On the Use of Dictionary Learning in Time Series Imputation

Xiaomeng Zheng
Department of Statistics
University of Auckland
Auckland, New Zealand
xzhe229@aucklanduni.ac.nz

Bogdan Dumitrescu
Department of Automatic Control and Computers
University Politehnica of Bucharest
Bucharest, Romania
bogdan.dumitrescu@upb.ro

Jiamou Liu
School of Computer Science
University of Auckland
Auckland, New Zealand
jiamou.liu@auckland.ac.nz

Ciprian Doru Giurcăneanu
Department of Statistics
University of Auckland
Auckland, New Zealand
c.giurcaneanu@auckland.ac.nz

***Abstract*—In this work, we show how dictionary learning (DL) can be employed in the imputation of univariate and multivariate time series. In the multivariate case, we propose to use a structured dictionary. The size of the dictionary and the sparsity level are selected by information theoretic criteria. We also evaluate the effect of removing the trend/seasonality before applying DL. We conduct an extensive experimental study on real-life data. The positions of the missing data are simulated by applying two strategies: (i) sampling without replacement, which leads to isolated occurrences of the missing data, and (ii) sampling via Polya urn model that is likely to produce long sequences of missing data. In all scenarios, the novel DL-based methods compare favorably with the state-of-the-art.**

***Index Terms*—Time series, missing data, dictionary learning, Polya urn, information theoretic criteria**

I. Introduction

Problem formulation: Assume that the data matrix $\boldsymbol{Z} \in \mathbb{R}^{T\times K}$ contains $K > 1$ time series that are observed at time points $1, \ldots, T$. Suppose that some of the data are missing. As the positions of the missing data are not necessarily the same for all the time series, we use the symbol Ψ_k to denote the indexes of the measurements that are available for the k-th time series, where $1 \leq k \leq K$. The set of indexes of missing data for the k-th time series is $\overline{\Psi}_k = \{1, \ldots, T\} \setminus \Psi_k$.

We propose to apply dictionary learning (DL) for the imputation of the missing values. Before presenting our approach, here are some details on the notation that we employ.

Notation: Bold letters are used for both vectors and matrices; $\mathbf{0}$ is the vector/matrix whose entries are all equal to zero. For an arbitrary vector $\boldsymbol{v}$, $\boldsymbol{v}_\Psi$ denotes the entries of the vector whose indexes belong to the set Ψ. When Ψ is given by the set of all positive integers from a to b ($a < b$), we prefer to write $\boldsymbol{v}_{a:b}$ instead of $\boldsymbol{v}_\Psi$. The symbol v_a is used for the a-th entry of $\boldsymbol{v}$. We use $[\boldsymbol{v}_1; \ldots; \boldsymbol{v}_d]$ to denote the column vector constructed by stacking d column vectors. If $\boldsymbol{A}$ is a matrix, then $\boldsymbol{A}_{\Psi:}$ defines the block for which the indexes of the rows belong to Ψ. The operator for transposition is $(\cdot)^\top$. The symbols $||\cdot||_2$ and $||\cdot||_F$ stand for the Euclidean norm and the Frobenius norm, respectively. For a real number x, the symbol $\lfloor x \rfloor$ denotes the greatest integer not larger than x.

DL imputation method: Let $\boldsymbol{z}$ be one of the columns of the data matrix $\boldsymbol{Z}$ in which the missing data indexed by $\overline{\Psi}$ are replaced with zeros. We define a matrix $\boldsymbol{Y}$ as follows: $\boldsymbol{Y} = [\boldsymbol{z}_{1:m}\ \boldsymbol{z}_{1+h:m+h} \cdots \boldsymbol{z}_{1+qh:m+qh}]$. The number of rows, m, depends on the sampling period; for example, for hourly data, a good choice is $m = 24$. The parameter h is called signal shift and controls the overlapping between the columns of $\boldsymbol{Y}$; the value of q is computed as $\lfloor (T-m)/h \rfloor$. The number of columns of the matrix $\boldsymbol{Y}$ is $N = q + 1$. We take $h = 1$.

