2015 IEEE International Conference on Data Mining

Missing Value Estimation for Hierarchical Time Series: A Study of Hierarchical Web Traffic

Zitao Liu
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15213
Email: ztliu@cs.pitt.edu

Yan Yan, Jian Yang
Yahoo! Labs
701 First Avenue
Sunnyvale, CA 94089
Email: {chrisyan,jianyang}@yahoo-inc.com

Milos Hauskrecht
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15213
Email: milos@cs.pitt.edu

***Abstract*—Hierarchical time series (HTS) is a special class of multivariate time series where many related time series are organized in a hierarchical tree structure and they are consistent across hierarchy levels. HTS modeling is crucial and serves as the basis for business planning and management in many areas such as manufacturing inventory, energy and traffic management. However, due to machine failures, network disturbances or human maloperation, HTS data suffer from missing values across different hierarchical levels. In this paper, we study the missing value estimation problem under hierarchical web traffic settings, where the user-visit traffic are organized in various hierarchical structures, such as geographical structure and website structure. We develop an efficient algorithm, *HTSImpute*, to accurately estimate the missing value in multivariate noisy web traffic time series with specific hierarchical consistency in HTS settings. Our *HTSImpute* is able to (1) utilize the temporal dependence information within each individual time series; (2) exploit the intra-relations between time series through hierarchy; (3) guarantee the satisfaction of hierarchical consistency constraints. Results on three synthetic HTS datasets and three real-world hierarchical web traffic datasets demonstrate that our approach is able to provide more accurate and hierarchically consistent estimations than other baselines.**

I. Introduction

Many organizations in business, economics and information technology domains operate in a multi-item, multi-level environment. Data collected and archived by these organizations over time often reflect this structure and consist of time series organized in a hierarchical tree structure. These related multivariate time series are referred to as *Hierarchical Time Series* (HTS) [1].

The modeling and analysis of hierarchical time series could be utilized for efficient business planning and management. For example, in manufacturing inventory management, daily gross demands of families of items and daily demands of individual items form an HTS [2]. Finding temporal patterns on hierarchical time series modeling demands can help one to understand products' popularity in market. Another example of HTS are the counts of daily page view (PV) representing the web traffic of the internet business. The page view counts are naturally organized into regional or web page hierarchy deployed by the company. Hierarchical web traffic modeling and analysis directly relates to the companies' online display advertising, which is one of the most profitable business models for Internet services. Accurate forecasting of PV traffic helps the companies (1) understand the "demand" and traffic from different geographical locations; (2) provide useful insights about user behaviors and patterns; (3) set advertisement listing price based on traffic demands in order to maximize the revenue.

The objective of this work is to study models and methods suitable for HTS of web traffic. While many advances were made in individual or general multivariate time series modeling, much less attention has been devoted to special multivariate time series with hierarchical dependences. For example, individual PV traffic modeling is not precise enough since web traffic usually organized in certain hierarchy and they tend to interact and co-move with each other over time. In IT industry, we are usually interested in and focus on a group of web traffic with different types of hierarchies, such as *geographic* hierarchy and *website* hierarchy.

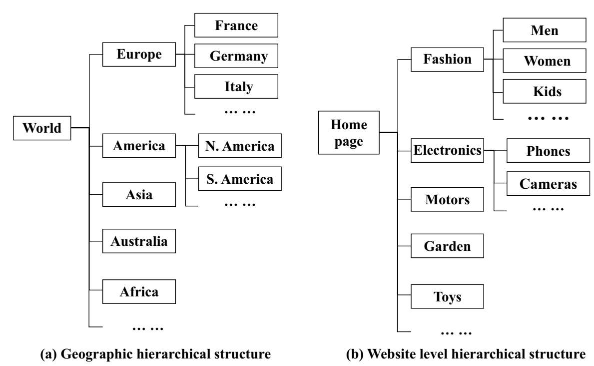

Figure 1. Geographic and website hierarchical structures of web traffic.

In a geographic hierarchy, daily PV time series represent the amount of traffic coming from certain areas and they are arranged through a geographic hierarchy (see Figure 1(a)). Internet companies pay attention to the locations where those traffic demands are generated. More specifically, the web traffic geographical distribution plays an important role in company's resource allocation, effective advertisement disseminations, demographic users targeting, etc. On the other hand, in a website hierarchy, daily PV time series represent the volumes of daily traffic originated from the corresponding web pages and they are organized via a website hierarchy (see Figure 1(b)). Intuitively, the higher volume of PVs usually indicates a more profitable online display ads placement. In a website hierarchy, even though the Internet users might access the child

1550-4786/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDM.2015.58

web page directly without accessing the parent page, which leads to the constraints violation imposed by the hierarchy, we intentionally adjust the parent daily PV as the sum of child daily PVs, which is a common practice in industry due to the ease of data storage, better insights of each web-sector popularity, and convenience for more efficient advertising planning.

