Time Series Link Prediction using NMF

Faith W. Mutinda*, Atsuhiro Nakashima*, Koh Takeuchi†, Yuya Sasaki*, Makoto Onizuka*

*Graduate School of Information Science and Technology, Osaka University, Osaka, Japan

†NTT Communication Science Laboratories

*{wavinya.faith, nakashima.atsuhiro, sasaki, onizuka}@ist.osaka-u.ac.jp, † koh.t@acm.org

***Abstract*—Data in many fields such as e-commerce, social networks, and web data can be modeled as 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 the 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 Holt-Winters method. NMF extracts the latent features while the Holt-Winters method captures the changes of features with time. Our method can predict hidden links that do not appear in historical graphs. Our experiments with real dataset show that our method has a higher prediction accuracy compared to existing methods.**

***Index Terms*—Link prediction, Holt-Winters forecasting, Non-negative matrix factorization (NMF)**

I. 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 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 the customer and the item. Graphs can be used in various data mining tasks, such as detecting hidden groups, detecting missing links, and ranking objects [9].

Real world data such as product sales is dynamic because their relationships change with time. Therefore, it is useful to predict how the links in the graphs change with 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 [8], [12], [13], [17], [18]. We formally define our problem as follows.

Problem definition: **(Time Series Link Prediction)**. Given the 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 [8], [12] 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 [19]. In addition, Holt-Winters method is commonly used in time series data forecasting because it effectively captures periodicity or seasonal fluctuations in the data [3], [10].

In this paper, we propose a new method that combines NMF and Holt-Winters method 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 Holt-Winters to predict the latent features at time $T + 1$ from the time series of the decomposed matrices from 1 to T. In addition, we propose two extensions to our method so as to improve the preciseness of the prediction; matrix decomposition by leveraging time invariant features and future prediction by ensemble learning. As for the former extension, we assume that given a bipartite graph with two sets of nodes, the latent features in one set are time invariant, while the latent features in the other set change with time. Then, we apply NMF and Holt-Winters method to an adjacency matrix of the bipartite graph, where the column or row features are held constant. As for the latter, ensemble learning improves the accuracy of predictions [6]. 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. We evaluate our method by comparing it with other methods using a real dataset and our method showed an improvement in the prediction accuracy.

Further, Vector Auto Regressive (VAR) model has been successfully used in multivariate time series data forecasting [21], [22]. We compare the performance of Holt-Winters and VAR forecasting methods, and the results show that Holt-Winters results are more accurate and therefore it outperforms VAR on our dataset. In spite of the popularity of VAR models, the results indicate that in some domains VAR does not perform well.

The remainder of this paper is organized as follows. Section

II describes related work. Section III overviews the preliminaries of this work. Section IV describes the details of our proposed method. Section V reviews the results of our experiments. Section VI provides our brief conclusion.

II. Related work

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

A. Matrix decomposition approach

Truncated singular value decomposition (TSVD) is a low-rank matrix approximation technique which can used for time series link prediction [7]. 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

$$\boldsymbol{X} \approx \boldsymbol{U}_K \Sigma_K \boldsymbol{V}_K^T, \tag{1}$$

where $\boldsymbol{U}_K$ and $\boldsymbol{V}_K$ are orthogonal matrices of size $M \times K$ and $N \times K$ 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 [8]. CT method removes the time series information of a three dimensional tensor by taking the sum in the time direction as shown below.

$$\boldsymbol{X} = \sum_{t=1}^{T} \boldsymbol{Z}_t, \tag{2}$$

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 assigns temporal weights to the elements and reduces a three-dimensional matrix to a two-dimensional matrix while maintaining the time series information of the three-dimensional tensor as shown below.

$$\boldsymbol{X} = \sum_{t=1}^{T} (1-\Theta)^{T-t} \boldsymbol{Z}_t, \tag{3}$$

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 (3), CWT assigns more weight to the most recent links. TSVD matrix decomposition method can be used with the matrices resulting from CT and CWT, 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 [8], [11]. 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:

$$\boldsymbol{X} \approx \sum_{k=1}^{K} \lambda_k a_k \circ b_k \circ c_k, \tag{4}$$

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 three matrices correspond to the axes of the original three-dimensional tensor.