We apply DL for approximating $\boldsymbol{Y} \in \mathbb{R}^{m\times N}$ by the product $\boldsymbol{DX}$, where $\boldsymbol{D} \in \mathbb{R}^{m\times n}$ is the *dictionary* and its columns are usually named *atoms*. The Euclidean norm of each atom equals one. The matrix $\boldsymbol{X} \in \mathbb{R}^{n\times N}$ is sparse in the sense that each of its columns contains at most s non-zero entries, the parameter s being named sparsity level. We emphasize that both $\boldsymbol{D}$ and $\boldsymbol{X}$ are learned from the data.

If $t \in \overline{\Psi}$, then z_t is a missing value, and this will be represented as a zero-entry in $\boldsymbol{Y}$. As there is overlap between the columns of $\boldsymbol{Y}$, it is likely that the missing value z_t leads to several zero-entries in $\boldsymbol{Y}$. We collect all the values of these entries from $\hat{\boldsymbol{Y}} = \boldsymbol{DX}$ and compute an estimate for z_t by averaging them. The imputation method is inspired from the applications of DL to inpainting.

Previous work: Pioneered in [1], the use of DL in image inpainting has seen several structural improvements, like the use of multiscale dictionaries [2] or trainlets DL [3], and adaptation to specific types of images, like those from computer tomography [4]. DL-based inpainting has also been applied to audio signals [5].

In what concerns time series imputation, various methods have been proposed in the previous literature. Because of the limited space, we briefly describe only those implemented in the `R`-packages `imputeTS` [6] and `MTSDI` [7], which are used in our experiments. The first package is dedicated to imputation of *univariate* time series and comprises sev-

eral methods like, for example, interpolation and Kalman smoothing (see [8] for more details). MTSDI is specialized in *multivariate* time series and relies on a variant of the Expectation-Maximization (EM) algorithm [9] for estimating the mean vector and the covariance matrix of a vector-valued random variable that is Gaussian distributed. During the iterations performed for the EM algorithm, all time series from the dataset are used. In the next stage, the estimates of the missing values are further improved by filtering each time series independently. To this end, cubic splines, autoregressive integrated moving average (ARIMA) models or regression models are employed [10].

Main contributions and the organization of the paper: We propose a DL algorithm using a structured dictionary for modeling multivariate time series with missing data, described in Sec. II. We investigate the effect of transforming the data before the DL estimation. The transformations that we consider are: (i) removing the deterministic trend and (ii) differencing the time series. We explain in Sec. III how this task can be performed when some of the data are missing. The DL-based imputation method for univariate time series is empirically evaluated in Sec. IV. The capabilities of the variant for multivariate time series are assessed in Sec. V. Sec. VI concludes the paper.

II. More details on the DL algorithm

We denote $\boldsymbol{M}$ a mask matrix with the same size as $\boldsymbol{Y}$ and having elements equal to zero where the elements of $\boldsymbol{Y}$ are zero (because they are missing) and otherwise equal to one. The DL goal is to design a dictionary $\boldsymbol{D}$ that minimizes $\|\boldsymbol{M}\odot(\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X})\|_F$, where the representation matrix $\boldsymbol{X}$ has at most s nonzeros on each column and $\odot$ denotes the element-wise matrix product. This is a DL problem with incomplete data that can be solved with the typical approach that alternates two steps: representation (compute $\boldsymbol{X}$, given $\boldsymbol{D}$) and atom update (optimize $\boldsymbol{D}$, given $\boldsymbol{X}$). We use a specialized version of Aproximate K-SVD (AK-SVD), described in [11, Sec. 5.9]; representations are found using Orthogonal Matching Pursuit (OMP) by simply ignoring the missing samples and working only with the present ones; the atom update formulas are simple adaptations of AK-SVD rules to the incomplete data case. This adapted AK-SVD is used for the imputation of a univariate time series, the method being called DLU in Sec. IV.

In the multivariate case, there are two extreme solutions: to train a dictionary for each univariate series, or a single dictionary for all series. The first one is adequate if the series are weakly correlated and are sufficiently long to produce enough training vectors; it is more time consuming. The second one is good especially when the series are well correlated and short. We propose a more flexible alternative. Let $\boldsymbol{Y}=[\boldsymbol{Y}_1\ \ldots\ \boldsymbol{Y}_K]$, where $\boldsymbol{Y}_i$ is made of data from series i. The dictionary has the structure $\boldsymbol{D}=[\boldsymbol{D}_1\ \ldots\ \boldsymbol{D}_K\ \boldsymbol{D}_{K+1}]$; the dictionary $\boldsymbol{D}_i$ is dedicated to the representation of series i, while $\boldsymbol{D}_{K+1}$ is common for all series. Only the representation step of the AK-SVD needs modifications: OMP uses only atoms from the dictionaries $\boldsymbol{D}_i$ and $\boldsymbol{D}_{K+1}$ for the representations of the vectors from $\boldsymbol{Y}_i$. The atom update step needs no change, as atoms are optimized using the representations of vectors to which they participate, no matter to what series the vectors belong. This DL algorithm for multivariate time series will be used for imputation in the method described in Sec. V and called DLM.