Like other real-world time series data, HTS data may come with missing values. This may happen due to various reasons, such as machine malfunctioning, network disturbances, human inappropriate operations, etc. For example, Facebook had its the worst outage in over five years in 2010 due to a server error that caused the corruption of its automated system. In January 2015, due to a bad code update from engineers, the web page tracker of Bing search engine went down for over four hours, which results in more than 300 millions searching queries archive lost to database. All these missing value problems severely affect the temporal pattern modeling and correspondingly the time series forecasting procedures, since the majority of time series modeling techniques assume sequences are fully observed, such as temporal predictive patterns extraction [3], ARIMA models [4], structural time series models [5], etc. Missing value problem easily breaks down the temporal models' common assumption that observation is obtained at every time stamp, which makes aforementioned models inapplicable in real settings.

Estimating the missing values in HTS is more challenging compared to traditional missing value imputation. In HTS any contamination will propagate over the entire structure: a missing value in the lower level will impair all the upper level observations. Even though some recent studies have been able to estimate missing values in more sophisticated ways, their estimates are not consistent when different time series in the same hierarchical structure yield to some critical constraints in volume, which makes those methods not valid solutions for HTS imputation.

In this paper, we develop a novel missing value estimation algorithm called *HTSImpute* which

1) utilizes the temporal dependences within each single time series via adopting a robust local weighted smoothing.
2) exploits the intra-relations between different time series through a tuning-free subspace projection.
3) guarantees that imputations satisfy the hierarchical consistency constraints by an optimal adjustment projection.

To the best of our knowledge, there has been no prior research existing to address the issue of estimating the missing values for HTS. The remainder of the paper is organized as follows: Section II provides a review on existing missing value imputation methods and the optimal combination forecasts for HTS. Section III describes three key steps and the overall procedure of the missing value estimation algorithm. Section IV focuses on the missing value estimations for three synthetic datasets and three real-world page view traffic datasets and compares the results to alternative imputation approaches. Furthermore, we empirically evaluate the efficiency of *HTSImpute*. Section V summarizes this work and outlines future extensions.

II. Background

In this section we briefly review two lines of work relevant to the problem studied of this paper.

A. Missing Value Imputation

Various imputation methods have been proposed to estimate the missing values in multivariate time series. These methods can be divided into three categories: (1) *Regression methods* that repeatedly fit a regression model on each sequence and predict the missing values by using the fitted (non)parametric model [6]; (2) *Subspace methods* such as matrix factorization [7], [8] and matrix completion [9] that apply subspace projection techniques to multivariate time series and utilize their low-rank property to capture the relations among the different time series data; (3) *Hidden variable methods*, such as probabilistic PCA [10], that model missing values using hidden random variables and apply the EM algorithm to iteratively reestimate them through their expectations.

B. Joint HTS Forecasting

An HTS defines a multivariate time series with equality constraints restricting their respective values. A forecasting model that jointly predicts the future values of HTS may or may not respect these constraints. A forecasting approach consistent with the HTS constraints was proposed and studied by [1]. The approach works by independently forecasting all time series which lie in the lowest level of the hierarchy and by applying a regression model to optimally combine and reconcile forecasts so that they are adding up appropriately across the hierarchy. The approach leads to unbiased forecasts and optimizes variances among all joint forecasts. Details of the approach can be found in Section III-C2.

III. Methodology

In this section, we present our *HTSImpute* algorithm for consistent missing value estimation for HTS. The algorithm works in the two stages. It first estimates the missing values for each time series individually, and after that it iteratively reestimates the missing values to assure the consistency of the estimates across the hierarchy. Before presenting these two stages we first introduce the terminology and notation.

A. Terminology and notation

Given an HTS of length T, we represent each time series i as a $T \times 1$ vector $\mathbf{y}_i$, and where y_{ij} denotes the *j*th observation of $\mathbf{y}_i$. All individual time series comprise an $n \times T$ matrix $\mathbf{Y}$, where *n* is the total number of time series in HTS such that each time series corresponds to a row of $\mathbf{Y}$, that is, $\mathbf{Y} = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_n]^\top$. Let $\mathbf{y}_{:,j}$ be the *j*th column of matrix $\mathbf{Y}$. Since HTS follows a tree structure, we adopt the terminology used for describing the trees. A *root* represents the top most node in a hierarchy, such as $\mathbf{y}_1$ in Figure 2. A *parent* is a time series whose observed value at every time stamp *t* is equal to the sum of values observed at the same time from its children. A *child* is a time series that is one of the components forming a time series at upper levels. A *leaf* node represents a time series at the lowest level of a hierarchy. Without loss of generality, we can rewrite $\mathbf{Y} = [\mathbf{P}; \mathbf{L}]$, where $\mathbf{P}$ as a $p \times T$ matrix which contains all the parents in $\mathbf{Y}$; and $\mathbf{L}$ is an $l \times T$ matrix which contains all leaf nodes in $\mathbf{Y}$. Note that $n = p + l$.

