Journal of Information Processing Vol.27 752–761 (Nov. 2019)

[DOI: 10.2197/ipsjjip.27.752]

Regular Paper

Time Series Link Prediction Using NMF

Faith Mutinda[3,a)] Atsuhiro Nakashima[4,b)] Koh Takeuchi[2,c)] Yuya Sasaki[1,d)]
Makoto Onizuka[1,e)]

Received: March 10, 2019, Accepted: July 4, 2019

Abstract: Data in many fields such as e-commerce, social networks, and web data can be modeled as bipartite graphs, where a node represents a person and/or an object and a link represents the relationship between people and/or objects. Since the relationships change with time, data mining techniques for time series graphs have been actively studied. In this paper, we study the problem of predicting links in the future graph from historical graphs. Although various studies have been carried out on link prediction, the prediction accuracy of existing methods is still low because it is difficult to capture continuous change with time. Therefore, we propose a new method that combines non-negative matrix factorization (NMF) and a time series data forecasting method. NMF extracts the latent features while the forecasting method captures and predicts the changes of the features with time. Our method can predict hidden links that do not appear in historical graphs. Our experiments with real datasets show that our method has a higher prediction accuracy compared to existing methods.

Keywords: link prediction, forecasting, non-negative matrix factorization (NMF)

1. Introduction

The amount of data in many applications such as e-commerce, social networks, and the web is increasing rapidly. Extracting useful information from these big data can improve the quality of applications or services and can create new profits. Data from these applications can be modeled as bipartite graphs, where nodes represent objects and links represent the relationship between the objects. For example, if a customer purchases an item from an e-commerce site, we can represent this relationship using a link between two nodes, those represent a customer and an item. Graphs can be used in various data mining tasks, such as detecting hidden groups, detecting missing links, and ranking objects [8].

Real world data such as product sales is dynamic because their relationships change with time. Therefore, it is important to take into consideration the temporal changes in the links so as to predict the links at a future time. By predicting the future link structure, we can predict future trends and behaviors. Therefore, e-commerce sites such as amazon can recommend products which match customers tastes and increase their sales and customer satisfaction [7], [13], [14], [18], [19]. We formally define our problem as follows.

Problem definition: **(Time Series Link Prediction)**. Given a set of time series bipartite graphs $G_1 = (V_1, V_2, E_1), \cdots, G_T = (V_1, V_2, E_T)$ from time 1 to T, where V_1 and V_2 represent the sets of nodes and E_t represents the set of weighted links at each time t, the task is to predict the set of binary links E_{T+1} in the future graph $G_{T+1} = (V_1, V_2, E_{T+1})$ at time $T + 1$.

Various link prediction methods have been proposed, but the prediction accuracy is still low [13] for time series graphs. In order to improve the prediction accuracy, it is important to model the features of the graph properly and capture how these features change with time. Non-negative matrix factorization (NMF) method can be used in modelling graph features because it extracts latent features effectively and therefore helps in understanding the underlying link structure [20]. In addition, data forecasting methods such as Holt-Winters, Vector Auto Regressive (VAR) and recurrent neural networks are commonly used in time series data forecasting. Holt-Winters method can capture periodicity or seasonal fluctuations in time series data [3], [11]. VAR is able to capture linear relationships among variables in multivariate time series data [21]. Recurrent neural networks especially Long-Short Term Memory (LSTM) have recently gained popularity in time series data forecasting because they can capture non-linear relationships in sequential data, as well as their capability to remember past information and take this past information into consideration while predicting future values [15], [25].

In this paper, we propose a new time series link prediction method that combines NMF and time series data forecasting methods for predicting the links in the future graph from historical graphs. Our method first applies NMF to extract the latent features from the matrix representing the structure of each historical graph at time t ($1 \leq t \leq T$) and, then applies a forecasting

1 Graduate School of Information Science and Technology, Osaka University, Suita, Osaka 565–0871, Japan
2 NTT Communication Science Laboratories, Kyoto 619–0237, Japan
3 Nara Institute of Science and Technology, Ikoma, Nara 630–0192, Japan
4 IBM, 19–21 Nihonbashi, Chuo, Tokyo 103–8510, Japan
a) mutinda.faith_wavinya.mz2@is.naist.jp
b) Atsuhiro.Nakashima@ibm.com
c) koh.t@acm.org
d) sasaki@ist.osaka-u.ac.jp
e) onizuka@ist.osaka-u.ac.jp

method to predict the latent features at time $T+1$ from the time series of the decomposed matrices from time 1 to T. In addition, we propose an extension to our method so as to improve the preciseness of the prediction; future prediction by ensemble learning. By employing ensemble learning, multiple models can be created by changing the parameters and they are combined to improve the prediction accuracy and avoid overfitting [5]. We evaluate our method by comparing it with other methods using real datasets and our method showed an improvement in the prediction accuracy.

The remainder of this paper is organized as follows. Section 2 overviews the preliminaries of this work. Section 3 describes the details of our proposed method. Section 4 reviews the results of our experiments. Section 5 describes related work. Section 6 provides our brief conclusion.

2. Preliminaries

In this section, we introduce the background of the methods proposed in this paper.

2.1 Time Series Graph

Let $G = (V_1, V_2, E)$ be a bipartite graph, where V_1 and V_2 are sets of nodes and $E \in V_1 \times V_2$ is a set of weighted links. The numbers of nodes in V_1 and V_2 are denoted by N and M, respectively. The bipartite graph G is represented in a form of two-dimensional adjacency matrix $\boldsymbol{X}$ of size $M \times N$. If there exists a link between nodes i and j, the (i, j) component of the matrix is assigned the weights of the links and zero if there is no link. The graph structure over a period of time is considered. We denote $\boldsymbol{X}^{(t)}$ as an adjacency matrix at each time t. We call the set of graphs over time range *time series graph*.

2.2 NMF: Non-negative Matrix Factorization

NMF decomposes a matrix of non-negative values to two matrices of low dimensions such that they do not include negative values [16]. This restriction enables NMF to provide different decomposition results compared to other matrix decomposition methods such as singular value decomposition (SVD) or principal component analysis (PCA). NMF results are easy to interpret, especially in tasks where the underlying features are interpreted as non-negative.

In NMF, a non-negative value matrix $\boldsymbol{X}$ of size $M \times N$ is decomposed into two non-negative matrices, $\boldsymbol{U}$ and $\boldsymbol{V}$ of size $M \times K$ and $K \times N$ respectively, such that $\boldsymbol{X} \approx \boldsymbol{UV}$. K is the base number of NMF and is an arbitrary parameter. In this paper, we use the multiplicative update rules for NMF where matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are initialized with random non-negative values. We use an iterative method to update matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ such that the divergence between the original matrix $\boldsymbol{X}$ and $\boldsymbol{UV}$ is minimized. The datasets in our experiments are expected to follow the Poisson distribution, therefore we use the Kullback- Leibler (KL) divergence. The update rules based on KL divergence are defined by the Eqs. (1), (2) below as discussed by Ref. [17].

$$u_{ik} \leftarrow u_{ik} \frac{\sum_j \frac{x_{ij} v_{kj}}{\sum_k u_{ik} v_{kj}}}{\sum_j v_{kj}}, \tag{1}$$

$$v_{kj} \leftarrow v_{kj} \frac{\sum_i \frac{x_{ik} u_{ik}}{\sum_k u_{ik} v_{kj}}}{\sum_i u_{ik}}. \tag{2}$$

The obtained decomposed matrices can be regarded as matrices in which the features of the original matrix are reduced into a low dimension, that is, the features of each axis of the original matrix are reduced into groups with K number of components.