III. Data pre-processing

Removing the deterministic trend: For each column $\boldsymbol{z}$ of the data matrix $\boldsymbol{Z}$, we focus primarily on removing the linear and the periodic components. Note that, for each univariate time series, the linear trend is always removed. Our implementation also allows to remove p periodic components with frequencies $\omega_1,\ldots,\omega_p$. The number of the periodic components as well as their frequencies are provided by the user. For example, they can be selected from the highest peaks of the periodogram.

For ease of writing, we define the matrix $\boldsymbol{H}$ which has the following entries

$$\begin{bmatrix} 1 & 1 & \cos(\omega_1) & \sin(\omega_1) & \ldots & \cos(\omega_p) & \sin(\omega_p) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & T & \cos(T\omega_1) & \sin(T\omega_1) & \ldots & \cos(T\omega_p) & \sin(T\omega_p) \end{bmatrix}$$

The linear coefficients of the model can be easily obtained from the existing data $\boldsymbol{z}_\Psi$ by applying the least squares (LS) estimation method: $\hat{\boldsymbol{\beta}}=(\boldsymbol{H}_{\Psi:}^\top\boldsymbol{H}_{\Psi:})^{-1}\boldsymbol{H}_{\Psi:}^\top\boldsymbol{z}_\Psi$. Hence, the transformed time series is the vector $\tilde{\boldsymbol{z}}$ with the property that $\tilde{\boldsymbol{z}}_\Psi=\boldsymbol{z}_\Psi-\boldsymbol{H}_{\Psi:}\hat{\boldsymbol{\beta}}$. All the entries of $\tilde{\boldsymbol{z}}$ that correspond to indexes from $\overline{\Psi}$ are equal to zero. At the same time, the missing values can be imputed as follows: $\hat{\boldsymbol{z}}_{\overline{\Psi}}=\boldsymbol{H}_{\overline{\Psi}:}\hat{\boldsymbol{\beta}}$.

Differencing the time series: Assuming that $\boldsymbol{z}$ has the same significance as above, for a given positive integer θ, we obtain the transformed time series $\{\tilde{z}_t\}_{1\le t\le T}$ by computing the differences $z_t-z_{t-\theta}$ for all t. There are two main difficulties in calculating the differences. The first one occurs when $t\le\theta$ and the second one is related to the case when the measurement at time moment $t-\theta$ is missing. In order to present the solution that we propose for circumventing the difficulties, we introduce the auxiliary vector $\boldsymbol{v}=[\boldsymbol{0};\ \boldsymbol{z}_{1:\ell-\theta};\ \boldsymbol{z}_{\ell-\theta+1:\ell};\ \boldsymbol{z}_{1:T-\theta}]$, where $\ell=\lfloor T/\theta\rfloor\theta$ and $\boldsymbol{0}$ denotes a column vector of length $T-\ell+\theta$. Note that $\boldsymbol{v}$ is defined by taking into consideration the seasonality of the time series. The length of $\boldsymbol{v}$ is $2T$.

The transformed time series, $\{\tilde{z}_t\}_{1\le t\le T}$, is obtained by applying the formula:

$$\tilde{z}_t=\begin{cases} z_t-g(\boldsymbol{u}^{[t]}), & t\in\{1,\ldots,\theta\}\cap\Psi \\ z_t-g(\boldsymbol{w}^{[t]}), & t\in\{\theta+1,\ldots,T\}\cap\Psi \\ 0, & t\in\overline{\Psi}\end{cases},$$

where the function $g(\cdot)$ returns the first non-zero entry of the vector in the argument. If the vector in the argument contains only zeros, then $g(\cdot)$ returns zero. The vector $\boldsymbol{u}^{[t]}$ has the expression:

$$\boldsymbol{u}^{[t]}=[v_{T+t-0};v_{T+t-0-\theta};\ldots;v_{T+t-0-q_0\theta};\\ \ldots;v_{T+t-(\theta-1)-0};v_{T+t-(\theta-1)-\theta};\ldots;v_{T+t-(\theta-1)-q_{\theta-1}\theta}],$$

where $q_{\text{ind}} = \lfloor (t - \text{ind} + \ell - 1)/\theta \rfloor - 2$, for $\text{ind} \in \{0, \ldots, \theta - 1\}$. The expression of $\boldsymbol{w}^{[t]}$ is the same as the one for $\boldsymbol{u}^{[t]}$, except that the formula for q_{ind} is $\lfloor (t - \text{ind} - 1)/\theta \rfloor - 1$, where $\text{ind} \in \{0, \ldots, \theta - 1\}$.

The seasonality of the time series $\boldsymbol{z}$ can be also used for providing an imputation method. With the notation introduced above, we have the following estimate for the missing value z_t:

$$\hat{z}_t = \begin{cases} g(\boldsymbol{u}^{[t]}), & t \in \{1, \ldots, \theta\} \cap \overline{\Psi} \\ g(\boldsymbol{w}^{[t]}), & t \in \{\theta + 1, \ldots, T\} \cap \overline{\Psi} \end{cases}.$$

IV. DL imputation method for univariate time series (DLU) - Experimental evaluation

Dataset: We consider $K = 20$ of the hourly sampled time series that have been recently used in the M4 forecasting competition [12]. For the competition, each time series was divided into two sub-series: one for training and another one for testing. As we are not interested in forecasting, we have combined the sub-series such that the length of each time series in the resulting dataset is $T = 1008$. We mention that all the time series are complete and we simulate the missing data as it is described in the next paragraph.

Missing data: For fairness, we assume that the number of missing data, M_{miss}, is the same for all time series. This implies that the percentage of missing data is $\rho = 100(M_{\text{miss}}/T)$ for each time series. The indexes of the missing data for a particular time series are independent of the positions of the missing data in the other time series from the same dataset. They are selected by either sampling without replacement M_{miss} integers from the set $\{1, \ldots, T\}$ or by using the Polya urn model (with finite memory) [13].

In Polya model, the urn initially contains R red balls and S black balls ($R < S$). At each time moment $t \geq 1$, a ball is drawn from the urn and, after each draw, $(1 + \Delta)$ balls of the same color as the drawn ball are returned to the urn. We take $\Delta > 0$. Since we want the model to have finite memory, the experiment is performed as described above only for $1 \leq t \leq M$, where the parameter M is a positive integer (see the discussion in [13]). At each time moment $t > M$, a ball is drawn from the urn and, after each draw, two operations are executed: (i) $(1 + \Delta)$ balls of the same color as the drawn ball are returned to the urn and (ii) Δ balls of the same color as the ball picked at time $t - M$ are removed from the urn. A sequence of random variables $\{\Xi_t\}_{1 \leq t \leq T}$ is defined as follows: $\Xi_t = 1$ if the ball drawn at time t is red and $\Xi_t = 0$ if the ball drawn at time t is black.

The indexes of the missing data correspond to the positions of ones in the sequence $\{\Xi_t\}_{1 \leq t \leq T}$. It is known that $\text{Prob}(\Xi_t = 1) = R/(R + S)$, hence in our simulations R and S are chosen such that $\text{Prob}(\Xi_t = 1) = M_{\text{miss}}/T$. More interestingly, we have that the correlation $\text{Corr}(\Xi_t, \Xi_{t-i}) = \delta/(1 + \delta)$, where $0 < i < M$ and $\delta = \Delta/(R + S)$ [14]. This property allows us to simulate bursts of missing data by choosing relatively large values for δ. Obviously, this is different from the situation when the sampling without replacement is applied and when is more likely to have isolated missing data. We mention that $M = 5$ and $\delta \in \{0.25, 0.5, 1\}$ in our settings. Let $\rho = 5\%$; for each value of δ, we write in parentheses the average length, followed by the maximum length of the sequences of missing data. The statistics collected from experiments with $K = 20$ time series are: $\delta = 0.25(1.28; 9)$, $\delta = 0.5(1.61; 16)$ and $\delta = 1(1.99; 20)$. For the same value of ρ, when sampling without replacement, the average length of the sequences is 1.03 and the maximum length is 3.