B. Forecasting approach

Various time series data forecasting methods have been studied [20]. 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 more weight is given to the most recent observation.

$$P_{t+1} = \alpha y_t + (1-\alpha) P_t, \tag{5}$$

where α is a learning coefficient which takes values $0 \leq \alpha \leq 1$ 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

$$P_{t+1} = \alpha y_t + (1-\alpha) P_t + b_t \text{ and} \tag{6}$$

$$b_t = \beta(P_t - P_{t-1}) + (1-\beta) b_{t-1}, \tag{7}$$

where β is the trend coefficient.

Auto Regressive (AR) models are also widely used in data forecasting [22]. The Vector Auto Regressive (VAR) model is used in prediction of future observations in multivariate time series data, and has the parameters; time lag set L, weight A_l and Gaussian noise ϵ_t.

$$y_t = \sum_{l \in L} A_l y_{t-l} + \epsilon_t. \tag{8}$$

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 [2]. VAR has also been used in space-time prediction problems of sensor data by combining AR model with tensor decomposition [21].

Recently, Recurrent Neural Networks especially Long-Short Term Memory (LSTM) have been studied for time series data forecasting [14]. 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.

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

III. PRELIMINARIES

As preliminaries of our method, we explain time series graph, NMF, and Holt-Winters method.

A. 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. We consider the graph structure over a period of time. The graph structure changes with time (e.g., sales data in every month). 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.*

B. 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 [15]. 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). In addition, 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. In our experiments, we use a supermarket dataset and, since sales events are expected to follow the Poisson distribution, we use the Kullback- Leibler (KL) divergence. The update rules based on KL divergence are defined by the equations below as discussed by [16].

$$u_{ik} \leftarrow u_{ik} \frac{\frac{\sum_j x_{ij} v_{kj}}{\sum_k u_{ik} v_{kj}}}{\sum_j v_{kj}}, \tag{9}$$

$$v_{kj} \leftarrow v_{kj} \frac{\frac{\sum_i x_{ik} u_{ik}}{\sum_k u_{ik} v_{kj}}}{\sum_i u_{ik}}. \tag{10}$$

This update calculation is repeated through several iterations. 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.

C. Holt-Winters method

Holt-Winters is a forecasting method suitable for time series data that has seasonality and trend [3], [10]. Seasonality means that the time series data has trends that repeat every m cycles. There are two variations of the Holt-Winters technique; additive and multiplicative methods that 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 dataset is fairly constant. The additive Holt-Winters method consists of three smoothing equations and a forecasting equation as shown below.

$$l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}). \tag{11}$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}. \tag{12}$$

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m}. \tag{13}$$

$$y_{t+h} = l_t + hb_t + s_{t-m+h}. \tag{14}$$

where $y_1, y_2, \cdots, y_t$ 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 equations (11-13) minimize the squared error and the forecast, (y_{t+h}) at h time periods ahead is calculated as shown in (14).

IV. PROPOSED METHOD

In our proposed method, we use NMF to extract the underlying latent features and Holt-Winters method to capture the periodic change of the extracted latent features and predict future actions.

A. Outline of proposed method

Fig. 1 shows the 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 $\boldsymbol{U}^{(1)}, \boldsymbol{V}^{(1)}, \cdots, \boldsymbol{U}^{(T)}, \boldsymbol{V}^{(T)}$. Holt-Winters 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 calculated as $\boldsymbol{X}^{(T+1)} = \boldsymbol{U}^{(T+1)}\boldsymbol{V}^{(T+1)}$.