2.3 Forecasting Approach

Various time series data forecasting methods have been studied [22]. Prediction of how a link changes between certain nodes is a univariate time series prediction task, and therefore future links can be predicted by solving all the node combinations.

Exponential smoothing is a simple univariate time series prediction method. In the first-order exponential smoothing method, prediction is performed on the observations $y_1, \cdots, y_t$ such that the most recent observation is given more weight.

$$P_{t+1} = \alpha y_t + (1-\alpha) P_t, \tag{3}$$

where α is a learning coefficient which takes values $0 \le \alpha \le 1$ and P_{t+1} is the predicted value for one time period ahead. However, the first-order exponential smoothing prediction formula does not capture change in trends such as increasing and decreasing trends effectively. In the second-order exponential smoothing method, prediction is performed with the addition of the term b_t which captures the trend such that

$$P_{t+1} = \alpha y_t + (1-\alpha) P_t + b_t \text{ and} \tag{4}$$

$$b_t = \beta (P_t - P_{t-1}) + (1-\beta) b_{t-1}, \tag{5}$$

where β is the trend coefficient.

Holt-Winters is a forecasting method suitable for time series data that has seasonality and trend [3], [11]. Seasonality means that the time series data has trends that repeat every m cycles. There are two variations of the Holt-Winters forecasting technique; additive and multiplicative methods. These variations differ in the seasonality component. The additive method is suitable when the seasonal variations in the series are roughly constant. On the other hand, the multiplicative method is suitable when the seasonal variations change proportionally to the level (average value) of the series. In this paper we choose the additive method because the change in our datasets is fairly constant. The additive Holt-Winters method consists of three smoothing equations and a forecasting equation as shown below.

$$l_t = \alpha (y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}). \tag{6}$$

$$b_t = \beta (l_t - l_{t-1}) + (1-\beta) b_{t-1}. \tag{7}$$

$$s_t = \gamma (y_t - l_{t-1} - b_{t-1}) + (1-\gamma) s_{t-m}. \tag{8}$$

$$y_{t+h} = l_t + h b_t + s_{t-m+h}. \tag{9}$$

$y_1, y_2, \cdots, y_t$ are the observed values and m is the seasonality parameter which represents the length of the seasonal cycle, for example, $m = 3$ for quarterly data, and $m = 12$ for monthly data.

α, β, γ are smoothing parameters. l_t is the smoothed estimate of the level at time t, b_t is the smoothed estimate of the change in the trend at time t and s_t is the smoothed estimate of the seasonal component at time t. The smoothing Eqs. (6)–(8) minimize the squared error and the forecast, (y_{t+h}) at h time periods ahead is calculated as shown in Eq. (9).

Auto Regressive (AR) models are also widely used in time series data forecasting [26], [28]. The AR model is used in various research fields, for example, in social economics, Simultaneous Auto Regressive (SAR) model is commonly used in the analysis of spatial data since it incorporates spatial auto-correlations into regression models [1].

The Vector Auto Regressive (VAR) model is popular in prediction of future observations in multivariate time series data. It extends the AR model to the multivariate setting by modelling linear interdependencies between multiple features in time series data [21]. Let $y_t \in \mathbb{R}^n$ denote a multivariate time series with time lag set L and weights $A_l \in \mathbb{R}^{n \times n}$, VAR approximates y_t as,

$$y_t = \sum_{l \in L} A_l y_{t-l} + \epsilon_t, \tag{10}$$

where ϵ_t is Gaussian noise and l is the lag of the model which determines the extent to which current time period data relies on data from previous time periods. VAR has also been used in space-time prediction problems of sensor data by combining AR model with tensor decomposition [26].

Recently, Recurrent Neural Networks (RNN) especially Long-Short Term Memory (LSTM) have also been studied for time series data forecasting [15]. This is because of their potential to capture long term dependencies in sequential data. The standard LSTM model takes a sequence of vectors $y_1, y_2, \cdots, y_t \in \mathbb{R}^n$ as input and produces a single output vector $\hat{y}_{t+1} \in \mathbb{R}^n$, where $\hat{y}_{t+1}$ is the predicted value for one time period ahead. The key concept of the LSTM model is a hidden state known as the cell state and three gates; input, forget and output gates. The cell state acts as a memory and transfers relevant information through the network. The input gate decides whether or not to accept new input, the forget gate decides what information to keep or delete and the output gate decides the output of the current time period. LSTM has the ability to remove old information or add new information to the cell state through the gates. In addition, the gates have a sigmoid activation which gives outputs values between 0 and 1. These values determine how the model updates or forgets information.

At time t, an LSTM cell receives the output of the previous block h_{t-1}, and the current input y_t. Then it calculates the value of the forget gate as,

$$f_t = \sigma(w_f[h_{t-1}, y_t] + b_f), \tag{11}$$

where w_f, b_f are weight and bias parameters respectively, σ is a sigmoid function and f_t is the gate output. The output of the forget gate determines what to ignore from the previous cell state. The input gate determines what parts of the current input should be added to the cell state as shown below.

$$i_t = \sigma(w_i[h_{t-1}, y_t] + b_i), \tag{12}$$

$$\hat{c}_t = \tanh(w_c[h_{t-1}, y_t] + b_c), \tag{13}$$

where w_i, b_i are the input gate weight and bias parameters respectively and i_t is the input gate output. w_c, b_c are weight and bias parameters for selecting candidate state $\hat{c}_t$. The input gate, forget gate and candidate state determine the current state cell c_t as,

$$c_t = f_t * c_{t-1} + i_t * \hat{c}_t. \tag{14}$$

The last step which is the output gate determines the current cell state output

$$o_t = \sigma(w_o[h_{t-1}, y_t] + b_o), \tag{15}$$

$$h_t = o_t * \tanh(c_t), \tag{16}$$

where w_o, b_o are weight and bias parameters and o_t is used to determine the output h_t. h_t can be used as the final predicted output y_t.

$*$ denotes element wise vector multiplication and tanh is an activation function which predicts what part should appear as output of current LSTM unit at time t.

3. Proposed Method

In our proposed method, we use NMF to extract the underlying latent features and a time series data forecasting method to capture the change of the extracted latent features and predict future actions. In this paper we use Holt-Winters, VAR and LSTM forecasting methods discussed in Section 5 as the time series data forecasting methods.