Let $\mathbf{h}$ define an $n \times 1$ indicator vector where each element h_i indexes the parent of $\mathbf{y}_i$ [1], and let $\mathbf{W}$ represent the missing value indicator matrix where $w_{ij} = 1$ indicates that y_{ij} is missing in $\mathbf{Y}$. We use $\hat{\mathbf{Y}}$, $\hat{\mathbf{P}}$, and $\hat{\mathbf{L}}$ to denote values of $\mathbf{Y}$, $\mathbf{P}$, $\mathbf{L}$ after missing value imputation such that $\hat{\mathbf{Y}} = [\hat{\mathbf{P}}; \hat{\mathbf{L}}]$. In general, given a hierarchy, a parent equals to the sum of children. For example, in Figure 2, $\mathbf{Y} = [\mathbf{y}_1\mathbf{y}_2 \dots \mathbf{y}_7\mathbf{y}_8]^\top$ is an HTS with $\mathbf{h} = [0, 1, 1, 2, 2, 3, 3, 3]$. $\mathbf{y}_1$ is the root; $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{y}_3$ are parents; $\mathbf{y}_2$, $\mathbf{y}_3$, $\mathbf{y}_4$, $\mathbf{y}_5$, $\mathbf{y}_6$, $\mathbf{y}_7$ and $\mathbf{y}_8$ are children; $\mathbf{y}_4, \mathbf{y}_5, \mathbf{y}_6, \mathbf{y}_7$ and $\mathbf{y}_8$ are leaves. Here $y_{1i} = y_{2i} + y_{3i}$, $y_{2i} = y_{4i} + y_{5i} + y_{6i}$, and $y_{3i} = y_{7i} + y_{8i}$ where $i = 1, \dots, T$. In terms of matrix operations we use $\otimes$ to denote element-wise matrix multiplication.

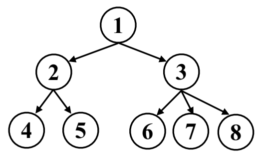

Figure 2. An HTS notation illustration.

B. Modeling Temporal Dependences

The first stage of *HTSImpute* algorithm aims to estimate the missing values by considering temporal dependences among time series values. Simply putting zeros or sample means into the missing positions ignores the temporal dependence among consecutive observations which leads to unsatisfactory results. Furthermore, due to the intricate nature of time series, fitting a specific function to the entire time series easily leads to either underfitting or overfitting problems. To implement the first stage of *HTSImpute*, we choose a nonparametric robust locally weighted regression model, *LO*cal regr*ESS*sion (LOESS) [11]. Briefly, LOESS fits a low-degree polynomial to the neighboring subset of data for each missing position. The polynomial is fitted by using weighted least squares, which heavily weights observations that are close to missing positions. Details about LOESS can be found in [11].

C. Enforcing HTS Consistency

In the second stage the *HTSImpute* algorithm aims to assure the consistency of the missing value estimates across the hierarchy. More specifically, at any time stamp *t*, the value of a parent time series should be always equal to the sum of values from its children.

The approach we propose reestimates the missing values in HTS iteratively by repeating the two steps: (1) performing a subspace projection where time series values are projected to a lower dimensional subspace and (2) by using the subspace to reestimate the values for all time series in HTS that are consistent with the hierarchy.

1) Subspace Projection: The restrictions imposed by the hierarchy may limit the rank of matrix $\mathbf{Y}$. Briefly under equality constraints induced by the hierarchy, any time series at node i can be expressed in terms of time series observed at its leaf nodes, that is, leaf nodes of the subtree rooted at node i. We capture this intuition in the following theorem.

[1] We use 0 to represent the parent of the root node

Theorem 1. The HTS matrix $\mathbf{Y}$ has rank $r \leq \min(l, T)$.

The finding expressed in Theorem 1 lets us project observational time series matrix $\mathbf{Y}$ onto a rank-*r* matrix. For most realistic cases we expect $T \gg l$. Then using $r = l$ gives us an efficient "tuning-free" algorithm which is a huge advantage when compared to existing subspace projection approaches, such as appropriate matrix rank satisfaction [7], [8] or penalty parameter selection [9] that rely on cross validation for tuning the parameters and as a results can be very time consuming.

More specifically, at each iteration k of *HTSImpute*, we project $\mathbf{Y}^{(k)}$ to a *r*-rank matrix through a projection operator $\mathcal{P}_{\text{r-SVD}}$, defined as follows:

$$\mathbf{Y}^{(k+1)} = \mathcal{P}_{\text{r-SVD}}\Big(\mathbf{W} \otimes \mathbf{Y} + (1 - \mathbf{W}) \otimes \mathbf{Y}^{(k)}\Big) \tag{1}$$

where $\mathcal{P}_{\text{r-SVD}}$ is the unweighted rank-*r* approximation of $\mathbf{X}$, computed from the Singular Value Decomposition(SVD) as follows: $\mathcal{P}_{\text{r-SVD}}(\mathbf{X}) = \mathbf{U}\mathbf{\Lambda}_r\mathbf{V}^\top$, where $\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top$ and $\mathbf{\Lambda}_r = \operatorname{diag}(\Lambda_1, \dots, \Lambda_r, 0, \dots, 0)$.

The subspace projection operator $\mathcal{P}_{\text{r-SVD}}$ truncates the singular values by only keeping the first *r* largest ones. This is a special case of weighted low rank approximation with a fixed rank where the observation indicator matrix $\mathbf{W}$ is treated as a zero/one weighting matrix [7].