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 $\boldsymbol{U}^{(t)}$ $(\boldsymbol{V}^{(t)})$ $(1 \leq t \leq T)$. To solve this problem, we first apply NMF to the average

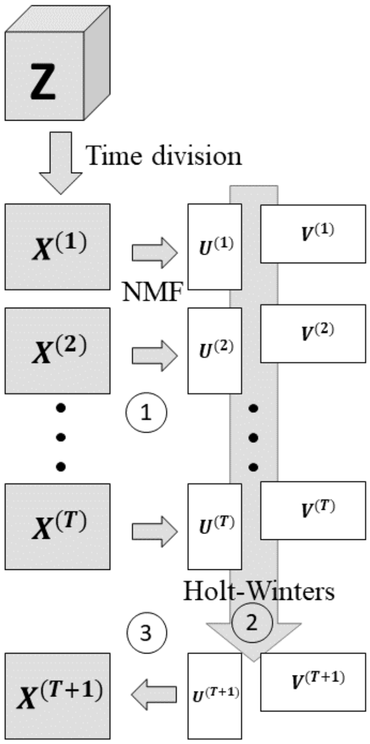

Fig. 1. Overview of proposed method: Step 1: NMF is applied to the matrices $\boldsymbol{X}^{(1)}, \cdots, \boldsymbol{X}^{(T)}$. Step 2: Holt-Winters method is applied to the sequence of $\boldsymbol{U}^{(t)}$ and $\boldsymbol{V}^{(t)}$ $(1 \leq t \leq T)$. Step 3: The predicted matrix at time $T+1$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

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 does not change, 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 outline 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 Holt-Winters is applied to forecast the 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).

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

Input $\boldsymbol{Z} = (\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, ..., \boldsymbol{X}^{(T)}), \boldsymbol{X}_{Ave}$
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: **Holt − Winters** : $\boldsymbol{U}^{(T+1)} \leftarrow \boldsymbol{U}_{list}$
8: **Holt − Winters** : $\boldsymbol{V}^{(T+1)} \leftarrow \boldsymbol{V}_{list}$
9: $\boldsymbol{X}^{(T+1)} = \boldsymbol{U}^{(T+1)}\boldsymbol{V}^{(T+1)}$

B. Matrix decomposition by time invariant features

In time series link prediction for bipartite graphs with two sets of nodes, we can assume that the latent features in one set are time invariant while the latent features in the other set change with time. For example, since we use supermarket dataset for this paper, this dataset contains two sets of nodes which represent customers and items. Then, we can assume that customers features (preferences) remain constant but products features change with time. This is because some items such as fruits are seasonal and special products are promoted during special events, and hence they have a greater effect on sales. Therefore, it is important to consider a model where temporal change is represented in only one of the nodes of the bipartite graphs.

Given the adjacency matrices of the bipartite graphs from time 1 to time T, we assume that the row features are time invariant but the column features change with time, and vice-versa. In the case of time invariant column features, NMF is applied to a matrix of size $TM \times N$ obtained by concatenating T matrices of size $M \times N$. The matrix is decomposed into two matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ of sizes $TM \times K$ and $K \times N$ respectively. The column features (matrix $\boldsymbol{V}$) will have the same values over the entire time, and change is represented only in the row features (matrix $\boldsymbol{U}$). Matrix $\boldsymbol{U}$ of size $TM \times K$ is then divided into T matrices, $\boldsymbol{U}^1, \boldsymbol{U}^2, \cdots, \boldsymbol{U}^T$ and Holt-Winters method is used to forecast the values of $\boldsymbol{U}^{T+1}$ at time $T+1$. The predicted matrix at time $T+1$ is calculated as $\boldsymbol{X}^{(T+1)} = \boldsymbol{U}^{(T+1)}\boldsymbol{V}$. Fig. 2 shows the proposed method with time invariant column features.