Figure 1 shows an outline of the proposed method. $\boldsymbol{Z}$ is a three-dimensional tensor which consists of the adjacency matrices of the bipartite graphs from time 1 to time T. We consider matrices $\boldsymbol{X}^{(1)}, \cdots, \boldsymbol{X}^{(T)}$ of uniform size at each time t. NMF is applied to the matrices $\boldsymbol{X}^{(1)}, \cdots, \boldsymbol{X}^{(T)}$ to obtain $\boldsymbol{U}^{(1)}, \boldsymbol{V}^{(1)}, \cdots, \boldsymbol{U}^{(T)}, \boldsymbol{V}^{(T)}$. A forecasting method is then applied to the sequence of $\boldsymbol{U}^{(t)}$ and $\boldsymbol{V}^{(t)}$ $(1 \leq t \leq T)$ to predict the values of $\boldsymbol{U}^{(T+1)}$ and $\boldsymbol{V}^{(T+1)}$ at time $T+1$, respectively. The predicted matrix at time $T+1$ is

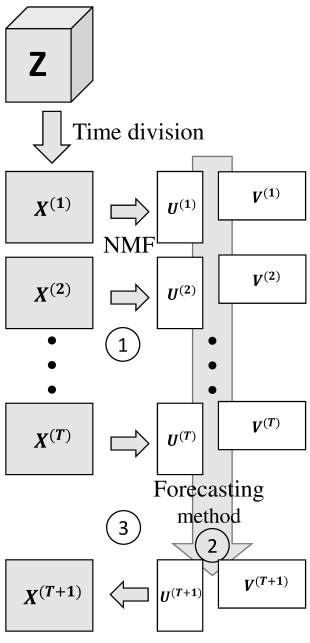


Fig. 1 Overview of proposed method: Step 1: NMF is applied to the matrices $\boldsymbol{X}^{(1)}, \cdots, \boldsymbol{X}^{(T)}$. Step 2: Forecasting method is applied to the sequence of $\boldsymbol{U}^{(t)}$ and $\boldsymbol{V}^{(t)}$ $(1 \leq t \leq T)$. Step 3: The predicted matrix at time $T+1$ is calculated as $\boldsymbol{X}^{(T+1)} = \boldsymbol{U}^{(T+1)}\boldsymbol{V}^{(T+1)}$.

Algorithm 1 Calculation of prediction matrix $\boldsymbol{X}^{(T+1)}$

Input $\boldsymbol{Z} = (\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots, \boldsymbol{X}^{(T)}), \boldsymbol{X}_{Ave}$
Output $\boldsymbol{X}^{(T+1)}$

1: **NMF** : $\boldsymbol{U}_{init}, \boldsymbol{V}_{init} \leftarrow \boldsymbol{X}_{Ave}, \boldsymbol{U}_{random}, \boldsymbol{V}_{random}$
2: **for each** $\boldsymbol{X}^{(t)} \in \boldsymbol{Z}$ **do**
3: **NMF** : $\boldsymbol{U}^{(t)}, \boldsymbol{V}^{(t)} \leftarrow \boldsymbol{X}^{(t)}, \boldsymbol{U}_{init}, \boldsymbol{V}_{init}$
4: $\boldsymbol{U}_{list} \leftarrow \boldsymbol{U}^{(t)}$
5: $\boldsymbol{V}_{list} \leftarrow \boldsymbol{V}^{(t)}$
6: **end for**
7: **Forecast** : $\boldsymbol{U}^{(T+1)} \leftarrow \boldsymbol{U}_{list}$
8: **Forecast** : $\boldsymbol{V}^{(T+1)} \leftarrow \boldsymbol{V}_{list}$
9: $\boldsymbol{X}^{(T+1)} = \boldsymbol{U}^{(T+1)}\boldsymbol{V}^{(T+1)}$

calculated as $\boldsymbol{X}^{(T+1)} = \boldsymbol{U}^{(T+1)}\boldsymbol{V}^{(T+1)}$.

It is intuitive and promising to combine NMF with forecasting approaches, however, one of the difficulties is that how we can control the latent features for each time step $(\boldsymbol{X}^{(1)}, \cdots, \boldsymbol{X}^{(T)})$ to be consistent with those in its previous time step. The detail is as follows. In NMF, the decomposition matrices are initialized with random non-negative values and then the multiplicative update rules are applied. The final decomposition matrices depend on the initial decomposition matrices. Therefore, there is no guarantee that the reduced K features appear in the same order for each decomposition matrix in $\boldsymbol{U}^{(t)}$ $(\boldsymbol{V}^{(t)})$ $(1 \le t \le T)$.

To solve this problem, we first apply NMF to the average matrix defined as $\boldsymbol{X}_{Ave}(i, j) = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{X}^{(t)}(i, j)$ to obtain $\boldsymbol{U}_{init}, \boldsymbol{V}_{init}$ which we use as the initial matrices at each time t. By using the same initial decomposition matrix at each time t, we expect that the order of the features tend to stay the same, that is, same features are likely to appear in the same position in each decomposition matrix. This ensures that the latent features are captured properly over the entire time.

Algorithm 1 shows the detail of the proposed method. The input $\boldsymbol{Z}$ is a list of same size matrices at each time t, and $\boldsymbol{X}_{Ave}$ is the average matrix. The output $\boldsymbol{X}^{(T+1)}$ is the predicted matrix at time $T + 1$. NMF is applied to the average matrix with random initial non-negative value matrices $\boldsymbol{U}_{random}, \boldsymbol{V}_{random}$ to obtain the initial decomposition matrices $\boldsymbol{U}_{init}, \boldsymbol{V}_{init}$ (line 1). NMF is then applied to the matrices at each time t, with $\boldsymbol{U}_{init}, \boldsymbol{V}_{init}$ as the initial matrices (lines 2 and 3). A list of decomposed matrices $\boldsymbol{U}_{list}, \boldsymbol{V}_{list}$ is created to which a forecasting method is applied to forecast the future values at time $T + 1$ (lines 7 and 8). The predicted matrix $\boldsymbol{X}^{(T+1)}$ is obtained by the dot product of $\boldsymbol{U}^{(T+1)}, \boldsymbol{V}^{(T+1)}$ (line 9).

3.1 Proposed Method with Ensemble Learning

Ensemble learning technique involves strategically combining several models so as to improve the stability and predictive performance. Previous studies on ensemble learning show that combining multiple individual models improves prediction accuracy compared to using a single model [5]. The idea of combining multiple models assumes that it is difficult for a single model to understand the underlying structure, but multiple models can capture different aspects of data. Ensemble learning is helpful when it is difficult to choose optimum values of parameters, and when one wants to avoid large errors [4].

The Holt-Winters seasonality parameter, m, and the number of features, K, for NMF need to be selected manually. It is difficult to search for optimum values for those parameters which give the best performance. We employ an ensemble approach, let $\boldsymbol{X}_{Km}$ denote the matrix of scores calculated for $K = 5, 10, \cdots, 100$ and $m = 1, 2, \cdots, 12$, the final matrix of ensemble scores is then calculated as,

$$\boldsymbol{X} = \sum_{K\in(5,10,\ldots,100)} \sum_{m\in(1,2,\ldots,12)} \frac{\boldsymbol{X}_{Km}}{\|\boldsymbol{X}_{Km}\|_F}, \tag{17}$$

where $\|\boldsymbol{X}_{Km}\|_F$ is the Frobenius norm for $\boldsymbol{X}_{Km}$.

