Hierarchical Temporal Regularized Matrix Factorization for High-Dimensional Demand Forecasting with Missing Values[1]

Sheng Zhang*, Ming Li†

*School of Industrial and Systems Engineering, Georgia Institute of Technology
†Supply Chain Platform, Alibaba Group

Abstract

Time series forecasting plays an important role in many business and industrial decision processes. In retail businesses, for example, forecasting demand is crucial for having the right inventory available at the right time. While matrix factorization (MF) techniques have been used to address the problem of forecasting high dimensional time series with missing values, they are unable to leverage the hierarchical information. In this paper, we propose hierarchical temporal regularized matrix factorization (HTRMF), which can incorporate hierarchical structure among items for demand forecasting. Experiments on two real datasets from Alibaba show that HTRMF outperforms several existing temporal MF and other approaches for common time series. For this reason, we believe that our method may have broader implications beyond the application considered in this work.

1 Introduction

Time series forecasting plays an important role in many business and industrial decision processes. A typical example of such tasks is demand forecasting. Mismatches of demand and supply are costly to sellers in a competitive market because such mismatches often result in missed sales opportunities, lost profits, excessive expediting costs, lost market share, and poor customer service. To maximize sales and marketing effectiveness, companies must accurately predict future demand and use this information to drive their business operations. This need for accurate predictions of demand is especially important for those involved in e-business due to the ease with which buyers can find alternative sellers that can satisfy their demand. According to the well-known bullwhip effect, a swing in customer demand usually get magnified upstream the supply chain, which may cause significant inefficiencies. Therefore, accurate customer demand is not only the first, but also one of the most important steps in advanced supply chain planning.

There are two major challenges to practitioners in modern forecasting applications. The first is that one is faced with forecasting thousands or millions of related time series. For example, an e-commerce company would track its daily sale information of thousands of items for multiple years and predict the demand for all products it offers. However, the prevalent forecasting methods in use today have been developed in the setting of forecasting individual or a small number of time series. In this approach, model parameters for each given time series are independently estimated from past observations. The model is typically manually selected to account for different factors, such as autocorrelation structure, trend, seasonality, and other explanatory variables.

[1]This work has been published in Alibaba's internal conference.

The second major challenge is that time series data typically involve missing values because of various reasons. For instance, missing values are mainly due to out-of-stock reasons in demand forecasting domain. In practice, only the recorded product sales are often available for estimation. Nonetheless, actual demand may be grater than observed sales when a product sells out, hence sales data are just censored rather than exact observations of demand. Unobservable lost sales are prevalent in retailing where unmet demand arises when products are out-of-stock. Demand predictions based on sales data without accounting for the stock-out effect potentially lead to two types of error. First, the forecasts for out-of-stock products are negatively biased and the extent of the bias depends on the stockout incidence frequency. The second type of error due to stockouts arises when customers purchase an alternative product and hence sales of substitute products increase. In this case, the estimates of demand for substitute products are positively biased.[2]

In most cases, to fit a model for customer demand forecasting we need historical data. We can assume customer demand equals the corresponding sales volume, which is much easier to obtain. However, this is accurate only if it is in stock. When the stock keeping unit (SKU) is out of stock, we can only estimate the demand by imputation.

On the other hand, the number of active SKUs can be huge. It is not unimaginable that customer demand forecast for millions of SKUs is required. To achieve this goal, we can cluster similar SKUs and train a model for each cluster. The level of aggregation is usually a result of compromisation. Theoretically, with bigger cluster the model can capture higher-level features and share statistical strength among individual SKUs. But in the mean time we also need to increase the model capacity, or the details of the SKUs will be overlooked. In practice, this may make the model more difficult to train.

While MF has been a popular choice for many different problems (Cohen et al., 2007; Falahatgar et al., 2016; Mairal et al., 2010; Mei et al., 2017), interestingly, its application to time series analysis has been less developed. As discussed earlier, modern time series are typically high dimensional with many missing values; consequently, the entire time series can be treated in the form of a matrix, and low-rank MF can be quite useful. The recent work of Yu et al. (2016) proposed temporal regularized matrix factorization (TRMF), based on this observation. Its results show that MF is also a powerful tool for time series forecasting and sharing information across time series can improve the forecasting accuracy. Nevertheless, one limitation of Yu et al. (2016) and other existing temporal MF approaches (Chen and Cichocki, 2005; Rallapalli et al., 2010; Zhang et al., 2009; Xiong et al., 2010) is the inability to use hierarchical information among items, which can be exploited to develop a more accurate forecasting method.

In this paper, we propose HTRMF to incorporate hierarchical structure over items into TRMF for time series missing value imputation and forecasting. We demonstrate HTRMF's effectiveness of incorporating hierarchical information through experiments on two datasets from Alibaba. In experiments, we also show that HTRMF outperforms several existing MF approaches and other methods for common time series imputation and forecasting tasks. Although HTRMF is proposed specifically for sales data in this paper, in principle it could be generalized and applied to other data with similar structure. In particular, it could be applied to the data matrix with hierarchy on one side and temporal dependency on the other side. Extending the TRMF framework for imputation and forecast, we introduce a hierarchical structure, with which features from various levels can be captured. In addition, the model can be conveniently trained with distributed computing systems. This makes the hierarchical TRMF model a competitive candidate for large-scale time

[2]Product substitution by customers due to stockouts is one potential source for correlation among time series.

series imputation and forecasting tasks.