Algorithm 2 shows the algorithm for the proposed method with time invariant column features. NMF is applied to matrix $\boldsymbol{X}$ of size $TM \times N$, which is then decomposed to matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ of sizes $TM \times K$ and $K \times N$ respectively (line 1). The $TM \times K$ matrix $\boldsymbol{U}$ is then divided into T matrices to create a list of decomposition matrices. Holt-Winters is applied to the list of matrices to predict the values of $\boldsymbol{U}^{T+1}$ at time $T+1$, (line 4). The predicted matrix at time $T+1$ is calculated as $\boldsymbol{X}^{T+1} = \boldsymbol{U}^{T+1}\boldsymbol{V}$ (line 5).

Assuming that the temporal change in one set of the nodes of a bipartite graph is significantly larger than that in the other node, this algorithm improves modelling of the latent features and prediction accuracy.

C. Holt-Winters and NMF methods 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

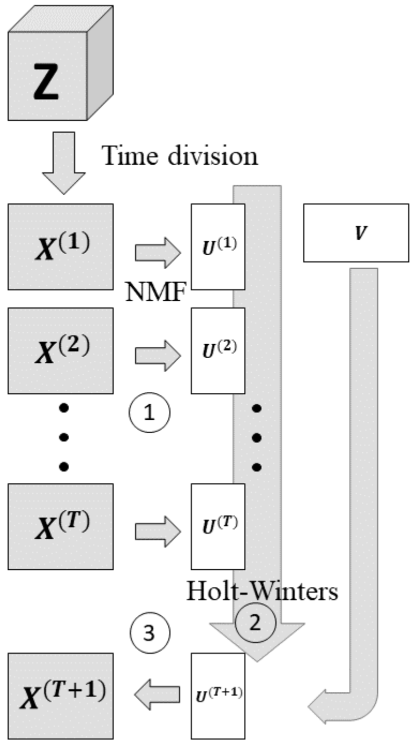

Fig. 2. Proposed method by leveraging time invariant features $\boldsymbol{V}$

Algorithm 2 Calculation of prediction matrix $\boldsymbol{X}^{(T+1)}$ by leveraging the time invariant column features

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

$\mathbf{NMF} \quad : \quad \boldsymbol{U}, \boldsymbol{V} \leftarrow \boldsymbol{X}$
$\textbf{Time Divide} \quad : \quad \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, ..., \boldsymbol{U}^{(T)} \leftarrow \boldsymbol{U}$
$\boldsymbol{U}_{list} \leftarrow \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, ..., \boldsymbol{U}^{(T)}$
$\mathbf{Holt-Winters} \quad : \quad \boldsymbol{U}^{(T+1)} \leftarrow \boldsymbol{U}_{list}$
$\boldsymbol{X}^{(T+1)} = \boldsymbol{U}^{(T+1)} \boldsymbol{V}$

combining multiple individual models improves prediction accuracy compared to using a single model [6]. 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,

$$\boldsymbol{X} = \sum_{K \in (5,10,\ldots,100)} \sum_{m \in (1,2,\ldots,12)} \frac{\boldsymbol{X}_{Km}}{\|\boldsymbol{X}_{Km}\|_F}, \tag{15}$$

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

V. Experiments

In this section, we describe experiments and their results by comparing our method proposed in Section IV with the former methods summarized in Section II. We also investigate the results of our variation of 1) fixing time invariant features and 2) ensemble learning described in Section IV.