Similarly, the lag parameter l for VAR needs to be selected manually. Let $\boldsymbol{X}_{Kl}$ denote the matrix of scores calculated for $K = 5, 10, \cdots, 100$ and $l = 1, 2, \cdots, 12$, the final matrix of ensemble scores is then calculated as,

$$\boldsymbol{X} = \sum_{K\in(5,10,\ldots,100)} \sum_{l\in(1,2,\ldots,12)} \frac{\boldsymbol{X}_{Kl}}{\|\boldsymbol{X}_{Kl}\|_F}, \tag{18}$$

where $\|\boldsymbol{X}_{Kl}\|_F$ is the Frobenius norm for $\boldsymbol{X}_{Kl}$.

4. Experiments

In this section, we describe experiments and their results by comparing our method proposed in Section 3 with existing link prediction methods using matrix decomposition, summarized in Section 5; CP, TSVD CT and TSVD CWT [6], [7], [12]. We also investigate the effectiveness of our variation of ensemble learning described in Section 3.

According to our problem statement in Section 1, given a set of nodes V_1 and V_2 and a set of weighted links $E_1, \cdots, E_T$, the task is to predict the set of unweighted binary links E_{T+1}. Assuming that there is a link of weight n between node i and node j at time t, we generate a three-dimensional tensor $\boldsymbol{Z}(i, j, t) = n$. In order to eliminate the influence of large values in the data, we normalize the data according to Eq. (19) which was proposed by Ref. [7].

$$\boldsymbol{Z}(i, j, t) = \begin{cases} 1 + \log(n) & n > 0 \\ 0 & n = 0. \end{cases} \tag{19}$$

Since the objective of our study is to predict whether there is a link or not between nodes i and j, the link information is represented as,

$$\boldsymbol{Y}(i, j) = \begin{cases} 1 & n > 0 \\ 0 & n = 0. \end{cases} \tag{20}$$

Our proposed method combines NMF and a time series data forecasting method, we consider three popular time series data forecasting methods; Holt-Winters, VAR and LSTM discussed in Section 2. We carry out experiments by combining NMF with each of the time series forecasting methods and evaluate their performance on our datasets. We use the average of previous time periods as the baseline method. The Holt-Winters smoothing parameters α, γ, β and seasonality component need to be decided before-hand. We use the approach discussed in [3] to establish the values for the smoothing parameters which minimize the sum of squared errors for one time step ahead forecast. The VAR lag parameter also needs to be decided before-hand; we search for a value which achieves the best performance. For LSTM, we adopt

Table 1 Dataset summary.

Dataset	Training Density	Test Density	Hidden Links Density
POS	7.48%	6.65%	0.25%
DBLP	3.52%	1.17%	0.65%

a sliding time window approach which uses multi-step lag observations as input. For example, for current time period t and a time window of 2, we use previous time steps $t-1$ and $t-2$ as the input values. We use one hidden layer and a dense layer for the output. In addition, a dropout of 0.2 is used to avoid over-fitting. In the case of TSVD, CP and NMF decomposition, instead of using a fixed value of K, we use an ensemble approach as shown below.

$$\boldsymbol{X} = \sum_{K\in(5,10,\ldots,100)} \frac{\boldsymbol{X}_K}{\|\boldsymbol{X}_K\|_F}, \tag{21}$$

where $\boldsymbol{X}_K$ is the matrix of scores calculated for $K = 5, 10, \cdots, 100$, $\boldsymbol{X}$ is the final matrix of ensemble scores and $\|\boldsymbol{X}_K\|_F$ is the Frobenius norm for $\boldsymbol{X}_K$.

We construct the receiver operating characteristic (ROC) curve and the area under the curve (AUC) measures the discrimination, that is, the ability to predict true positives and true negatives correctly. The AUC value therefore shows how well a model predicts.

We compare the performance of the methods in predicting all links and hidden links. In all link prediction, we compare how the different methods perform in predicting the links in the test dataset. On the other hand, the prediction of the hidden links addresses a difficult task, that is, how the different methods predict links that do not previously exist in the training dataset.

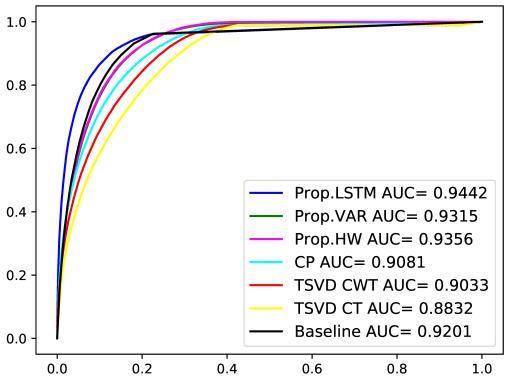
4.1 Data

We use two real datasets; a point-of-sales (POS) dataset and DBLP dataset. The point-of-sales dataset consists of supermarket sales data for a period of 24 months from July 2013 to June 2015. Supermarket data is periodic because some products such as vegetables are seasonal and special products are promoted during special events such as Christmas and Valentines. This dataset contains 25,668 customers and 113,688 items. From this dataset we extract the top 1,000 frequent customers and top 500 items with the highest number of sales. We transform the data to adjacency matrices of the bipartite graphs, and use data for the first 23 months as training set and data for the last month as the test set. The density of the training and test sets is shown in **Table 1**.

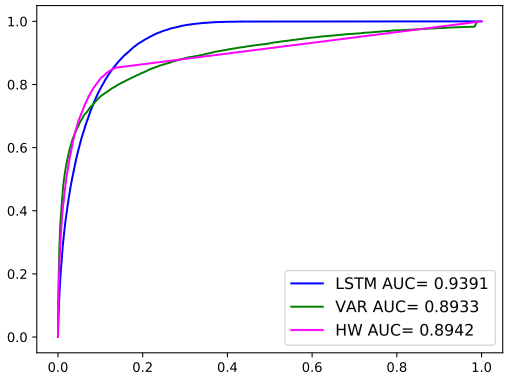
The DBLP dataset consists of publications for a period of 17 years from 1991 to 2007. The data consists of author and conference relations, when an author's paper is presented at a conference, a link is made between the author and the conference. Similar to the POS dataset, we select the top 1000 authors and 500 conferences with the highest number of publications. We use the data for the first 16 years as training set and data for the last year as the test set. The density of the training and test sets is shown in Table 1.

4.2 Results

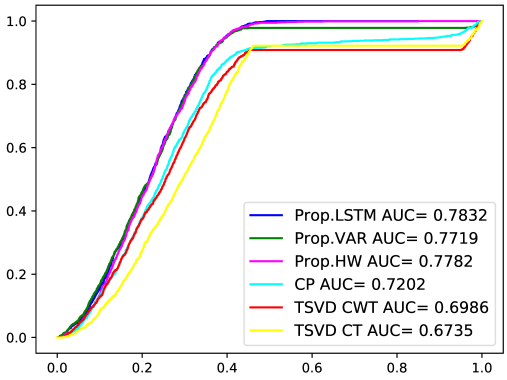
We compare the performance of the methods in terms of AUC values. We use prop.LSTM, prop.HW and prop.VAR to denote the proposed method of NMF with LSTM, Holt-Winters and VAR time series data forecasting methods, respectively.



(a) All link prediction



(b) All link prediction
Results for forecasting methods without NMF



(c) Hidden link prediction

Fig. 2 ROC curves for POS dataset comparison with existing methods.