2) Hierarchical Consistency Projection: For each column $\mathbf{y}_{:,t}$ in $\mathbf{Y}$, all the observations (including the missing value estimations) have to be projected back to the hierarchical consistent space. We adopt the joint forecast strategy from [1] to conduct this projection. Assume that our hierarchical observation sequences $\mathbf{Y}$ are corrupted by Gaussian noise $\epsilon_{\mathbf{Y}}$ which follows the same hierarchical consistency constraints. The optimal joint forecast strategy assumes that any observation sequence from HTS is a linear combination of the true means of leaf nodes $\hat{\mathbf{L}}$ ($\mathbf{L} = \hat{\mathbf{L}} + \epsilon_{\mathbf{L}}$) [1]. Then $\mathbf{Y}$ can be expressed concisely as

$$\mathbf{Y} = \mathbf{\Omega} \cdot \hat{\mathbf{L}} + \epsilon_{\mathbf{Y}} = \mathbf{\Omega} \cdot (\hat{\mathbf{L}} + \epsilon_{\mathbf{L}}) \tag{2}$$

where $\mathbf{\Omega}$ is the $n \times l$ summing matrix and each row of $\mathbf{\Omega}$, noted as Ω_i, defines the linear combination coefficients for the corresponding time series i. $\mathbf{\Omega}$ is made of two parts: (1) parent linear combination coefficients matrix $\mathbf{\Omega}_{\mathbf{P}}$; (2) the $l \times l$ identity matrix $\mathbf{I}_{l \times l}$. $\mathbf{\Omega} = [\mathbf{\Omega}_{\mathbf{P}}; \mathbf{I}_{l \times l}]$. $\mathbf{\Omega}_{\mathbf{P}}$ is built in a bottom-up fashion row by row, where $\Omega_{P_i} = \sum_{j=1}^{n} \Omega_j \cdot \mathbb{1}_{\{h_j = i\}}$. For instance, the summing matrix for HTS in Figure 2 is $\mathbf{\Omega} = [1\ 1\ 1\ 1\ 1; 1\ 1\ 1\ 0\ 0; 0\ 0\ 0\ 1\ 1; \mathbf{I}_{5 \times 5}]$.

According to eq.(2), we define the hierarchical consistency projection operator by using the ordinary least square as follows:

$$\mathcal{P}_{\text{HTS}}(\mathbf{Y}, \mathbf{\Omega}) = \mathbf{\Omega}(\mathbf{\Omega}^\top\mathbf{\Omega})^{-1}\mathbf{\Omega}^\top\mathbf{Y} \tag{3}$$

Using eq.(3), the hierarchically inconsistent time series observations are projected back to the corresponding consistent space defined by Theorem 1.

D. HTSImpute Summary

The *HTSImpute* method is summarized in Algorithm 1.

Special Cases: According to Theorem 1, the rank of $\mathbf{Y}$ is $r \leq \min(l, T)$. In most cases, we expect the length of time series is much larger than the number of leaf nodes in hierarchy, i.e., $T \gg l$ and the subspace projection significantly reduces the dimensionality of the problem. However, when the number of leaf nodes exceeds the length of sequences($l > T$), $\mathbf{Y}$ becomes a full rank matrix and the subspace projection step can be skipped.

Complexity Analysis: In every iteration, *HTSImpute* performs two projections: subspace projection ($\mathcal{P}_{\text{r-SVD}}(\cdot)$) and hierarchical consistency projection ($\mathcal{P}_{\text{HTS}}(\cdot)$). The former requires $\mathcal{O}(nT \cdot \min(n, T))$ to have rank r matrix approximation and the latter needs $\mathcal{O}(l^3)$ to conduct the hierarchical adjustments. The total running time complexity per iteration is $\mathcal{O}(nT \cdot \min(n, T) + l^3)$. In most cases, the time series' length is much larger than the number of individual time series in the hierarchy, i.e., $T \gg n > l$, which leads to a $\mathcal{O}(n^2 T)$ per-iteration.

Algorithm 1 *HTSImpute*: Hierarchical Time Series Imputation.

INPUT:
- A $n \times T$ partially observed HTS matrix $\mathbf{Y} = [\mathbf{P}; \mathbf{L}]$
- A $n \times T$ observation indicator matrix $\mathbf{W}$
- A $n \times l$ summing matrix $\mathbf{\Omega}$

PROCEDURE:

// Initialize the missing value through *LOESS*
$\mathbf{Y}^{(0)} = \text{Loess}(\mathbf{Y}, \mathbf{W})$
repeat
  // Subspace projection
  $\mathbf{Y}^{(k)} = \mathcal{P}_{\text{r-SVD}}\ (\ \mathbf{W} \otimes \mathbf{Y} + (1 - \mathbf{W}) \otimes \mathbf{Y}^{(k)}\ )$
  // Hierarchical consistency projection
  $\mathbf{Y}^{(k)} = \mathcal{P}_{\text{HTS}}(\mathbf{Y}^{(k)}, \mathbf{\Omega})$
until convergence

OUTPUT:
- $\hat{\mathbf{Y}}$ estimated HTS

IV. Experiment

We compare our *HTSImpute* missing value estimation to five other methods from different categories mentioned in Section II: local regression by LOESS (LOESS) [11]; non-negative matrix factorization based on both Kullback-Leibler divergence (NMF_K) [8] and Euclidean distance (NMF_E) [12]; standard matrix completion (MC) using *SoftImpute* [9]; weighted low rank approximation (wLRA) [7]; and probabilistic principle component analysis (pPCA) [10].