In our experiments we use a point-of-sales dataset for a period of 24 months obtained from a supermarket. 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. The dataset contains 25668 customers and 113688 items. From this dataset we extract the top 1000 frequent customers and top 500 items with highest number of sales. We transform the data to adjacency matrices of bipartite graphs, and use the data for 23 months as training dataset and the data for the last month as the test dataset. We generate a three-dimensional tensor $\boldsymbol{Z}$ extended in the time direction, that is, if customer i buys n items in month t, then $\boldsymbol{Z}(i,j,t) = n$. We eliminate the influence of large values in the data by normalizing the data according to the equation below which was proposed by [8].

$$\boldsymbol{Z}(i,j,t) = \begin{cases} 1 + log(n) & n > 0 \\ 0 & n = 0. \end{cases} \tag{16}$$

The link prediction problem can be regarded as a matrix completion problem with implicit feedback [1]. Given a $M \times N$ adjacency matrix, we assume that the matrix is binary where the presence of links is represented by 1s and absence of links is represented by 0s. Since the objective of our study is to predict whether there is a link or not between the customers and items, if customer i purchased item j n times, the link information is represented as,

$$\boldsymbol{Y}(i,j) = \begin{cases} 1 & n > 0 \\ 0 & n = 0. \end{cases} \tag{17}$$

In addition, 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. We set the seasonality component to $m = 3$ months, by assuming seasonality appears every quarter year in sales data. Moreover, Holt-Winters method can also be used with the temporal information extracted by CP (described in Section II) to predict future observations, we refer to this as CP+HW. We compare the performance of our proposed methods with the methods described in Section II; CP+HW, CP and TSVD CWT. For TSVD, CP and NMF decomposition, instead of using a fixed value of K, we use an ensemble approach as shown below.

$$\boldsymbol{X} = \sum_{K \in (5,10,\ldots,100)} \frac{\boldsymbol{X}_K}{\|\boldsymbol{X}_K\|_F}, \tag{18}$$

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 $\boldsymbol{X}_K$.

In addition, we select a threshold value based on the Youden's J statistic [2] and use this threshold to determine the presence or absence of a link [5]. That is, predicted values above the threshold are considered as positive links and predicted values below the threshold indicate that there is no link between the corresponding customer and item. Further, 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. We compare the performance of the methods in predicting all links and hidden links. In all links 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. Since our objective is to predict the significant entries, we treat all nonzero entries as ones (positive links) and the other entries as zeros (no links). The total number of links in our test dataset was 33233 and the number of hidden links was 1258.

A. Comparison with existing methods

We compare the performance of the methods in terms of their precision, recall, and F-measure. We use Prop. default, TN, FP, FN, TP, P.L, C.P.L, Prec., Rec. and F-meas. to denote proposed method, the number of true negatives, false positives, false negatives, true positives, predicted links, correctly predicted links, precision, recall, and F-measure, respectively. In addition, we use simple average as the baseline method.

Table I shows the performance in the prediction of all links. Holt-Winters method has the highest accuracy and F-measure. Excluding the Holt-Winters method, our method has the highest precision and F-measure. The average prediction performs well because we considered the most frequent users and items in our dataset and forecasts for only one time step ahead. However, for multiple-step ahead forecasting and infrequent items, the performance reduces significantly because average method cannot capture data patterns.

Table II shows the results of the methods in predicting hidden links. As expected, the F-measures for hidden links prediction are lower than the F-measures for all links prediction. The results show that the Holt-Winters and average prediction methods cannot predict hidden links at all and their F-measure is 0. Our proposed method achieved the highest accuracy and F-measure values. This result indicates that the proposed method is better in extracting hidden features by analyzing the underlying graph structure.

Fig. 3 shows the performance of the methods in terms of the area under the receiver operating characteristic curve (AUC) for all links prediction. ROC curve shows the trade-off between sensitivity and specificity, and therefore curves close to the top-left corner indicate a better performance [1]. The methods perform well because their ROC curves are close to the top-left corner. Holt-Winters method is the best when the false positive rate is low. As the false positive rate increases, the proposed method is the best and has the highest AUC value.