4.2.1 Comparison with Existing Methods

Figure 2 shows the performance of the methods for the POS dataset. The proposed method achieves the highest AUC for both all link prediction and hidden link predictions. As expected the AUC values for hidden link prediction are low compared to all link prediction. This is because hidden link prediction is a challenging task which involves predicting links which do not previ-

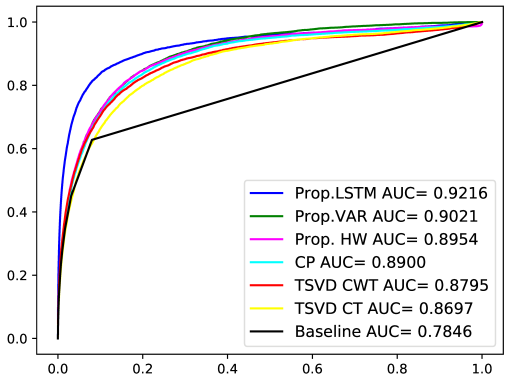
ously exist in the training set. The proposed method with LSTM forecasting achieves the best performance. This implies that LSTM's ability to capture long-term time dependencies in time series data improves link prediction. LSTM is capable of identifying complex patterns from time series data since its able to keep useful information and discard what is not useful. Note that the proposed method with Holt-Winters forecasting performed better than the proposed method with VAR forecasting. Holt-Winters forecasting method performs well with data that has seasonality and supermarket sales data is periodic because some products are seasonal and special products are promoted during special events. Further, the baseline method also performs well in all link prediction for this dataset. This is because we considered the most frequent customers and items, and in sales data it is expected that the frequent items are purchased at each time step. On the other hand, the baseline method cannot capture data patterns and therefore it cannot predict any hidden links. Also, for multiple-step ahead forecasting and infrequent items, the performance of the baseline method reduces significantly.

Figure 3 shows the results for the DBLP dataset. Similar to the POS dataset, our proposed method achieves the best performance in all link prediction and hidden link prediction in the DBLP dataset. The proposed method with LSTM forecasting achieves the highest performance. Note that for this dataset the proposed method with VAR forecasting performs better than the proposed method with Holt-Winters forecasting. VAR forecasting method performs well for stationary data, and the DBLP dataset does not have strong seasonality and is fairly stationary.

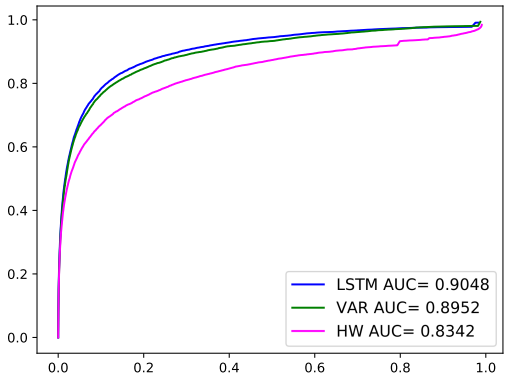
In order to evaluate the effect of NMF in extracting latent features from time series graphs, we carried out experiments using only the data forecasting methods; LSTM, Holt-Winters and VAR, without the NMF method. Figure 2 (b) and Fig. 3 (b) show the results for the forecasting methods without NMF for the POS and DBLP datasets respectively. The methods did not outperform the proposed method. In addition, the forecasting methods only predict values for existing links and cannot predict hidden links at all. This is because they make predictions by focusing only on one feature at each time period, and cannot capture hidden latent features, hence they cannot predict hidden links. This shows that the proposed method with NMF works well in extracting hidden latent features by analyzing the underlying graph structure. By extracting latent features from historical graphs NMF is able to capture nodes with common features, even if those nodes have not been linked in the past, and hence predict hidden links from past graph structure.

4.2.2 Effect of Ensemble Learning

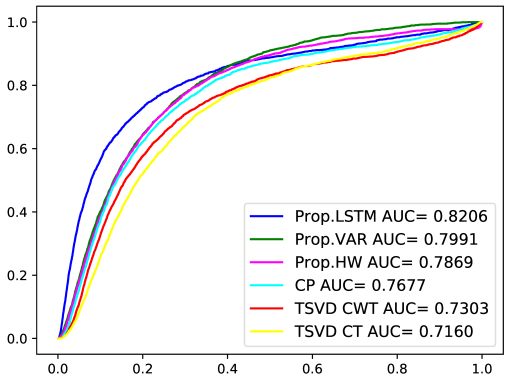
In Section 3.1 we described an extension to our proposed method by applying ensemble learning. Here we evaluate the effect of ensemble learning to the proposed method of NMF with Holt-Winters and VAR forecasting methods as shown by Eqs. (17) and (18) described in Section 3.1. We use prop.VAR w/ensemble, prop.HW w/ensemble to denote the proposed method of NMF with VAR and Holt-Winters forecasting methods with ensemble learning, respectively. Prop.VAR w/o ensemble and prop.HW w/o ensemble denotes the proposed method of NMF with VAR and Holt-Winters forecasting methods without ensemble learning, respectively.



(a) All link prediction



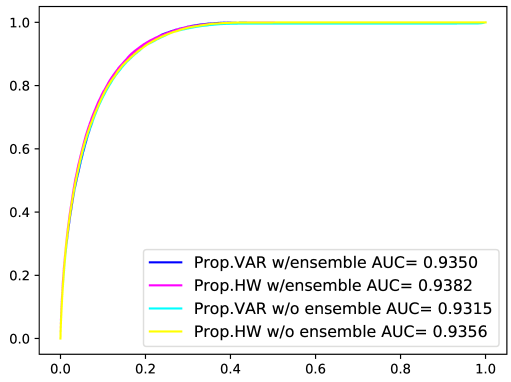
(b) All link prediction
Results for forecasting methods without NMF



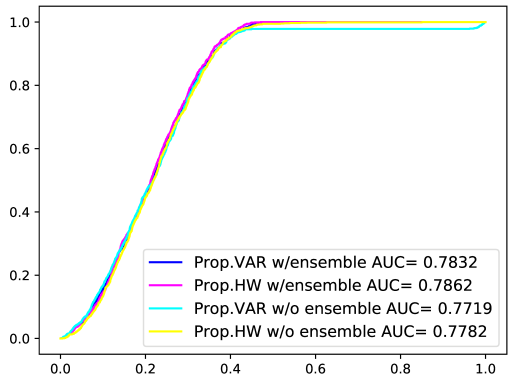
(c) Hidden link prediction

Fig. 3 ROC curves for DBLP dataset comparison with existing methods.