The rest of the paper is organized as follows. We first discuss related work in Section 2 and then review TRMF framework in Section 3. In Section 4, we present our HTRMF framework and describe the training and inference procedure. We demonstrate the superiority of the proposed approach via extensive experiments in section 6 and conclude the paper in Section 7.

2 Related Work

2.1 Time-Series Models

Prominent examples of methods for forecasting individual time series include autoregressive (AR) models (Box and Jenkins, 1968) and dynamic linear models (DLM) (Kalman, 1960; West and Harrison, 2006); Hyndman et al. (Hyndman et al., 2008) and Durbin and Koopman (Durbin and Koopman, 2012) provide a unifying review of these and related techniques. But models such as AR and DLM are not well-suited for modern forecasting applications with large and diverse time series corpora, due to their inherent computational inefficiency (Yu et al., 2016). Further, traditional AR and DLM cannot infer shared patterns from a dataset of similar time series, as they are fitted on each time series separately. On the other hand, it is also challenging for many classic time series models to deal with data containing missing values (Anava et al., 2015).

2.2 Existing MF Approaches for Time Series

Because squared Frobenius norm does not take into account the dependencies among the columns of factor matrices, standard MF formulation is not applicable to data with temporal dependencies. Thus, most existing temporal MF approaches (Chen and Cichocki, 2005; Rallapalli et al., 2010; Zhang et al., 2009; Xiong et al., 2010) handle temporal dependencies through graph-based regularizer, where the dependencies are described by a graph. However, graph-based regularization leads to poor forecasting abilities in either cases where there are negative correlations between two time points, or where the explicit temporal dependency structure is not available and has to be inferred.

In a recent approach (Yu et al., 2016), related closely to our work, the authors factorize the matrix of sales data across items and time. A key property of their solution is that, the columns of one of the factor matrices is regularized by an AR constraint. The coefficients of this constraint are learned from the data, and can be used to forecast future values.

In all the previous temporal MF approaches, items are considered to be of the same type or in the same group. However, in some situations items belong to different groups and vary in their types. In such scenarios the structural correlations among items is a significant source of information which could be exploited to improve missing value imputation and forecasting performance.

2.3 Hierarchical Matrix Factorization

There are some previous studies on incorporating hierarchical structure into MF to capture the multi-level information. Menon et al. (Menon et al., 2011) use the hierarchical structure to help factorize the click through rate matrix on advertisements. For predicting plant traits, Shan et al. (Shan et al., 2012) propose a hierarchical probabilistic MF with multi-level plants information. Zhong et al. (Zhong et al., 2012) also propose a hierarchical MF, which divides the rating matrix level-by-level by considering the local contexts and then applies MF to each sub-matrix. Wang et al.

(Wang et al., 2014) propose a hierarchical group MF method to explore and model the structural correlations among users and items.

The aforementioned methods have empirically been shown helpful for improving prediction performance. However, all of them only focus on MF for data without temporal dependency and thus are not applicable to time series forecasting.

3 Background

3.1 Matrix Factorization

MF is a class of collaborative filtering algorithms, originally proposed for recommender systems (Koren et al., 2009). Take modelling a group of user's rating of a group of items for example, the rating of a specific item by a specific user can be approximated by the inner product of the embedding vector of the user and the embedding vector of the item. Thus if we form a rating matrix with entry at (i, j) being the rating of item j by user i, the matrix can be factored into the product of two low-rank matrices, one for the user embeddings and the other for the item embeddings. It is not required that all the entries in the rating matrix are available. We can estimate the embedding matrices from the observed entries only, and use the embeddings to predict the entries that are missing.

For a group of times series observed in the same time window, MF can also be utilized for data imputation. Specifically, let $Y \in \mathbb{R}^{n \times T}$ be the data matrix of n time series of length T. The element $Y(i,t)$ of Y in row i and column t, i.e. the value of time series i at time t, can be approximated by

$$Y(i,t) \approx \mathbf{f}_i^{\mathrm{T}} X(t), \tag{1}$$

where $\mathbf{f}_i \in \mathbb{R}^d$ is the embedding vector for time series i and $X(t) \in \mathbb{R}^d$ is the latent temporal embedding vector at time t as shown in Figure 1. The embedding vectors can be estimated by minimizing the following objective function

$$\min_{Z} \ \frac{1}{2} \sum_{(i,t) \in \Omega_k} \left(Y(i,t) - \mathbf{f}_i^{\mathrm{T}} X(t) \right)^2 + \lambda_F \mathcal{R}_F(F) + \lambda_X \mathcal{R}_X(X), \tag{2}$$

where $F = (\mathbf{f}_1, \cdots, \mathbf{f}_n)$ is the time series embedding matrix, $X = (X(1), \cdots, X(T))$ is the latent temporal embedding matrix, $Z = \{F, X\}$ is the set for all the embedding matrices, $\mathcal{R}_F(\cdot)$, $\mathcal{R}_X(\cdot)$ are the regularization functions (e.g. squared ℓ_2 norm), λ_F, λ_X are the hyper-parameters, and Ω is the set of indices for all the available entries of Y, respectively.