TABLE I
COMPARISON WITH EXISTING METHODS: ALL LINKS PREDICTION

	Prop. default	**Base-line**	**CP+ HW**	**CP**	**HW**	**TSVD CWT**
TN	393342	389048	376670	373733	409894	369646
FP	73425	77719	90097	93034	56873	97121
FN	3924	2764	4623	4587	5203	4887
TP	29309	30469	28610	28646	28030	28346
P.L	102734	108188	118707	121680	84903	125467
C.P.L	29309	30469	28610	28646	28030	28346
Prec.	28.53	28.16	24.10	23.54	**33.01**	22.59
Rec.	88.19	**91.68**	86.09	86.20	84.34	85.29
F-meas.	43.11	43.09	37.66	36.98	**47.45**	35.72

TABLE II
COMPARISON WITH EXISTING METHODS: HIDDEN LINKS PREDICTION

	Prop. default	**Baseline**	**CP+ HW**	**CP**	**HW**	**TSVD CWT**
P.L	25695	0	27343	26731	0	36275
C.P.L	689	0	589	559	0	667
Prec.	**2.68**	0	2.15	2.09	0	1.84
Rec.	**54.77**	0	46.82	44.44	0	53.02
F-meas.	**5.11**	0	4.12	3.99	0	3.55

B. Effectiveness of leveraging time invariant features

In Section IV we described the method of matrix decomposition by leveraging time invariant features to improve accuracy of the link prediction task. We use **Prop. column**

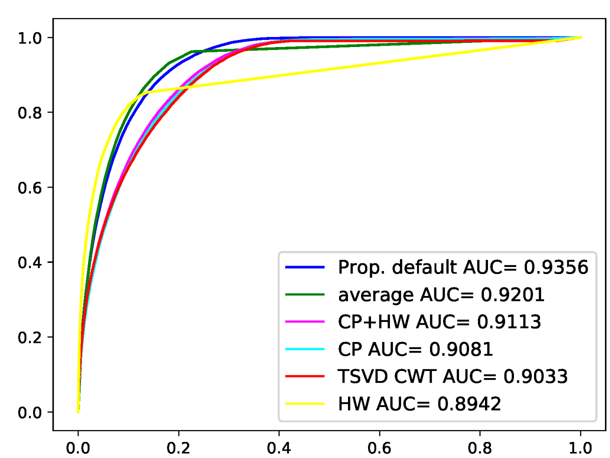

Fig. 3. ROC curve comparison with existing methods

[2] $J = \max(sensitivity + specificity - 1)$

TABLE III
COMPARISON WITH METHODS OF TIME INVARIANT FEATURES: ALL LINKS PREDICTION

	Prop. default	Prop. column	Prop. row
TN	393342	392518	389586
FP	73425	74249	75040
FN	3924	3705	4820
TP	29309	29528	30554
P.L	102734	103777	105594
C.P.L	29309	29528	30554
Prec.	28.53	28.45	**28.94**
Rec.	88.19	**88.85**	86.37
F-meas.	43.11	43.10	**43.35**

TABLE IV
COMPARISON WITH METHODS OF TIME INVARIANT FEATURES: HIDDEN LINKS PREDICTION

	Prop. default	Prop. column	Prop. row
P.L	25695	29120	23395
C.P.L	689	668	638
Precision	2.68	2.29	**2.73**
Rec.	**54.77**	53.10	50.72
F-meas.	5.11	4.40	**5.18**

and **Prop. row** to denote variants of our proposed method by leveraging the time invariant column features and the row features, respectively.

When predicting all links, table III shows that our method of leveraging time invariant row features achieved the best performance in terms of precision and F-measure. Table IV shows the performance in predicting hidden links. Our method of leveraging time invariant row features achieved the highest precision and F-measure. These results indicate that there is little change in row features with time and there are large temporal changes in column features. In our dataset, rows represents customers while columns represent items. The results show that the customers tastes remained constant while the product features changed with time. This can be explained by the fact that some products in supermarkets, especially food products such as vegetables change depending on seasons. However, only a single dataset was used for our experiments, and the performance of each method may vary depending on datasets. Therefore, by analyzing the characteristics of the dataset, it is possible to select the appropriate method and hence improve the accuracy.

