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 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.

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

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
web page directly without accessing the parent page, which leads to the constraints 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.
Let \( h \) define an \( n \times 1 \) indicator vector where each element \( h_i \) indexes the parent of \( y_i \), and let \( W \) represent the missing value indicator matrix where \( w_{ij} = 1 \) indicates that \( y_{ij} \) is missing in \( Y \). We use \( Y, P, \) and \( L \) to denote values of \( Y, P, L \) after missing value imputation such that \( Y = [P; L] \).

In general, given a hierarchy, a parent equals to the sum of children. For example, in Figure 2, \( Y = [y_1, y_2, \ldots, y_7, y_8]^\top \) is an HTS with \( h = [0, 1, 1, 2, 2, 3, 3] \); \( y_1 \) is the root; \( y_1, y_2 \) and \( y_3 \) are parents; \( y_2, y_3, y_4, y_5, y_6, y_7 \) and \( y_8 \) are children; \( y_4, y_5, y_6, y_7 \) and \( y_8 \) are leaves. Here \( y_{i1} = y_{i2} + y_{i3}, y_{i2} = y_{i4} + y_{i5} + y_{i6}, \) and \( y_{i3} = y_{i7} + y_{i8} \) where \( i = 1, \ldots, T \). In terms of matrix operations we use \( \odot \) to denote element-wise matrix multiplication.

![Figure 2. An HTS notation illustration.](image)

### B. Modeling Temporal Dependences

The first stage of \textit{HTSImpute} algorithm aims to estimate the missing values by considering temporal dependences among time series values. Simply putting zeros or 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 \textit{HTSImpute}, we choose a nonparametric robust locally weighted regression model, \textit{LOcal regESSion} (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 \textit{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 \( Y \). 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.

---

**Theorem 1.** The HTS matrix \( Y \) has rank \( r \leq \min(l, T) \).

The finding expressed in Theorem 1 lets us project observational time series matrix \( Y \) 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 \textit{HTSImpute}, we project \( Y^{(k)} \) to a \( r \)-rank matrix through a projection operator \( P_{r-SVD} \), defined as follows:

\[
Y^{(k+1)} = P_{r-SVD}(W \odot Y + (I - W) \odot Y^{(k)})
\]

where \( P_{r-SVD} \) is the unweighted rank-\( r \) approximation of \( X \), computed from the Singular Value Decomposition(SVD) as follows: \( P_{r-SVD}(X) = U \Lambda V^\top \), where \( X = UAV^\top \) and \( \Lambda_r = \text{diag}([\lambda_1, \ldots, \lambda_r, 0, \ldots, 0]) \).

The subspace projection operator \( P_{r-SVD} \) 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 \( W \) is treated as a zero/one weighting matrix [7].

#### 2) Hierarchical Consistency Projection:

For each column \( y_{i,t} \) in \( Y \), 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 \( Y \) are corrupted by Gaussian noise \( \epsilon_Y \) 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 \( L \) (\( L = L + \epsilon_L \)) [1]. Then \( Y \) can be expressed concisely as

\[
Y = \Omega \cdot L + \epsilon_Y = \Omega \cdot (L + \epsilon_L)
\]

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

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

\[
P_{HTS}(Y, \Omega) = \Omega (\Omega^\top \Omega)^{-1} \Omega^\top Y
\]

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

### D. \textit{HTSImpute} Summary

The \textit{HTSImpute} method is summarized in Algorithm 1.

#### Special Cases:

According to Theorem 1, the rank of \( Y \) 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 \( l > T \), \( Y \) becomes a full rank matrix and the subspace projection step can be skipped.

**Complexity Analysis:** In every iteration, HTSImpute performs two projections: subspace projection \((P_{K-SVD}()\) and hierarchical consistency projection \((P_{HTS}())\). The former requires \( O(nT \cdot \min(n, T)) \) to have rank \( r \) matrix approximation and the latter needs \( O(l^3) \) to conduct the hierarchical adjustments. The total running time complexity per iteration is \( O(nT \cdot \min(n, T) + l^3) \). 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 \( O(n^2T) \) per-iteration.

**Algorithm 1 HTSImpute:** Hierarchical Time Series Imputation.

**INPUT:**
- A \( n \times T \) partially observed HTS matrix \( Y = [P; L] \)
- A \( n \times T \) observation indicator matrix \( W \)
- A \( n \times l \) summing matrix \( \Omega \)

**PROCEDURE:**
1. Initialize the missing value through LOESS
2. \( Y^{(0)} = \text{Loess}(Y, W) \)
3. repeat
4. Subspace projection
5. \( Y^{(k)} = P_{K-SVD} (\mathbf{W} \odot Y + (1 - \mathbf{W}) \odot Y^{(k)}) \)
6. Hierarchical consistency projection
7. \( Y^{(k)} = P_{HTS}(Y^{(k)}, \Omega) \)
8. until convergence

**OUTPUT:**
- \( \hat{Y} \) 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 estimation. 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:

\[
\text{Avg-MAPE} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{T} (1 - w_{ij}) \cdot |1 - \hat{y}_{ij}/y_{ij}|}{\sum_{i=1}^{n} \sum_{j=1}^{T} (1 - w_{ij})} \times 100\%
\]
\[
\text{Avg-HCG} = \frac{1}{pT} \sum_{i=1}^{p} \sum_{j=1}^{T} |1 - \sum_{k=1}^{n} \mathbb{I}(y_{ij} = k) \cdot \hat{y}_{ij}/\hat{p}_{ij}|\]

where \( \hat{y}_{ij} \) and \( \hat{p}_{ij} \) are the \( ij \)th elements from \( \hat{Y} \) and \( \hat{P} \).

**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.

**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 represents 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 ($M_8$), 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.

![HTS structure for EMEA, FP, Media datasets.](image)

**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).**

<table>
<thead>
<tr>
<th>Method</th>
<th>EMEA</th>
<th>FP</th>
<th>Media</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOESS</td>
<td>0.11</td>
<td>0.04</td>
<td>0.09</td>
</tr>
<tr>
<td>NMF_K</td>
<td>14.75</td>
<td>51.44</td>
<td>49.77</td>
</tr>
<tr>
<td>NMF_E</td>
<td>152.98</td>
<td>55.46</td>
<td>57.93</td>
</tr>
<tr>
<td>MC</td>
<td>4.72</td>
<td>2.53</td>
<td>5.95</td>
</tr>
<tr>
<td>pPCA</td>
<td>15.43</td>
<td>8.56</td>
<td>10.39</td>
</tr>
<tr>
<td>wLRA</td>
<td>4.56</td>
<td>2.08</td>
<td>4.89</td>
</tr>
<tr>
<td>HTSImpute</td>
<td>0.45</td>
<td>0.11</td>
<td>0.27</td>
</tr>
</tbody>
</table>

**Empirical Convergence Analysis**

In our HTSImpute algorithm, the algorithm ends by checking if the following condition holds true: $\| Y^{(k+1)} - Y^{(k)} \|_F / \| Y^{(k)} \|_F \leq \delta$, where $Y^{(k+1)}$ and $Y^{(k)}$ are the estimated time series at $k+1$th and $k$th iterations and $\delta$ is the termination threshold. In the empirical convergence analysis experiments, we set $\delta$ 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
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