We evaluate and compare the performances of different methods by calculating both the average Mean Absolute Percentage Error (MAPE) [13] and the average Hierarchical Consistency Gap (HCG) [14] for the missing value estimations. Avg-MAPE measures the deviation proportion from the estimated values to the truth and the Avg-HCG evaluates the hierarchical consistency quality by measuring differences between parent and the sum of children. The Avg-MAPE and Avg-HCG are defined as follows:

$$\text{Avg-MAPE} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{T} (1 - w_{ij}) \cdot |1 - \hat{y}_{ij}/y_{ij}|}{\sum_{i=1}^{n} \sum_{j=1}^{T} (1 - w_{ij})} \times 100\%$$

$$\text{Avg-HCG} = \frac{1}{pT} \sum_{i=1}^{p} \sum_{j=1}^{T} |1 - \sum_{k=1}^{n} \mathbb{1}_{\{h_j = k\}} \cdot \hat{y}_{kj}/\hat{p}_{ij}|$$

where $\hat{y}_{ij}$ and $\hat{p}_{ij}$ are the *ij*th elements from $\hat{\mathbf{Y}}$ and $\hat{\mathbf{P}}$.

A. Synthetic Datasets

We perform experiments on synthetic datasets to verify that the proposed method is performing reasonably well against various hierarchical structures: *syn_wide*, *syn_balance* and *syn_deep* (depicted in Figures 3(a)-(c)). We follow [1] for the simulated time series generation. Each leaf time series in the synthetic datasets is generated by an ARIMA(p, d, q) process with d taking values 1 or 2 with equal probability, and p and q each taking values 0, 1 and 2 with equal probability. For each simulated leaf time series, the parameters of each ARIMA process are chosen randomly based on a uniform distribution over the stationary and invertible regions. Then we obtain the parent time series by aggregating the corresponding child time series.

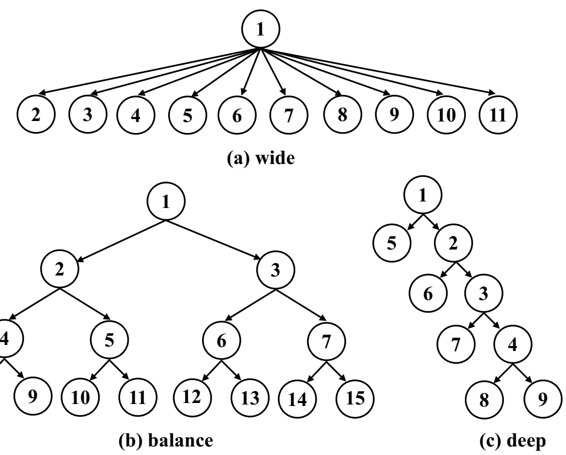

Figure 3. Synthetic HTS structure for *syn_wide*, *syn_balance* and *syn_deep*.

B. Real-World Datasets

In order to evaluate the performance of our proposed method in the real-world settings, we perform extensive comparisons on web traffic datasets with different types of hierarchies: *EMEA* dataset, *FP* dataset, and *Media* dataset from a popular web portal site and their corresponding hierarchical structures are shown in Figure 4.

1) EMEA dataset: The *EMEA* dataset is a bi-level HTS dataset with geographic hierarchical structure (shown in Figure 4(a)). *EMEA* records daily page view traffic time series coming from the Europe, the Middle East and Africa (EMEA). It contains 1 parent (E_1) and 7 leaves, which represent the daily PV from France(E_1), Germany(E_2), Greece(E_3), Italy(E_4), Spain(E_5), UK(E_6) and others(E_7). It ranges from 2009/12/08 to 2014/03/06.

2) FP dataset: The *FP* dataset is a tri-level HTS dataset representing the traffic volumes to the front page of a popular web portal site (shown in Figure 4(b)). *FP* dataset records daily page view traffic to the front page via different channels. In Figure 4(b), F_2 and F_3 represent the front page daily PV from mobile devices and the desktop machines and F_4, F_5 and F_6 represent the traffic from either mobile app, mobile wap or tablet. It ranges from 2013/01/01 to 2015/03/15.

3) Media dataset: The *Media* dataset is a four-level HTS dataset representing the daily page views to the media category from a popular web portal site (shown in Figure 4(c)). The root node in the hierarchy represent the total traffic volumes belong to "Media" (M_1) and M_2 to M_7 represent the daily PV from auto, finance, news, sports, celebrity and others respectively. For the "sports" traffic (M_5), similar to the *FP* dataset, we monitor the different channels of incoming traffic to "sports" where M_8 and M_9 represent the front page daily PV from mobile devices and the desktop machines and M_{10}, M_{11} and M_{12} represent the traffic from our fantasy football app, sports mobile app and mobile wap. It ranges from 2013/01/01 to 2015/03/15.

