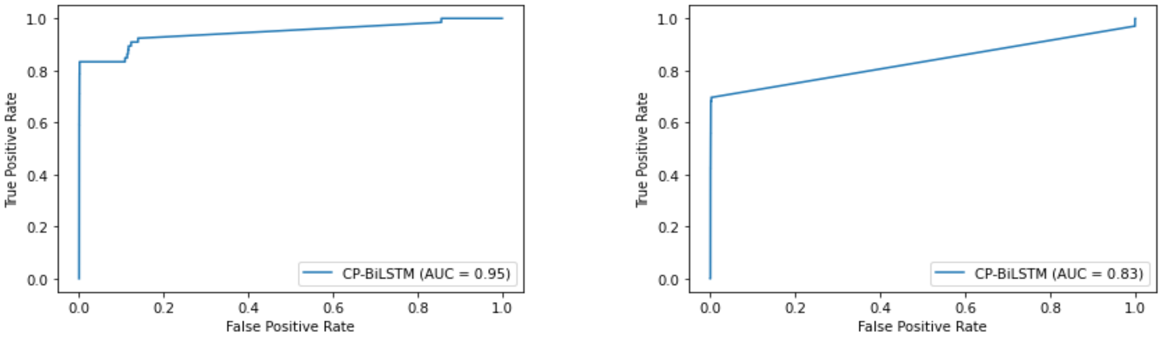
Figure 6 – AUC and ROC of the Jennrich'algorithm with exponential smoothing

The impact of increasing T on the change in AUC is illustrated in separate graphics, as shown in Figure 6-7. The outcomes measured by ROC and AUC of Jennrich's algorithm with exponential smoothing are depicted in Figure 6. The AUC of this model decreases from 0.93 to 0.88 when T is increased from 1 to 30, indicating a relatively lower performance as the time period increases. In Figure 7, a similar assumption can be applied to the results of the ALS decomposition with a simple exponential smoothing as a forecasting method.



(a) $T = 1$ (b) $T = 30$

Figure 7 – The ALS algorithm with exponential smoothing (AUC and ROC)



(a) Jennrich’s algorithm (b) ALS algorithm

Figure 8 – The CP algorithms with BiLSTM (AUC and ROC)

Conclusion

In this work, we present a method for link prediction in large-scale time-evolving networks, which completely differs from standard graph-based methods. This method is a combination of tensor decomposition and time-series forecasting. The dataset that we used to evaluate our approach is derived from WSDM. In data preprocessing, the dataset is converted to a three-way tensor. In the tensor decomposition part, the observed tensor is decomposed to two-way tensors, which are factor matrices of each mode that give a relative pattern of the network. As a tensor decomposition model, we used two alternative algorithms of CP decomposition such as Jennrich’s algorithm and the ALS algorithm. The results show that Jennrich’s algorithm is more efficient in problems considering the temporal trend. In forecasting, we utilized the third mode factor matrix of decomposed tensor and predicted new links via BiLSTM and exponential smoothing. By comparing the AUC of each method, we conclude that the combination of the Jennrich algorithm and BiLSTM shows the best performance. In future work, we aim to investigate other decomposition algorithms in link prediction, such as Tucker and Tensor Train decomposition methods with a dataset presented as a multi-way tensor.

References

1 Hitchcock Frank L. (1927) The expression of a tensor or a polyadic as a sum of products, Journal of Mathematics and Physics, 6.1-4, pp. 164–189.

2 Sidiropoulos Nicholas D. et al. (2017) Tensor decomposition for signal processing and machine learning, IEEE Transactions on Signal Processing, 65.13, pp. 3551–3582.

3 Kolda Tamara G. and Brett W. Bader (2009) Tensor decompositions and applications, SIAM review, 51.3, pp. 455–500.

4 Tucker, Ledyard R. (1966) Some mathematical notes on three-mode factor analysis, Psychometrika, 31.3: 279–311.

5 Carroll, J. Douglas and Jih-Jie Chang (1970) Analysis of individual differences in multidimensional scaling via an N-way generalization of “Eckart-Young” decomposition, Psychometrika 35.3, pp. 283–319.

6 Harshman, Richard A. (1970) Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multimodal factor analysis, pp. 1–84.

7 Bader, Brett W., Richard A. Harshman and Tamara G. Kolda. Temporal analysis of semantic graphs using ASALSAN, Seventh IEEE international conference on data mining (ICDM 2007).

8 Nickel, Maximilian, Lorenzo Rosasco and Tomaso Poggio. (2016) Holographic embeddings of knowledge graphs, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1.

9 Ma, Yunpu, Volker Tresp and Erik A. Daxberger. (2019) Embedding models for episodic knowledge graphs, Journal of Web Semantics 59: 100490.

10 Acar E., Dunlavy D.M., Kolda T.G. (2009) Link prediction on evolving data using matrix and tensor factorizations, IEEE International conference on data mining workshops, pp. 262–269.

11 Ermiş, Beyza, Evrim Acar and A. Taylan Cemgil. (2015) Link prediction in heterogeneous data via generalized coupled tensor factorization, Data Mining and Knowledge Discovery, 29, pp. 203–236.

12 Xiong, Liang et al. (2010) Temporal collaborative filtering with Bayesian probabilistic tensor factorization, Proceedings of the 2010 SIAM international conference on data mining, Society for Industrial and Applied Mathematics.

13 Gao S., Denoyer L., Gallinari P. (2010) Tensor decomposition model for link prediction in multi-relational networks, 2nd IEEE International Conference on Network Infrastructure and Digital Content, IEEE, pp. 298–302.

14 Chi, Eric C. and Tamara G. Kolda. (2012) On tensors, sparsity, and nonnegative factorizations, SIAM Journal on Matrix Analysis and Applications, 33.4, pp. 1272–1299.

15 de Araujo, Miguel Ramos, Pedro Manuel Pinto Ribeiro and Christos Faloutsos. Tensorcast: Forecasting with context using coupled tensors (best paper award), 2017 IEEE International Conference on Data Mining (ICDM).

Information about author

Aliturliyeva Albina Erbolatovna
Master's student in Data Science, Kazakh-British Technical University, 59, Tole bi street, Almaty, 050000, Kazakhstan
ORCID ID 0009-0002-6758-5608
E-mail: a_aliturliyeva@kbtu.kz

Автор туралы мәлімет

Алитурлиева Альбина Ерболатовна
Деректер ғылымының магистранты, Қазақстан-Британ техникалық университеті, Төле би көш., 59, 050000, Алматы қ., Қазақстан
ORCID ID 0009-0002-6758-5608
E-mail: a_aliturliyeva@kbtu.kz

Информация об авторе

Алитурлиева Альбина Ерболатовна
Магистр наук о данных, Казахстанско-Британский технический университет, ул. Толе би, 59, 050000, г. Алматы, Казахстан
ORCID ID 0009-0002-6758-5608
E-mail: a_aliturliyeva@kbtu.kz