Figure 4 shows the results for method of ensemble learning for the POS dataset. We observe that ensemble learning method led to an increase in the prediction accuracy of the proposed methods with Holt-Winters and VAR forecasting, especially in hidden link prediction. The AUC values for all link prediction increased by 0.38% and the AUC values for hidden link prediction increased by 1.46% for the proposed method with VAR forecasting. On the other hand, the AUC values for all link prediction increased by

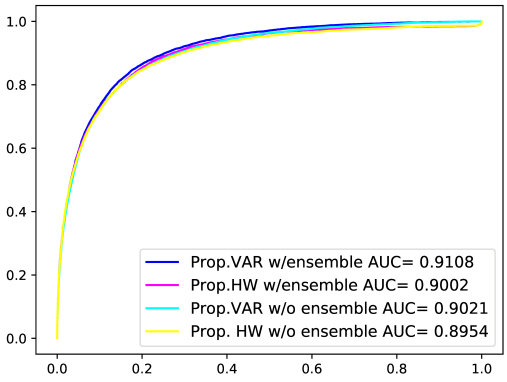


(a) All link prediction

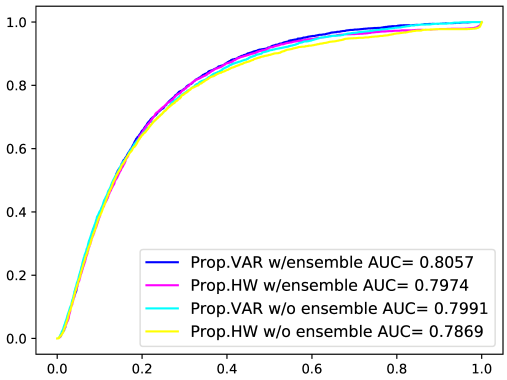


(b) Hidden link prediction

Fig. 4 ROC curves for POS dataset method of ensemble learning.



(a) All link prediction



(b) Hidden link prediction

Fig. 5 ROC curves for DBLP dataset method of ensemble learning.

0.35% and the AUC values for hidden link prediction increased by 1.02% for the proposed method with Holt-Winters forecasting.

Figure 5 shows the results for the DBLP dataset. Ensemble learning method led to an increase in the prediction accuracy of the proposed methods with Holt-Winters and VAR forecasting. The AUC values for all link prediction increased by 0.96% and the AUC values for hidden link prediction increased by 0.83% for the proposed method with VAR forecasting. On the other hand, the AUC values for all link prediction increased by 0.53% and the AUC values for hidden link prediction increased by 1.33% for the proposed method with Holt-Winters forecasting.

Proposed method with LSTM forecasting achieved the best performance for all our datasets as shown in Fig. 2 and Fig. 3. The proposed method with Holt-Winters forecasting performed better than proposed method with VAR forecasting for the POS dataset. And, the proposed method with VAR forecasting performed better than proposed method with Holt-Winters forecasting method for the DBLP dataset. Previous research shows that the performance of the time series forecasting methods vary greatly across different datasets [9], [15]. Therefore, it is important to analyze the characteristics of the dataset and choose the method which achieves the best performance.

4.2.3 Analysis of Features Extracted by NMF

In this section we show examples of the latent features extracted by NMF method. **Figure 6** and **Fig. 7** contain examples of features extracted for three consecutive time steps from our datasets. Figure 6 shows the values of the customers features ($\boldsymbol{U}_k$) and items features ($\boldsymbol{V}_k$) for $k = 50$ for three consecutive time steps, $t = 1, \cdots, 3$ for the POS dataset. Figure 7 shows the values of the author features ($\boldsymbol{U}_k$) and conference features ($\boldsymbol{V}_k$) for $k = 30$ for three consecutive time steps, $t = 3, \cdots, 5$ for the DBLP dataset. Top conferences are Interspeech, ICIP, IJCAI, and ICASSP. Some of the top authors are Kang G. Shin, Randy H. Katz, Geoffrey C. Fox, and others listed in the caption.

5. Related Work

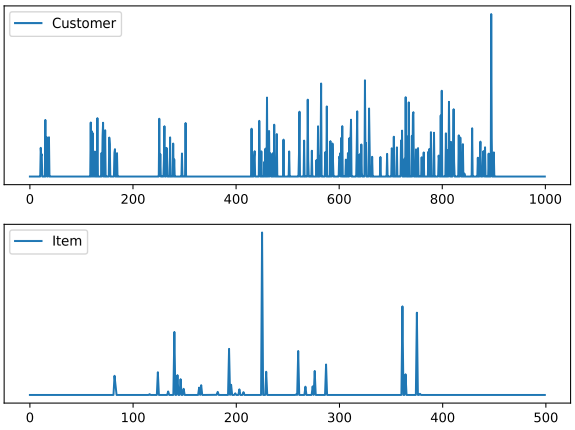
In this section, we describe existing methods for the link prediction problem using matrix decomposition techniques.

Truncated singular value decomposition (TSVD) is a low-rank matrix approximation technique which can used for time series link prediction [6]. TSVD decomposes a matrix $\boldsymbol{X}$ of size $M \times N$ into three matrices. The best K rank approximation of the original matrix is given by

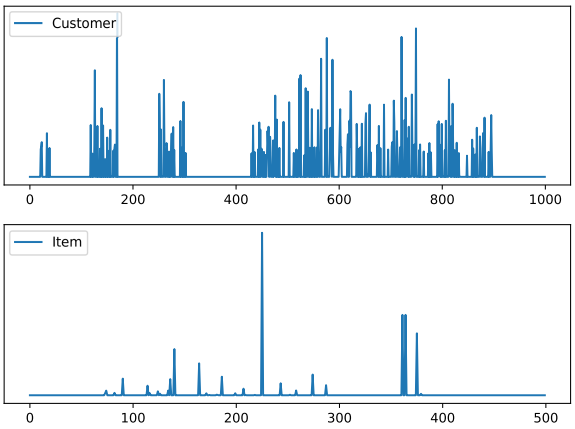
$$\boldsymbol{X} \approx \boldsymbol{U}_K \Sigma_K \boldsymbol{V}_K, \tag{22}$$

where $\boldsymbol{U}_K$ and $\boldsymbol{V}_K$ are orthogonal matrices of size $M \times K$ and $K \times N$ respectively, and Σ_K is a $K \times K$ diagonal matrix.

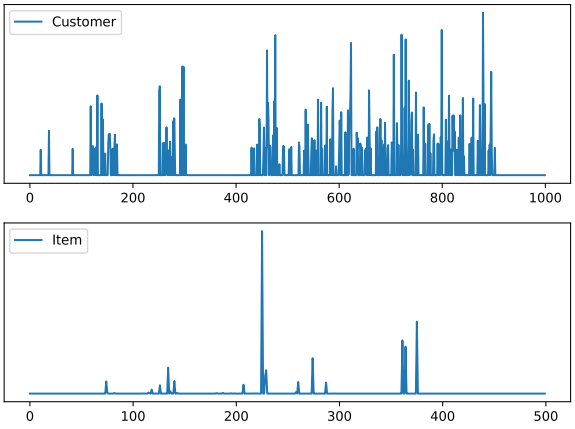
A three dimensional tensor can be reduced to a two dimensional matrix using the collapsed tensor (CT) and collapsed weighted tensor (CWT) techniques proposed in Ref. [7]. CT



(a) Factors from feature 50 at time $t = 1$



(b) Factors from feature 50 at time $t = 2$



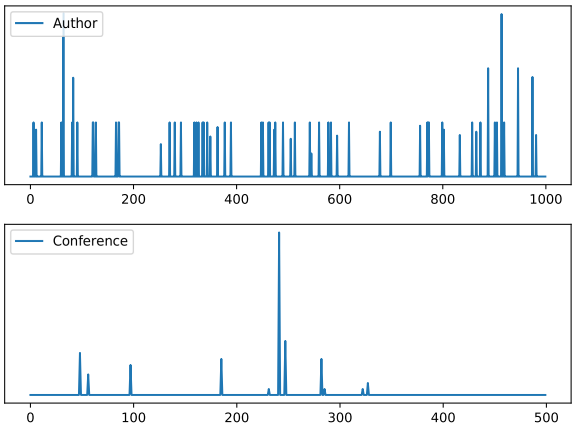
(c) Factors from feature 50 at time $t = 3$

Fig. 6 Examples of features from NMF model for three consecutive months for POS data.