After training the MF model, the missing value of the i-th time series at time t can be estimated by

$$\widehat{Y}(i,t) = \widehat{\mathbf{f}}_i^{\mathrm{T}} \widehat{X}(t), \tag{3}$$

where $\widehat{\mathbf{f}}_i$ and $\widehat{X}(t)$ are the estimated embedding vector for time series i and estimated latent temporal embedding vector at time t, respectively.

3.2 Temporal Regularized Matrix Factorization

Although MF can be utilized for time series data imputation, it is not capable of time series forecasting. This is because the MF model does not impose any constraint on the relation between

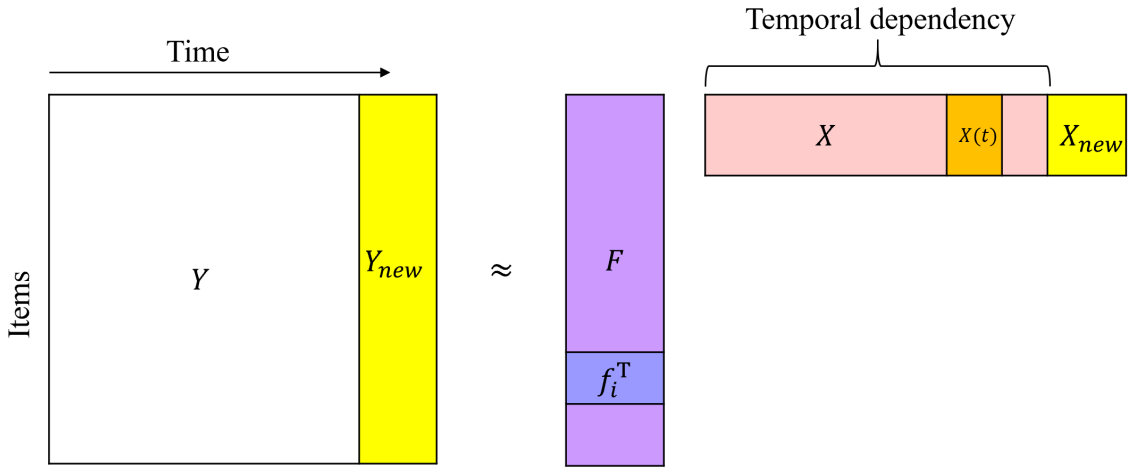

Figure 1. MF for multiple time series. F captures features for each time series in the matrix Y , and X captures the latent and time-varying variables (Yu et al., 2016).

the latent temporal embedding vectors at different time points. As a result we cannot predict $X(T+1)$ even if we know all about $X(1), X(2), \cdots, X(T)$. In order to perform forecasting, we need to introduce additional regularization terms for temporal dependency between the latent temporal embedding vectors, as in the TRMF model (Yu et al., 2016). Mathematically, the embedding vectors are estimated by solving the following optimization problem

$$\min_{Z,\Theta} \frac{1}{2} \sum_{(i,t)\in\Omega_k} \left(Y(i,t) - \mathbf{f}_i^{\mathrm{T}} X(t)\right)^2 + \lambda_F \mathcal{R}_F(F) + \lambda_X \mathcal{T}_M\left(X|W,\eta\right) + \lambda_W \mathcal{R}_W\left(W\right), \tag{4}$$

where

$$\mathcal{T}_M\left(X|W,\eta\right) = \frac{1}{2} \sum_{t=L+1}^{T} ||X(t) - \sum_{l=1}^{L} W(l) \circ X(t-l)||_2^2 + \frac{\eta}{2}||W||_2^2, \tag{5}$$

is the AR regularization term, in which $W(l)$ is the l-th column of the AR coefficient matrix $W \in \mathbb{R}^{d\times L}$, $W(l) \circ X(t-l)$ is the Hadamard (elementwise) product of $W(l)$ and $X(t-l)$, $||\cdot||_2$ is the ℓ_2 norm, η is the regularization coefficient, respectively. In order to regularize the AR coefficient matrix, an addition term $\lambda_W \mathcal{R}_W\left(W\right)$ with regularization coefficient λ_W and regularization function $\mathcal{R}_W\left(\cdot\right)$ is added.