Fig. 4 shows the ROC curves for our method of leveraging time invariant features in all links prediction. Although the ROC curve of the methods overlap largely, the AUC value of the method of matrix decomposition by time invariant row features is slightly higher than that of the proposed method and, therefore, it has the best performance in terms of AUC value.

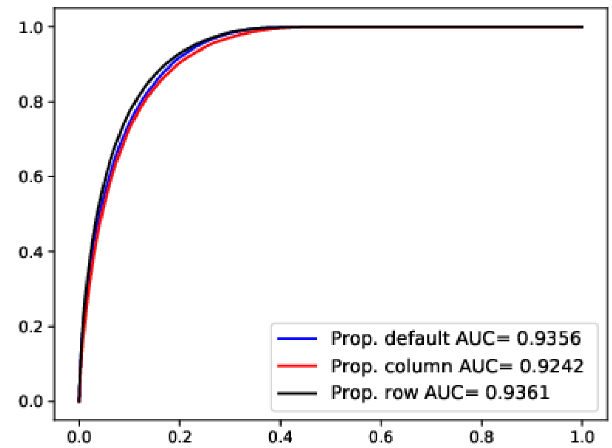

Fig. 4. ROC curves for our methods by leveraging time invariant features

TABLE V
COMPARISON WITH METHODS OF ENSEMBLE LEARNING: ALL LINKS PREDICTION

	Prop. default	Prop. column	Prop. row	Base-line	CP+ HW	HW
TN	394418	391455	394137	389048	385875	432022
FP	72349	75312	72630	77719	80892	34745
FN	3781	3215	3442	2764	4084	6832
TP	29452	30018	29791	30469	29149	26401
P.L	101801	105330	102421	108188	110041	61146
C.P.L	29452	30018	29791	30469	29149	26401
Prec.	28.93	28.50	29.09	28.16	26.49	**43.18**
Rec.	88.62	90.33	89.64	**91.68**	87.71	79.44
F-meas.	43.62	43.33	43.92	43.09	40.69	**55.95**

C. Comparison with method of ensemble learning

In section IV we described an extension to our proposed method by applying ensemble learning. Here we vary the Holt-Winters seasonality parameter, m and K for NMF as shown by equation (15) described in section IV. Table V shows that the Holt-Winters method has the highest accuracy and F-measure in all links prediction. Excluding Holt-Winters method, the method of leveraging time invariant row features has the highest accuracy. Comparing the results of table I, table III and table V, we observe that ensemble learning method led to a significant increase in the prediction accuracy of the proposed methods and Holt-Winters method. In hidden links prediction, table VI shows that the proposed method achieved the highest precision and F-measure. Although the F-measure is low, it increased significantly from 5.11 to 5.86. Since we considered a single dataset in our experiments, the performance of the proposed method and the method of matrix decomposition by time invariant features might vary. Therefore, depending on the dataset one can select the method with the best performance.

Fig. 5 shows the ROC curves for this method in all links

TABLE VI
COMPARISON WITH METHODS OF ENSEMBLE LEARNING: HIDDEN LINKS PREDICTION

	Prop. default	Prop. column	Prop. row	Base-line	CP+ HW	HW
P.L	19227	21897	21636	0	24210	0
C.P.L	600	555	636	0	545	0
Prec.	**3.12**	2.53	2.94	0	2.25	0
Rec.	47.69	44.12	**50.56**	0	43.32	0
F-meas.	**5.86**	4.79	5.56	0	4.28	0