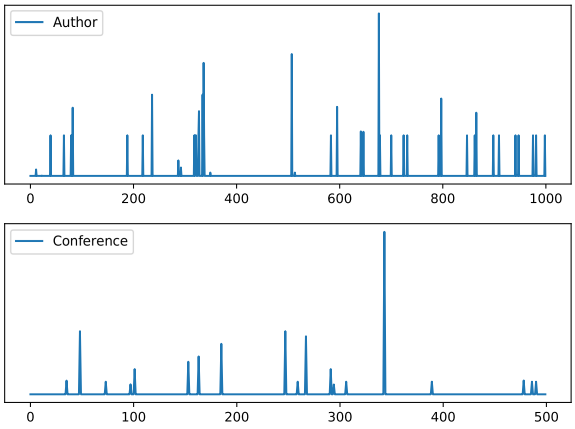
method removes the time series information of a three dimensional tensor by taking the sum in the time direction as shown below.

$$\boldsymbol{X} = \sum_{t=1}^{T} \boldsymbol{Z}_t, \tag{23}$$

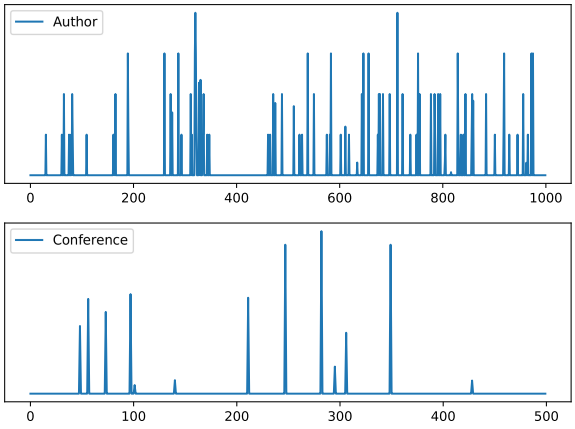
where $\boldsymbol{Z}_t$ is the matrix at time t and $\boldsymbol{X}$ is the sum of all matrices from time 1 to time T. CWT method on the other hand assigns temporal weights to the elements and reduces a three-dimensional matrix to a two-dimensional matrix while maintaining the time



(a) Factors from feature 30 at time $t = 3$: Top authors are Randy H. Katz, Geoffrey C. Fox and Hans-Peter Kriegel. Top conferences are Interspeech, IJCAI and ISCAS.



(b) Factors from feature 30 at time $t = 4$: Top authors are Stephan Olari and Philip S. Y and Akinori Yonezawa. Top conferences are Interspeech, ICIP and ICASSP.



(c) Factors from feature 30 at time $t = 5$: Top authors are Kang G. Shin, Guang R. G and Mike P. Papazogl. Top conferences are Interspeech, ICASSP and ICIP.

Fig. 7 Examples of features from NMF model for three consecutive years for DBLP data.

series information of the three-dimensional tensor as shown below.

$$\boldsymbol{X} = \sum_{t=1}^{T} (1 - \Theta)^{T-t} \boldsymbol{Z}_t, \tag{24}$$

where $\boldsymbol{Z}_t$ is the matrix at time t, $\boldsymbol{X}$ is the final matrix after summing all the matrices from time 1 to time T, and $\Theta \in (0, 1)$ is a parameter that is chosen by the user depending on the experiments on the training data. As shown by Eq. (24), CWT assigns more weight to the most recent links. TSVD matrix decomposition method can be used with the matrices resulting from CT and CWT, in this paper we refer to them as TSVD CT and TSVD CWT, respectively.

In addition, Canonical Polyadic (CP) decomposition is a common tensor matrix decomposition method which decomposes a three dimensional matrix to three rank-one tensors, just like SVD decomposes a matrix to three rank-one matrices [7], [12]. However unlike SVD, the resultant tensors from CP decomposition are not orthogonal, but they are unique and hence they can be used for forecasting. The CP decomposition of a tensor $\boldsymbol{X}$ of size $M \times N \times T$ is defined as follows:

$$\boldsymbol{X} \approx \sum_{k=1}^{K} \lambda_k a_k \circ b_k \circ c_k, \tag{25}$$

where K is the number of components and the symbol $\circ$ represents the outer product[*1]. a_k and b_k extract the column and row features respectively, while c_k extracts the temporal components. The size of a_k, b_k and c_k is $M \times K$, $K \times N$ and $K \times T$ respectively. The three matrices correspond to the axes of the original three-dimensional tensor.

Non-negative matrix factorization (NMF) is commonly used in extracting latent features from non-negative data [16], [23]. Given a non-negative matrix $\boldsymbol{X}$ of size $M \times N$, NMF decomposes the matrix into two matrices, $\boldsymbol{U}$ and $\boldsymbol{V}$ of low dimensions such that they do not include negative values, such that $\boldsymbol{X} \approx \boldsymbol{UV}$. The size of $\boldsymbol{U}$ and $\boldsymbol{V}$ is $M \times K$ and $K \times N$ respectively. Many extensions of NMF have been proposed [2], [10], [24], [27]. Reference [2] proposed an online NMF to deal with continuously incoming data, where rows are added at each time step. They added a new regularization constraint to the original NMF optimization problem so as to consider previously extracted features. Suppose a row vector $\boldsymbol{X^{(t)}}$ is observed at each time step t, online NMF decomposes the concatenation of $\boldsymbol{X^{(t)}}$ and $\boldsymbol{V^{(t-1)}}$ to obtain $\boldsymbol{U^{(t)}}$ and $\boldsymbol{V^{(t)}}$ for the current time step. Reference [10] proposed a streaming algorithm of NMF for temporally evolving data matrices. Similar to online NMF, streaming NMF accommodates incremental updates of NMF parameters. Streaming NMF captures emerging features and simultaneously eliminates the irrelevant features. Reference [24] proposed a dynamic NMF approach for capturing trending topics (features) from social media data. They considered the situation where at each time step a new matrix is observed and defined a new objective function with temporal regularization and squared error. They used doubly-regularized NMF to allow the rank of the NMF model to change over time as new topics emerge, and at the same time maintain consistency of the established trends. Reference [27] proposed a collective time-based matrix factorization technique to detect and track how input changes over time. They introduced a mapping factor $\boldsymbol{M^{(t)}}$ to capture the temporal dependencies, that is, how much the data at the current time step t is related to the data at the previous time step $t-1$. The approximation of the matrix for current time step is given by

$$\boldsymbol{X} \approx \boldsymbol{U^{(t)}} \boldsymbol{M^{(t)}} \boldsymbol{V^{(t-1)}}, \tag{26}$$

where $\boldsymbol{M^{(t)}} \geq 0$.

[*1] Three-way outer product is defined as: $X = a \circ b \circ c$.