With the estimated AR coefficient matrix $\widehat{W}$ and latent temporal regularized embedding matrix $\widehat{X}$, we can forecast the latent temporal regularized embedding vectors by iteratively calculating

$$\widehat{X}(T+h) = \sum_{l=1}^{L} \widehat{W}(l) \circ \widehat{X}(T+h-l), \quad h = 1, 2, \cdots \tag{6}$$

and forecast the value of the i-th time series at time $T+h$ by

$$\widehat{Y}(i, T+h) \approx \widehat{\mathbf{f}}_i^{\mathrm{T}} \widehat{X}(T+h). \tag{7}$$

4 Hierarchical TRMF

4.1 Motivation

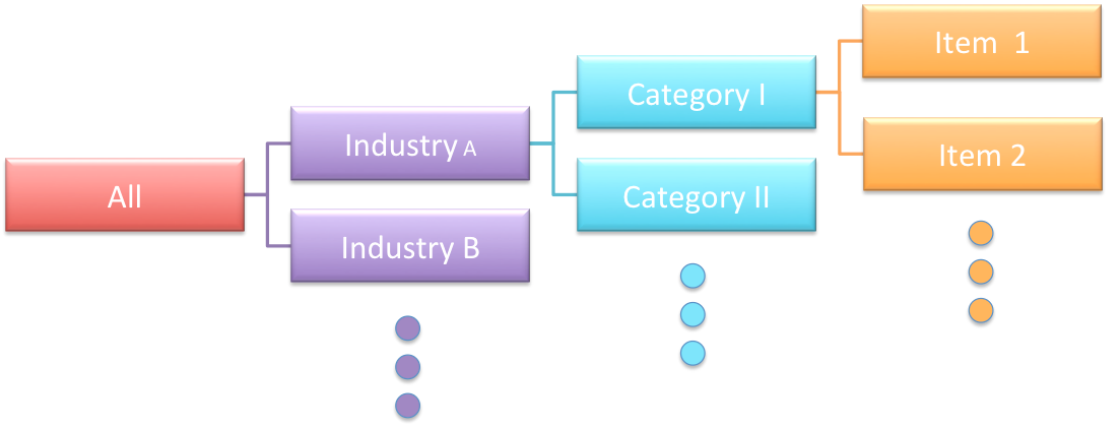

Figure 2: Hierarchy of items.

The key observation that motivated the HTRMF is that, in an e-commerce platform items are hierarchically organized. As shown in Figure 2, a group of items can belong to a category, and several categories can be arranged into an industry. The demand for an item can be affected by several levels of factors, including those related to the category, those related to the industry, and so on. Under the TRMF framework, in theory we can model every levels of factors for all the items in a single model, provided that the dimension of the latent temporal embedding vectors is sufficiently large. However, in practice training the model would be a big challenge. Since the dimensions of the latent temporal embedding vectors and latent item embedding vectors are considerably high, the computation cost may be unacceptable. In addition, too many free parameters can lead to competition between the items and overfitting.

4.2 The Model

As shown in Figure 3, in order to address the issue in the previous subsection, we partition the latent temporal embedding matrix into several parts, each of which corresponds to a different level. We also partition the item embedding matrix accordingly. For illustration purpose we assume a simple hierarchy. There are two categories I and II, which belong to the same industry A. Based on this hierarchy we set the latent temporal embedding vector for items from category I to be a concatenation of $X_A(t)$ and $X_I(t)$, and that for items from category II to be a concatenation of $X_A(t)$ and $X_{II}(t)$, respectively. On one hand, items from category I and items from category II belong to different categories, so we let the latent temporal embedding vectors $X_I(t)$ and $X_{II}(t)$ be totally independent, they can even have different dimensions. But on the other hand, items from categories I and II do belong to the same industry A, so we also reflect this in the model by binding the corresponding latent temporal embedding vectors, i.e. the $x_A(t)$s for the items from category I and the items from category II are exactly the same.

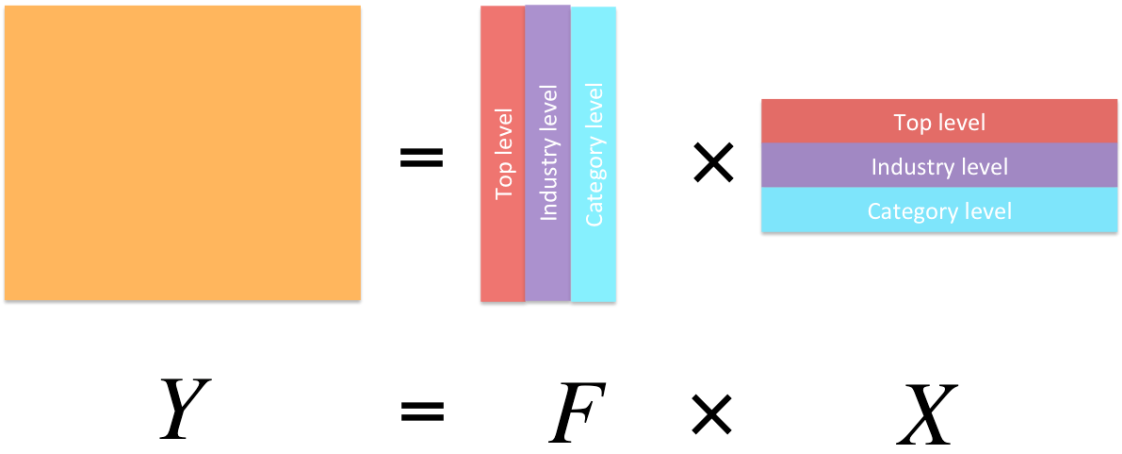

Figure 3: Diagram of the HTRMF model.