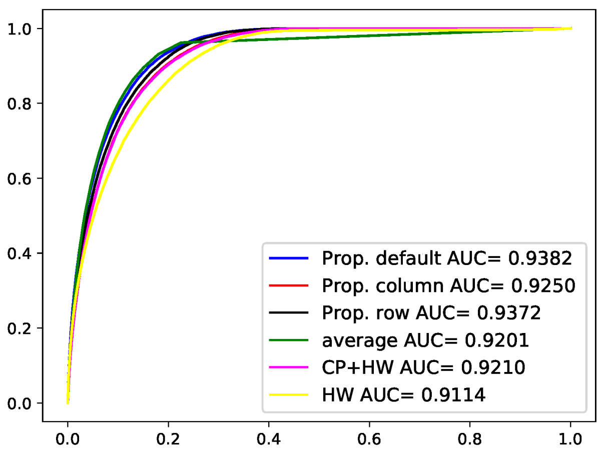

Fig. 5. ROC curves for methods of ensemble learning

prediction. As expected, the AUC values of the proposed methods increased compared to the values in Fig. 4. All methods perform well because the ROC curves are close to the top-left corner. The proposed method has the highest AUC value.

In addition, we carried out additional experiments with LSTM and VAR forecasting methods described in section II. Although previous studies show that LSTM and VAR perform well in time series forecasting, they did not outperform the proposed method on our dataset. In addition, our dataset contains monthly sales data for a period of 24 months, the time period is short, making LSTM unsuitable. Further, LSTMs are non-linear models with different architectures, identifying a suitable architecture is challenging compared to our proposed method.

VI. 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 periodicity of the features by combining NMF and Holt-Winters methods. We also proposed extensions to the proposed method, through matrix decomposition by time invariant features and applying ensemble learning to NMF and Holt-Winters methods 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, 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, in this paper we considered only a sales dataset, it is important to evaluate the performance of our methods with different datasets. Second, our proposed method does not use attribute information, therefore attribute-based prediction is important to further improve the accuracy of our methods.

REFERENCES

[1] Charu C Aggarwal et al. *Recommender systems*. Springer, 2016.
[2] Luc Anselin and Anil K Bera. Introduction to spatial econometrics. *Handbook of applied economic statistics*, pages 237–290, 1998.
[3] Chris Chatfield and Mohammad Yar. Holt-winters forecasting: some practical issues. *The Statistician*, pages 129–140, 1988.
[4] Robert T Clemen. Combining forecasts: A review and annotated bibliography. *International journal of forecasting*, 5(4):559–583, 1989.
[5] Predictivities Based ROC Curve. Medicalbiostatistics. com.
[6] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.
[7] Susan T Dumais, George W Furnas, Thomas K Landauer, Scott Deerwester, and Richard Harshman. Using latent semantic analysis to improve access to textual information. In *SIGCHI*, pages 281–285, 1988.
[8] Daniel M Dunlavy, Tamara G Kolda, and Evrim Acar. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):10, 2011.
[9] Lise Getoor and Christopher P Diehl. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 7(2):3–12, 2005.
[10] Prajakta S Kalekar. Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi School of Information Technology*, 4329008:1–13, 2004.
[11] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
[12] Yehuda Koren. Collaborative filtering with temporal dynamics. In *SIGKDD*, pages 447–456. ACM, 2009.
[13] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.
[14] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104. ACM, 2018.
[15] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791, 1999.
[16] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562. MIT Press, 2001.
[17] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.
[18] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80, 2003.
[19] Xin Luo, Mengchu Zhou, Mingsheng Shang, Shuai Li, and Yunni Xia. A novel approach to extracting non-negative latent factors from non-negative big sparse matrices. *IEEE access*, 4(1), 2016.
[20] Spyros Makridakis and Michele Hibon. The M3-competition: results, conclusions and implications. *International journal of forecasting*, 16(4):451–476, 2000.
[21] Koh Takeuchi, Hisashi Kashima, and Naonori Ueda. Autoregressive tensor factorization for spatio-temporal predictions. In *ICDM*, pages 1105–1110, 2017.
[22] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in neural information processing systems*, pages 847–855, 2016.