6. Conclusion

In this paper we addressed the time series link prediction problem. We proposed a method of extracting latent features from time series data and modelling the features by combining NMF and time series data forecasting methods. We also proposed an extension to the proposed method, through applying ensemble learning to improve the prediction accuracy. We compared the performance of our methods with existing link prediction methods in predicting existing links and hidden links. As a result of the experiments with real datasets, we confirmed that the proposed methods perform well, especially in predicting hidden links.

There are several directions which can be considered for future work. First, our proposed method does not use attribute information, therefore attribute-based prediction is important to further improve the accuracy of our methods. Second, more complex deep learning methods such as bidirectional recurrent neural networks [25] will be investigated.

References

[1] Anselin, L. and Bera, A.K.: Introduction to spatial econometrics, *Handbook of Applied Economic Statistics*, pp.237–290 (1998).

[2] Cao, B., Shen, D., Sun, J.-T., Wang, X., Yang, Q. and Chen, Z.: Detect and track latent factors with online nonnegative matrix factorization, *IJCAI*, Vol.7, pp.2689–2694 (2007).

[3] Chatfield, C. and Yar, M.: Holt-winters forecasting: Some practical issues, *The Statistician*, pp.129–140 (1988).

[4] Clemen, R.T., Combining forecasts: A review and annotated bibliography, *International Journal of Forecasting*, Vol.5, No.4, pp.559–583 (1989).

[5] Dietterich, T.G., Ensemble methods in machine learning, *International Workshop on Multiple Classifier Systems*, pp.1–15, Springer (2000).

[6] Dumais, S.T., Furnas, G.W., Landauer, T.K., Deerwester, S. and Harshman, R.: Using latent semantic analysis to improve access to textual information, *SIGCHI*, pp.281–285 (1988).

[7] Dunlavy, D.M., Kolda, T.G. and Acar, E.: Temporal link prediction using matrix and tensor factorizations, *ACM Trans. Knowledge Discovery from Data (TKDD)*, Vol.5, No.2, p.10 (2011).

[8] Getoor, L. and Diehl, C.P.: Link mining: A survey, *ACM SIGKDD Explorations Newsletter*, Vol.7, No.2, pp.3–12 (2005).

[9] Goel, H., Melnyk, I., Oza, N., Matthews, B. and Banerjee, A.: Multivariate aviation time series modeling: Vars vs. lstms (2016).

[10] Hayashi, K., Maehara, T., Toyoda, M. and Kawarabayashi, K.: Real-time top-r topic detection on twitter with topic hijack filtering, *Proc. 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.417–426, ACM (2015).

[11] Kalekar, P.S., Time series forecasting using holt-winters exponential smoothing, *Kanwal Rekhi School of Information Technology*, 4329008:1–13 (2004).

[12] Kolda, T.G. and Bader, B.W.: Tensor decompositions and applications, *SIAM Review*, Vol.51, No.3, pp.455–500 (2009).

[13] Koren, Y.: Collaborative filtering with temporal dynamics, *SIGKDD*, pp.447–456, ACM (2009).

[14] Koren, Y., Bell, R. and Volinsky, C.: Matrix factorization techniques for recommender systems, *Computer*, Vol.8, pp.30–37 (2009).

[15] Lai, G., Chang, W.-C., Yang, Y. and Liu, H.: Modeling long-and short-term temporal patterns with deep neural networks, *ACM SIGIR*, pp.95–104 (2018).

[16] Lee, D.D. and Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization, *Nature*, Vol.401, pp.788–791 (1999).

[17] Lee, D.D. and Seung, H.S.: Algorithms for non-negative matrix factorization, *NIPS*, pp.556–562, MIT Press (2001).

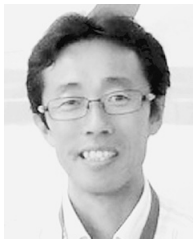
[18] Liben-Nowell, D. and Kleinberg, J.: The link-prediction problem for social networks, *Journal of the American society for Information Science and Technology*, Vol.58, No.7, pp.1019–1031 (2007).
[19] Linden, G., Smith, B. and York, J.: Amazon.com recommendations: Item-to-item collaborative filtering, *IEEE Internet Computing*, Vol.1, pp.76–80 (2003).
[20] Luo, X., Zhou, M., Shang, M., Li, S. and Xia, Y.: A novel approach to extracting non-negative latent factors from non-negative big sparse matrices, *IEEE Access*, Vol.4, No.1 (2016).
[21] Lütkepohl, H.: *New Introduction to Multiple Time Series Analysis*, Springer Science & Business Media (2005).
[22] Makridakis, S. and Hibon, M.: The M3-competition: Results, conclusions and implications, *International Journal of Forecasting*, Vol.16, No.4, pp.451–476 (2000).
[23] Paatero, P. and Tapper, U.: Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, *Environmetrics*, Vol.5, No.2, pp.111–126 (1994).
[24] Saha, A. and Sindhwani, V.: Learning evolving and emerging topics in social media: A dynamic nmf approach with temporal regularization, *Proc. 5th ACM International Conference on Web Search and Data Mining*, pp.693–702, ACM (2012).
[25] Schuster, M. and Paliwal, K.K.: Bidirectional recurrent neural networks, *IEEE Trans. Signal Processing*, Vol.45, No.11, pp.2673–2681 (1997).
[26] Takeuchi, K, Kashima, H. and Ueda, N.: Autoregressive tensor factorization for spatio-temporal predictions, *ICDM*, pp.1105–1110 (2017).
[27] Vaca, C.K., Mantrach, A., Jaimes, A. and Saerens, M.: A time-based collective factorization for topic discovery and monitoring in news, *Proc. 23rd International Conference on World Wide Web*, pp.527–538, ACM (2014).
[28] Yu, H.-F., Rao, N. and Dhillon, I.S.: Temporal regularized matrix factorization for high-dimensional time series prediction, *Advances in Neural Information Processing Systems*, pp.847–855 (2016).

Faith Mutinda received her B.E. degree from Kenyatta University, Kenya, in 2015. She is currently a Master student in Nara Institute of Science and Technology, Japan. Her research interests include recommender systems and time series data analysis.

Atsuhiro Nakashima received his B.E. and M.E. degrees from Osaka University, Japan, in 2016 and 2018 respectively. His research interests include link prediction, recommender systems, and matrix factorization.

Koh Takeuchi received his B.E. and M.E. degrees from Waseda University in 2009 and 2011, and his Ph.D. degree in Informatics from Kyoto University in 2019. In 2011, he joined NTT Communication Science Laboratories, Japan. He is currently a research scientist at Ueda Research Laboratory of NTT Communication Science Laboratories, Kyoto, Japan. His research interests include machine learning, data mining, and spatio-temporal data analysis.

Yuya Sasaki received his B.E., M.E., and Ph.D. degrees from Osaka University, Japan, in 2009, 2011, and 2014, respectively. He is currently an Assistant Professor in Graduate School of Information Science and Technology, Osaka University. His research interests include database systems, graph data processing, and urban computing.



Makoto Onizuka is a Professor at Graduate School of Information Science and Technology, Osaka University. He developed LiteObject (object-relational main memory database system), XMLToolkit (XML stream engine and unix-like XML data processing tools), CBoC type2 (Common IT Bases over Cloud Computing at NTT). His current research focuses on cloud-scale data management and Big data mining.