4.3 Formulation

For simplicity we assume that the hierarchy consists of two levels, and one item belongs to one group only. This can be easily extended to model the case when there are multiple levels and/or the case when an item/subgroup can belong to multiple groups.

Let $Y \in \mathbb{R}^{n \times T}$ be the data matrix of n time series of length T. For demand imputation and forecasting, the element $Y(i,t)$ of Y in row i and column t is the demand for item i at time t. The items belong to K different categories, so that Y can be partitioned into K submatrices accordingly as follows:

$$Y = \left[\begin{array}{c} Y_1 \\ \hline Y_2 \\ \hline \vdots \\ \hline Y_K \end{array}\right], \tag{8}$$

where the submatrix $Y_k \in \mathbb{R}^{n_k \times T}$ is the data matrix for category k, n_k is the number of items in that category, and $\sum_{k=1}^{K} n_k = n$.

For the t-th column of submatrix Y_k, namely $Y_k(t)$, we can decompose it as following, considering both the factors specific for category k and the factors common for all the K categories:

$$Y_k(t) \approx \underbrace{L_k^{\mathrm{T}} X_k(t)}_{\text{for category k only}} + \underbrace{G_k^{\mathrm{T}} X(t)}_{\text{for all the categories}} = \left[\begin{array}{c|c} L_k^{\mathrm{T}} & G_k^{\mathrm{T}} \end{array}\right] \left[\begin{array}{c} X_k(t) \\ \hline X(t) \end{array}\right], \tag{9}$$

where $L_k \in \mathbb{R}^{d_k \times n_k}$ is the local latent item embedding matrix, and $X_k(t)$ is the t-th column of the local latent temporal embedding matrix $X_k \in \mathbb{R}^{d_k \times T}$, respectively. These matrices L_k and $X_k(t)$ account for the factors specific within category k. Similarly, the factors common for all the K categories are taken care of by matrices $G_k \in \mathbb{R}^{d \times n_k}$ and $X \in \mathbb{R}^{d \times T}$. As stated above, the global latent temporal embedding matrix X is shared among all the K categories.

We can go a step further. It is reasonable to assume that the global latent temporal embeddings affect different items within a category in nearly the same way, except for the scales. In other words,

we can decompose G_k into the following form

$$G_k = \mathbf{g}_k \boldsymbol{\alpha}_k^{\mathrm{T}} = \mathbf{g}_k \left[\ \alpha_{k,1}, \alpha_{k,2}, \cdots, \alpha_{k,n_k} \ \right], \tag{10}$$

where $\alpha_{k,i}$ is the scaling factor for the global effect on the i-th item in category k.

Now we are ready to give the formula for each element of $Y_k(t)$. Let $Y_k(i,t)$ be the i-th element of $Y_k(t)$, and $L_k(i)^{\mathrm{T}}$ be the i-th row of L_k^{T}, then we have

$$Y_k(i,t) \approx \underbrace{L_k(i)^{\mathrm{T}} X_k(t)}_{\text{for category k only}} + \underbrace{\alpha_{k,i} \mathbf{g}_k^{\mathrm{T}} X(t)}_{\text{for all the categories}}. \tag{11}$$

With these notations at hand, we come to the following HTRMF formulation

$$\begin{aligned} \min_{Z,\Theta} \ & \frac{1}{2} \sum_{k=1}^{K} \sum_{(i,t)\in\Omega_k} \left(Y_k(i,t) - L_k(i)^{\mathrm{T}} X_k(t) - \alpha_{k,i} \mathbf{g}_k^{\mathrm{T}} X(t) \right)^2 \\ & + \sum_{k=1}^{K} \left(\lambda_{L_k} \mathcal{R}_{L_k}(L_k) + \lambda_{X_k} \mathcal{T}_{M_k} \left(X_k | W_k, \eta_k \right) + \lambda_{W_k} \mathcal{R}_{W_k} \left(W_k \right) \right) \\ & + \sum_{k=1}^{K} \left(\lambda_{\mathbf{g}_k} \mathcal{R}_{\mathbf{g}_k}(\mathbf{g}_k) + \lambda_{\boldsymbol{\alpha}_k} \mathcal{R}_{\boldsymbol{\alpha}_k}(\boldsymbol{\alpha}_k) \right) + \lambda_X \mathcal{T}_M \left(X | W, \eta \right) + \lambda_W \mathcal{R}_W \left(W \right), \end{aligned} \tag{12}$$

where $Z = \{L_k, \mathbf{g}_k, X_k, X\}$ is the set for all the embedding variables, $\Theta = \{\alpha_{k,i}, W_k, W\}$ is the set for all the parameters, λ_{L_k}, λ_{X_k}, η_k, λ_{W_k}, $\lambda_{\mathbf{g}_k}$, $\lambda_{\boldsymbol{\alpha}_k}$, λ_X, η, λ_W are the hyper-parameters, and Ω_k is the set of indices for all the available entries of Y_k, respectively.

Note that when we set any of $\lambda_{\mathbf{g}_k}$, $\lambda_{\boldsymbol{\alpha}_k}$, λ_X to be positive infinity, the learned term $\alpha_{k,i} \mathbf{g}_k^{\mathrm{T}} X(t)$ in (12) would be zero. In this case the HTRMF model degenerate to K independent conventional TRMF models.