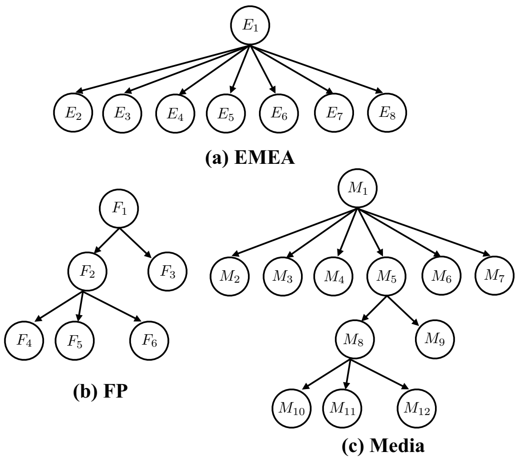

Figure 4. HTS structure for *EMEA*, *FP*, *Media* datasets.

C. Experimental Setting

To examine proposed algorithm's performance, we test *HTSImpute* against other baselines with different missing percentages. The missing percentages control the amount of observations needed to be estimated. In our experiments, we set the missing percentage to be 1%, 3%, 5%, 10%, 15%, and 20%. We randomly generate the missing positions in each run and propagate the missing values through the hierarchy: any missing from the child node will lead to a missing value of its parent. Furthermore, we recover the missing values by the equality constraints if only one value is missing from the parent-child tree unit. Under each setting, we repeat the experiment 10 times and report the average results.

D. Results

We conduct a set of comprehensive experiments to evaluate our *HTSImpute* from the following three aspects: (1) imputation accuracy; (2) imputation consistency; and (3) efficiency.

1) Imputation Accuracy: The Avg-MAPE results on both synthetic and real-world datasets are shown in Table I and Table II. It turns out that *HTSImpute* outperforms all other methods in terms of Avg-MAPE. The missing value estimation results are statistically different at 0.05 level for all datasets from each method. We determined the significance by running the pairwise t-test comparing the *HTSImpute* to all other methods on corresponding estimates. With increment of missing percentages, the Avg-MAPE of our *HTSImpute* goes up due to the fact that when missing percentage gets bigger, it is the more difficult to conduct the accurate estimations in general. With missing percentage changes from 1% to 20%, Avg-MAPEs of *HTSImpute* have increased by 42%, 30%, 0.5% for *syn_wide*, *syn_balance*, *syn_deep* separately, which shows that our method is more robust to the unbalanced tree structures that are usually more common in a website hierarchy. Since in the website hierarchy organization, some subcategories are more popular than others so having "deeper" hierarchies. The same phenomena also happens in the real-world data experiments, where the Avg-MAPEs increased by 233% in the *Media* dataset but 16% in the *FP* dataset.

2) Imputation Consistency: Tables III and Table IV show the Avg-HCG performance in $\log_{10}$ scale for each method on both synthetic and real-world datasets. As we can see, estimations from general imputation methods (LOESS, NMF, MC, wLRA, and pPCA) cannot satisfy the hierarchical consistent constraints and hence are not suitable for the missing value estimation under HTS settings. In contrast, our *HTSImpute* uses the summing matrix to get final estimations which always leads to hierarchically consistent results.

3) Efficiency: In order to examine the efficiency of our *HTSImpute* approach, we conduct evaluations on *EMEA*, *FP* and *Media* datasets in terms of both running time and number of iterations needed to achieve convergence.

Running Time Analysis The experiments are conducted on a Macbook Pro with 2.5 GHz Intel Core i7 and 16 GB 1600 MHz DDR3 under the same setting we used in previous section. We compute the average running time of each method and the results are shown in Table V. As we can see, *HTSImpute* runs faster than all other iterative methods (NMF, MC, pPCA, wLRA).

Table V. AVERAGE RUNNING TIME ON *EMEA*, *FP* AND *Media* DATASETS (SECONDS).

	LOESS	NMF_K	NMF_E	MC	pPCA	wLRA	HTSImpute
EMEA	0.13	146.75	152.98	4.72	15.43	4.56	0.45
FP	0.04	51.44	55.46	2.53	8.56	2.08	0.11
Media	0.09	49.77	57.93	5.95	10.39	4.89	0.27

Empirical Convergence Analysis In our *HTSImpute* algorithm, the algorithm ends by checking if the following condition holds true: $\|\mathbf{Y}^{(k+1)} - \mathbf{Y}^{(k)}\|_F / \|\mathbf{Y}^{(k)}\|_F \leq \delta$, where $\mathbf{Y}^{(k+1)}$ and $\mathbf{Y}^{(k)}$ are the estimated time series at *k+1*th and *k*th iterations and δ is the termination threshold. In the empirical convergence analysis experiments, we set δ equal to 10^{-6} and limit the maximum number of iterations under 1000. We conduct evaluations on *EMEA*, *FP* and *Media* datasets in terms of number of iterations needed to achieve convergence. We focus on comparisons between *wLRA* and *HTSImpute* to explore the effectiveness of hierarchical consistency projection. Results show that it takes *HTSImpute* 35.37, 20.6 and 46.13 iterations to get converged for *EMEA*, *FP* and *Media* datasets, while *wLRA* requires 1000, 813.77 and 930.23 iterations. As we can see, the hierarchical consistency projection plays an important role in achieving convergence and our *HTSImpute* converges much faster by using fewer iterations on all the real-world datasets.