5 Training the Model

5.1 Divide the Model into Submodels

A close inspection of (12) reveals that the embedding variables $Z = \{L_k, \mathbf{g}_k, X_k, X\}$ can be separated into embedding variables $Z_k = \{L_k, X_k\}$ that model the local factors for each category, and embedding variables $Z_G = \{\mathbf{g}_k, X\}$ that model the global factors for all the categories. The corresponding parameters $\Theta = \{\alpha_{k,i}, W_k, W\}$ can also be separated into $\Theta_k = \{\alpha_{k,i}, W_k\}$ and $\Theta_G = \{W\}$ in a similar way. Based on this, we can utilize coordinate descent to train the model. In particular, as illustrated in Figure 4 we alternate between training the local embedding variables together with associated local parameters, and training their global counterparts.

When training for the local embeddings and parameters, we fix the global embeddings and parameters. Since there is no interaction between the local embeddings, we can train the local embeddings $\{Z_k\}$ for each group in parallel, possibly with different executors or computers. For each Z_k we minimize the relevant part of the objective function. In other words, we train the

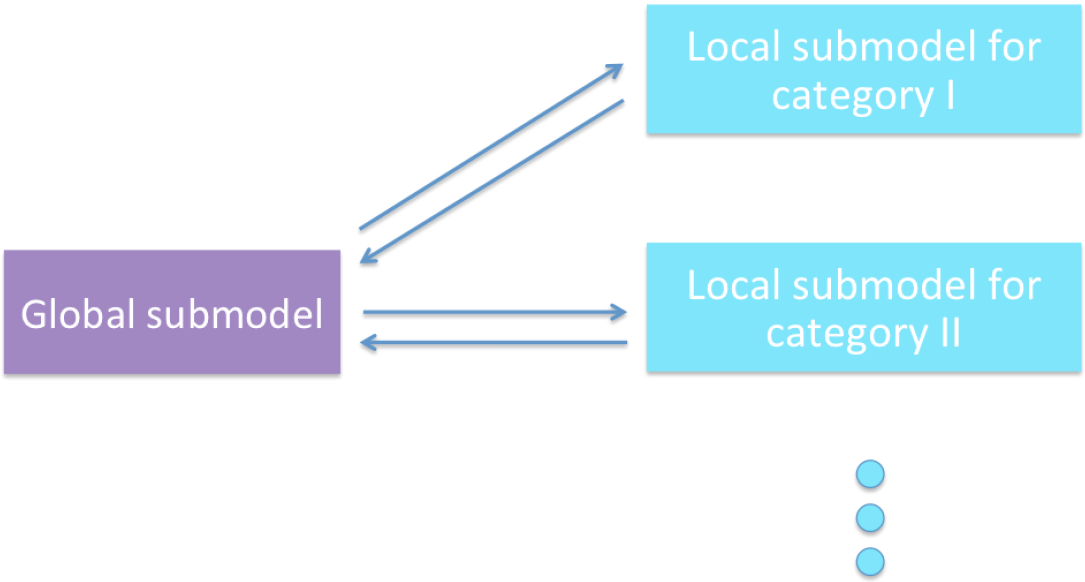

Figure 4: Training the HTRMF model.

submodel for category k by solving the following subproblem

$$\begin{aligned}\min_{Z_k,\Theta_k} \quad & \frac{1}{2}\sum_{(i,t)\in\Omega_k}\left(Y_k(i,t)-L_k(i)^{\mathrm{T}}X_k(t)-\alpha_{k,i}\mathbf{g}_k^{\mathrm{T}}X(t)\right)^2 \\ & +\lambda_{L_k}\mathcal{R}_{L_k}(L_k)+\lambda_{X_k}\mathcal{T}_{M_k}\left(X_k|W_k,\eta_k\right)+\lambda_{W_k}\mathcal{R}_{W_k}\left(W_k\right).\end{aligned} \tag{13}$$

After training the local submodels for the categories separately, we train the global submodel, during which we fix the local embeddings and parameters and solve the global subproblem

$$\begin{aligned}\min_{Z_G,\Theta_G} \quad & \frac{1}{2}\sum_{k=1}^{K}\sum_{(i,t)\in\Omega_k}\left(Y_k(i,t)-L_k(i)^{\mathrm{T}}X_k(t)-\alpha_{k,i}\mathbf{g}_k^{\mathrm{T}}X(t)\right)^2 \\ & +\sum_{k=1}^{K}\left(\lambda_{\mathbf{g}_k}\mathcal{R}_{\mathbf{g}_k}(\mathbf{g}_k)+\lambda_{\boldsymbol{\alpha}_k}\mathcal{R}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_k)\right)+\lambda_X\mathcal{T}_M\left(X|W,\eta\right)+\lambda_W\mathcal{R}_W\left(W\right).\end{aligned} \tag{14}$$

5.2 Training the Submodels

Training the submodels is essentially the same as that in Yu et al. (2016). We set all the regularization terms denoted by $\mathcal{R}(\cdot)$ to be squared ℓ_2 norm. With coordinate descent, we alternately optimize for one specific variable/parameter matrix or vector, with other variables/parameters held fixed. Fortunately, when optimizing for any variable/parameter vector or matrix, we can always convert the minimization problem into

$$\min_{\beta} \quad \frac{1}{2}\left(\mathbf{v}-U\beta\right)^{\mathrm{T}}\left(\mathbf{v}-U\beta\right)+\frac{1}{2}\beta^{\mathrm{T}}C\beta, \tag{15}$$

Table 1. Forecasting with Missing Values results: ND/NRMSE for each methods. Lower values are better. "-" indicates an unavailability due to inablity to forecast.

Dataset	HTRMF-AR	TRMF-AR	TCF	DLM	MF	Mean
Alibaba-1	**0.43/0.78**	0.46/0.83	0.53/0.91	0.51/0.88	-/-	1.09/2.01
Alibaba-2	**0.67/1.16**	0.73/1.24	0.82/1.33	0.78/1.29	-/-	1.23/3.38

Table 2. Missing Value Imputation results on **Alibaba-1** dataset: ND/NRMSE for each methods. Lower values are better.

Dataset	$\frac{\#\text{ observed entries}}{\#\text{ total entries}}$	HTRMF-AR	TRMF-AR	TCF	DLM	MF	Mean
Alibaba-1	0.75	**0.33/0.68**	0.34/0.71	0.45/0.88	0.49/0.90	0.52/0.97	1.29/1.77
Alibaba-1	0.65	**0.36/0.76**	0.37/0.79	0.54/0.97	0.55/0.99	0.62/1.08	1.31/1.85

where $\beta \in \mathbb{R}^q$ is the variable/parameter vector or vectorized matrix, and $\mathbf{v} \in \mathbb{R}^p$, $U \in \mathbb{R}^{p \times q}$, $C \in \mathbb{R}^{q \times q}$ are known vector or matrices. This minimization problem has a known analytical solution

$$\widehat{\beta} = \left(U^{\mathrm{T}}U + C\right)^{-1} U^{\mathrm{T}}\mathbf{v}, \tag{16}$$

which can be conveniently implemented with (sparse) linear equation solvers.

6 Experimental Results

In this section, we perform extensive experiments on time series forecasting and missing value imputation on real-world datasets. We have made our experimental codes publicly available[3] and you can find more reproducibility-related information there.

6.1 Datasets

Alibaba-1 and **Alibaba-2** are two propriety datasets from Alibaba containing weekly item sales information of 2,527 and 1,262 items for 107 weeks, respectively. The time series of sales for each item start and end at same time points. 15.18 % and 25.85% of entries are missing due to out-of-stock reasons. Since stock-out effect usually lasts over a length of time, we observe blocks of data are missing in the two datasets. The hierarchical information of the two datasets is shown in Figure 5 and Figure 6.

[3] https://github.com/xiaojianzhang/exp-htrmf-kdd2019

Table 3. Missing Value Imputation results on **Alibaba-2** dataset: ND/NRMSE for each methods. Lower values are better.

Dataset	$\frac{\#\text{ observed entries}}{\#\text{ total entries}}$	HTRMF-AR	TRMF-AR	TCF	DLM	MF	Mean
Alibaba-2	0.65	**0.46/0.98**	0.48/1.09	0.59/1.14	0.56/1.03	0.64/1.19	2.10/3.27
Alibaba-2	0.55	**0.49/1.03**	0.56/1.15	0.62/1.23	0.55/1.13	0.71/1.26	2.23/3.58

	Category ID	# of items
Industry A	1	249
	2	177
	3	108
	4	96
	5	137
	6	69
	7	232
	8	303
	9	199
	10	88
	11	154
	12	117
	13	57
	14	369
	15	79
	16	93

Figure 5: The two-level hierarchy of **Alibaba-1** dataset.

6.2 Benchmark Methods

We tested the following benchmarks in our experiments:

- HTRMF-AR: The proposed formulation (12) with the AR temporal regularizer.
- TRMF-AR: MF with the AR temporal regularizer proposed in Yu et al. (2016).
- TCF: MF with the simple temporal regularizer proposed in Xiong et al. (2010).
- MF: Standard MF (Koren et al., 2009).
- DLM: Dynamic linear model (Kalman, 1960; Ghahramani and Hinton, 1996).
- Mean: this is the baseline approach, which for each time series predicts everything to be its historical mean.

For each method and data set, we perform a grid search over various parameters (such as latent dimensions and regularization coefficients) following a rolling validation approach described in Nicholson et al. (2014) and report the best ND and NRMSE.

6.3 Evaluation Criteria

As the range of values varies in two time series datasets, we compute two normalized metrics namely normalized deviation (ND) and normalized root-mean-square error (NRMSE) whose definition are given as follows (Yu et al., 2016):

$$\text{ND} = \frac{\sum_{(i,t)\in\Omega_{test}} |\widehat{Y}(i,t) - Y(i,t)|}{\sum_{(i,t)\in\Omega_{test}} |Y(i,t)|} \tag{17}$$

	Category ID	# of items
Industry B	1	238
	2	119
	3	177
	4	203
	5	89
	6	63
	7	155
	8	218

Figure 6: The two-level hierarchy of **Alibaba-2** dataset.