V. CONCLUSION

In this paper, we presented an iterative algorithm, named *HTSImpute*, for HTS missing value estimation. Comparing with the traditional missing value estimation techniques, the advantages of it are (1) utilizing temporal dependence information within each individual time series; (2) exploiting intra-relations between different time series across the hierarchy; (3) guaranteeing hierarchical consistency. Experiment results on both synthetic and real-world datasets demonstrated that

Table I. AVG-MAPE RESULTS ON *syn_wide*/*syn_balance*/*syn_deep* DATASETS.

# Missing Percentage (%)	1	3	5	10	15	20
LOESS	4.83/4.64/4.98	5.20/4.99/4.53	5.49/5.11/4.62	5.73/5.20/4.62	5.97/5.23/4.63	6.21/5.35/4.67
NMF_K	4.36/3.98/4.62	4.83/4.37/4.10	5.15/4.57/4.20	5.47/4.64/4.21	5.73/4.77/4.25	6.01/4.91/4.34
NMF_E	4.64/4.53/5.19	5.03/4.83/4.60	5.34/4.95/4.72	5.65/5.05/4.65	5.94/5.06/4.67	6.23/5.20/4.70
MC	50.59/16.65/37.08	49.41/11.77/29.41	48.22/15.83/28.16	44.23/23.30/35.22	41.49/27.54/41.11	38.61/30.09/45.87
pPCA	102.42/100.57/97.92	99.79/101.91/98.96	100.33/99.59/98.45	101.19/101.10/100.41	101.13/100.17/99.18	100.33/100.78/99.47
wLRA	7.57/5.48/5.78	7.65/5.92/5.27	7.91/5.89/5.35	8.26/5.67/5.16	8.50/5.68/5.09	8.50/5.80/5.09
HTSImpute	**4.05/3.26/3.71**	**4.50/3.60/3.18**	**4.80/3.72/3.33**	**5.12/3.88/3.54**	**5.44/4.00/3.63**	**5.75/4.23/3.73**

Table II. AVG-MAPE RESULTS ON *EMEA*/*FP*/*Media* DATASETS.

# MP (%)	1	3	5	10	15	20
LOESS	18.28/11.57/30.72	18.15/11.84/30.49	18.48/11.66/57.78	18.75/11.78/76.61	18.82/11.82/64.75	18.89/11.78/76.40
NMF_K	15.82/15.12/29.33	15.92/14.11/28.90	16.52/13.83/56.06	17.00/14.48/76.79	17.16/14.74/64.46	17.31/14.61/77.32
NMF_E	14.61/21.19/30.75	14.20/17.75/29.93	14.36/16.49/55.10	15.28/17.41/77.26	15.45/15.70/62.45	15.49/15.97/72.74
MC	14.80/57.30/67.44	23.61/44.27/62.62	30.02/46.76/72.88	35.98/57.11/82.39	37.47/63.37/76.23	37.52/66.85/82.15
pPCA	100.56/101.51/100.18	100.25/100.17/99.84	99.93/100.21/101.74	100.00/100.03/101.90	100.08/100.01/102.69	100.12/100.19/103.86
wLRA	15.46/42.39/105.19	16.25/73.64/86.56	16.72/96.91/161.26	17.16/41.54/145.10	17.30/34.03/116.95	17.22/30.10/121.89
HTSImpute	**11.33/7.19/20.64**	**11.59/7.48/20.48**	**12.09/7.40/48.15**	**12.98/7.87/67.64**	**13.74/8.22/56.55**	**14.40/8.36/68.72**

Table III. AVG-HCG RESULTS ON *syn_wide*/*syn_balance*/*syn_deep* DATASETS.

# Missing Percentage (%)	1	3	5	10	15	20
LOESS	-2.67/-3.01/-3.14	-2.25/-2.58/-2.71	-2.04/-2.37/-2.49	-1.84/-2.12/-2.24	-1.73/-1.99/-2.11	-1.68/-1.91/-2.02
NMF_K	-2.31/-2.67/-3.02	-2.17/-2.61/-2.89	-2.00/-2.41/-2.64	-1.82/-2.26/-2.57	-1.72/-2.15/-2.39	-1.69/-2.07/-2.29
NMF_E	-2.24/-2.58/-2.55	-2.10/-2.52/-2.51	-2.01/-2.39/-2.60	-1.85/-2.24/-2.35	-1.73/-2.14/-2.25	-1.70/-2.07/-2.19
MC	-0.13/-3.20/-2.83	0.49/-2.36/-2.25	0.74/-1.63/-1.66	1.05/-1.00/-1.02	1.20/-0.64/-0.68	1.40/-0.41/-0.49
pPCA	0.36/0.03/0.38	0.80/0.72/0.02	1.06/0.47/0.44	1.38/0.77/0.87	1.51/0.97/0.93	1.70/1.11/1.10
wLRA	-5.68/-6.71/-6.65	-5.31/-6.40/-6.38	-5.04/-6.22/-6.24	-4.52/-5.93/-5.95	-3.87/-5.68/-5.75	-2.65/-5.36/-5.53
HTSImpute	**-16.45/-16.81/-15.87**	**-15.98/-16.37/-15.47**	**-15.32/-16.15/-15.27**	**-14.54/-15.90/-15.03**	**-14.41/-15.77/-14.92**	**-14.28/-15.69/-14.87**

Table IV. AVG-HCG RESULTS ON *EMEA*/*FP*/*Media* DATASETS ($\log_{10}$ SCALE).