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{|\Omega_{test}|}\sum_{(i,t)\in\Omega_{test}}\left(\widehat{Y}(i,t)-Y(i,t)\right)}}{\frac{1}{|\Omega_{test}|}\sum_{(i,t)\in\Omega_{test}}|Y(i,t)|} \tag{18}$$

where Ω_{test} denotes the index set for test data points, and $Y(i,t)$ and $\widehat{Y}(i,t)$ represent true value and predicted value, respectively.

6.4 Forecasting with Missing Values

We compare the methods on the task of forecasting in the presence of missing values in the training data. For each datasets, we repeat the experiment in Yu et al. (2016) that considers 6-week ahead forecasting and use last 27 weeks as the test periods. The detailed results are shown in Table 1. We can clearly observe the superiority of HTRMF-AR. Our proposed approach outperforms all other considered approaches in forecasting performance.

6.5 Missing Value Imputation

We next consider the case of imputing missing values in the data. As in Li et al. (2009), we assume that blocks of data are missing, corresponding to stock-out effect for example, over a length of time. To create data with missing entries, we first fixed the percentage of data that we were interested in observing, and then uniformly at random occluded blocks of a predetermined length (3 for the two datasets). The goal was to predict the missing values. Note that, since there are already missing values which are unknown to us in centain time series in each datasets, we create different number of missing values, which we know the true values, for each time series so as to balance the total number of missing entries among all time series. Table 2 and Table 3 show that HTRMF-AR outperforms the methods we compared to on all cases.

7 Conclusion

In this paper, we focus on large-scale demand forecasting with missing values. We have proposed HTRMF which can incorporate hierarchical information among items into temporal MF. Experimental results on two real-world datasets from Alibaba show that HTRMF improves the forecasting and missing value imputation accuracy by effectively using the hierarchical structure over items.

For future works, we are interested in extending our HTRMF to (i) incorporate additional information, such as additional features along with the observed time series, and (ii) process data as a stream using online MF.

Acknowledgments

This work was supported in part by the Alibaba Group research intern project.

References

O. Anava, E. Hazan, and A. Zeevi. Online time series prediction with missing data. In *International Conference on Machine Learning*, pages 2191–2199, 2015.

G. E. Box and G. M. Jenkins. Some recent advances in forecasting and control. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 17(2):91–109, 1968.

Z. Chen and A. Cichocki. Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints. *Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep*, 68, 2005.

S. Cohen, W. Nutt, and Y. Sagic. Deciding equivalances among conjunctive aggregate queries. *J. ACM*, 54(2), Apr. 2007. doi: 10.1145/1219092.1219093. URL http://doi.acm.org/10.1145/1219092.1219093.

J. Durbin and S. J. Koopman. *Time series analysis by state space methods*, volume 38. Oxford University Press, 2012.

M. Falahatgar, M. I. Ohannessian, and A. Orlitsky. Near-optimal smoothing of structured conditional probability matrices. In *Advances in Neural Information Processing Systems*, pages 4860–4868, 2016.

Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.

R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

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

L. Li, J. McCann, N. S. Pollard, and C. Faloutsos. Dynammo: Mining and summarization of coevolving sequences with missing values. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 507–516. ACM, 2009.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

J. Mei, Y. De Castro, Y. Goude, and G. Hébrail. Nonnegative matrix factorization for time series recovery from a few temporal aggregates. In *International Conference on Machine Learning*, pages 2382–2390, 2017.

A. K. Menon, K.-P. Chitrapura, S. Garg, D. Agarwal, and N. Kota. Response prediction using collaborative filtering with hierarchies and side-information. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 141–149. ACM, 2011.

W. B. Nicholson, D. S. Matteson, and J. Bien. Structured regularization for large vector autoregression. *Cornell University*, 2014.

S. Rallapalli, L. Qiu, Y. Zhang, and Y.-C. Chen. Exploiting temporal stability and low-rank structure for localization in mobile networks. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 161–172. ACM, 2010.

H. Shan, J. Kattge, P. Reich, A. Banerjee, F. Schrodt, and M. Reichstein. Gap filling in the plant kingdom—trait prediction using hierarchical probabilistic matrix factorization. *arXiv preprint arXiv:1206.6439*, 2012.

X. Wang, W. Pan, and C. Xu. Hgmf: Hierarchical group matrix factorization for collaborative recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 769–778. ACM, 2014.

M. West and J. Harrison. *Bayesian forecasting and dynamic models*. Springer Science & Business Media, 2006.

L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 211–222. SIAM, 2010.

H.-F. Yu, N. Rao, and I. S. Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in neural information processing systems*, pages 847–855, 2016.

Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-temporal compressive sensing and internet traffic matrices. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 267–278. ACM, 2009.

E. Zhong, W. Fan, and Q. Yang. Contextual collaborative filtering via hierarchical matrix factorization. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 744–755. SIAM, 2012.