# MP (%)	1	3	5	10	15	20
LOESS	-2.05/-2.61/-2.05	-1.61/-2.14/-1.61	-1.41/-1.93/-1.42	-1.19/-1.67/-1.19	-1.09/-1.54/-1.09	-1.04/-1.45/-1.01
NMF_K	-1.94/-2.52/-1.94	-1.59/-2.04/-1.63	-1.40/-1.95/-1.39	-1.19/-1.71/-1.18	-1.09/-1.60/-1.08	-1.04/-1.51/-1.02
NMF_E	-1.84/-2.01/-1.74	-1.54/-1.88/-1.50	-1.38/-1.80/-1.35	-1.18/-1.62/-1.16	-1.09/-1.55/-1.07	-1.04/-1.47/-1.01
MC	-2.27/-1.91/-1.19	-1.04/-1.35/-0.03	-0.55/-0.96/0.40	0.03/-0.19/0.27	0.34/0.30/0.43	0.56/1.34/0.86
pPCA	0.83/0.79/0.60	1.23/0.97/1.10	1.39/1.39/1.37	2.12/1.49/1.44	2.27/2.18/1.99	2.20/1.78/1.81
wLRA	-2.01/-5.65/-3.21	-1.57/-4.83/-2.30	-1.38/-1.58/-1.50	-1.16/-1.67/-0.69	-1.06/-1.55/-0.63	-1.01/-1.47/-0.71
HTSImpute	**-15.97/-16.66/-16.15**	**-15.53/-16.32/-15.70**	**-15.33/-16.04/-15.52**	**-15.11/-15.80/-15.29**	**-15.00/-15.71/-15.06**	**-14.96/-15.61/-14.90**

HTSImpute outperforms other state of the art approaches in terms of both accuracy (MAPE) and consistency (HCG). Going forward, we plan to extend this approach under probabilistic framework and evaluate *HTSImpute* under non-Gaussian assumptions. Further, we plan to derive more theoretical results regarding with the convergence study of *HTSImpute*.

ACKNOWLEDGMENT

We would like to thank Pengyuan Wang and James Li for useful discussions and comments on this work.

REFERENCES

[1] R. J. Hyndman, R. A. Ahmed, G. Athanasopoulos, and H. L. Shang, “Optimal combination forecasts for hierarchical time series,” *Computational Statistics & Data Analysis*, vol. 55, no. 9, pp. 2579–2589, 2011.

[2] H. Widiarta, S. Viswanathan, and R. Piplani, “Forecasting item-level demands: an analytical evaluation of top–down versus bottom–up forecasting in a production-planning framework,” *IMA Journal of Management Mathematics*, vol. 19, no. 2, pp. 207–218, 2008.

[3] I. Batal, D. Fradkin, J. Harrison, F. Moerchen, and M. Hauskrecht, “Mining recent temporal patterns for event detection in multivariate time series data,” in *KDD*. ACM, 2012, pp. 280–288.

[4] S. E. Said and D. A. Dickey, “Testing for unit roots in autoregressive-moving average models of unknown order,” *Biometrika*, vol. 71, no. 3, pp. 599–607, 1984.

[5] A. C. Harvey, *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.

[6] N. L. Crookston, A. O. Finley *et al.*, “yaimpute: An r package for knn imputation,” *Journal of Statistical Software*, vol. 23, no. 10, pp. 1–16, 2008.

[7] N. Srebro, T. Jaakkola *et al.*, “Weighted low-rank approximations,” in *ICML*, vol. 3, 2003, pp. 720–727.

[8] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *NIPS*, 2001, pp. 556–562.

[9] R. Mazumder, T. Hastie, and R. Tibshirani, “Spectral regularization algorithms for learning large incomplete matrices,” *JMLR*, vol. 11, pp. 2287–2322, 2010.

[10] M. E. Tipping and C. M. Bishop, “Probabilistic principal component analysis,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[11] W. S. Cleveland, “Robust locally weighted regression and smoothing scatterplots,” *Journal of the American Statistical Association*, vol. 74, no. 368, pp. 829–836, 1979.

[12] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, “Metagenes and molecular pattern discovery using matrix factorization,” *Proceedings of the National Academy of Sciences*, vol. 101, no. 12, pp. 4164–4169, 2004.

[13] Z. Liu and M. Hauskrecht, “A regularized linear dynamical system framework for multivariate time series analysis,” in *AAAI*, 2015.

[14] Z. Liu, Y. Yan, and J. Yang, “A probabilistic framework for hierarchical time series forecasting,” *Proceedings of the ICML Workshop on Demand Forecasting*, 2015.