Information theoretic (IT) criteria for DLU: We have tested six different criteria for choosing the dictionary size n and the sparsity s, but we only give the formulas for two of them that produced the best results in our experiments. Both of them are based on the celebrated Bayesian Information Criterion (BIC) [15]. More precisely, they are obtained from the selection rule in [16] that was used in DL when complete data were available.

Let η be the total number of ones in the mask matrix $\boldsymbol{M}$ defined in Sec. II. For computing the goodness-of-fit term, first we calculate the root mean square error: $\text{RMSE} = ||\boldsymbol{M} \odot (\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X})||_F/\sqrt{\eta}$. The complexity of the model is mainly given by the number of parameters: $\text{NoP} = sN + (m - 1)n$. The expression of the first criterion is: $\text{BIC} = 2 \log \text{RMSE} + \frac{\log \eta}{\eta} \text{NoP}$, where the symbol $\log(\cdot)$ is used for the natural logarithm. The second criterion is an "extended" version (see [16], [17]) of the first one: $\text{EBIC} = \text{BIC} + \frac{2N}{\eta} \log \binom{n}{s}$.

Experimental settings: Whenever DLU is applied to a univariate time series $\boldsymbol{z}$ with missing values, ten random initializations of the dictionary are considered. For each initialization, fifty iterations of the DL algorithm are executed for each pair (n, s). As our data was sampled hourly, it is natural to have $m = 24$. Then the possible values of the dictionary size n are taken to be multiple integers of m: $2m, 3m, \ldots, 8m$. The possible values of the sparsity s are: 2, 3, 4 and 6. Hence, for each time series a number of $10 \cdot 7 \cdot 4 = 280$ different imputations are produced. With the convention that, for the time series $\boldsymbol{z}$, $\check{\boldsymbol{z}}_{\overline{\Psi}}$ is the vector of true values for the missing data and $\hat{\boldsymbol{z}}_{\overline{\Psi}}$ is the vector of estimates for the missing data, an Oracle decides that the best imputation is the one which minimizes $||\check{\boldsymbol{z}}_{\overline{\Psi}} - \hat{\boldsymbol{z}}_{\overline{\Psi}}||_2$. This method is called DLU(Oracle) because it assumes knowledge of the ground truth.

A more advanced option is to transform the time series before applying DLU. The transformations are those presented in Sec. III, and a detailed list of the transformations applied to each time series can be found in the supplementary material [18, Tables I:XL]. DLU is applied to the transformed data as explained above, and the imputation method is dubbed Trend+DLU(Oracle). Practical imputation methods can be obtained by replacing Oracle either with BIC or with EBIC; they are called Trend+DLU(BIC) and Trend+DLU(EBIC), respectively. The crude imputation method introduced in Sec. III, which relies only on the trend/seasonality of the time series, is named Trend.

For comparison, we also consider the imputation method from [6]. We call it imputeTS when it is applied to the original time series and Trend+imputeTS when it is applied to the transformed time series. We employ only the transformations presented in Sec. III, and we do not use the transformations implemented in the R-package. Because we do not want to fit models for which the user should provide the structural parameters, we chose the default option that performs imputation via interpolation.

Another method considered in our experiments is the one from [7]. As it is designed for multivariate time series, we apply it to all $K = 20$ time series simultaneously. We call it either MTSDI or Trend+MTSDI depending if it is used for the original data or for the transformed data. For reasons that have been already mentioned in connection with imputeTS-method, we select the option that does the filtering by using cubic splines.

Results: For all time series, we show in the supplementary material [18, Tables I-XL] the values of the normalized errors $\frac{||\check{\boldsymbol{z}}_{\overline{\Psi}} - \hat{\boldsymbol{z}}_{\overline{\Psi}}||_2}{||\check{\boldsymbol{z}}_{\overline{\Psi}}||_2}$, which are computed for each imputation method. In order to rank the methods, we calculate scores as follows. For each time series, the imputation method that yields the minimum normalized errors gets one point, the method that leads to the second smallest normalized errors gets one half of a point, and all other methods get zero points. The number of points accumulated by each method from the experiments with all time series are divided by K such that to make it sure that the scores have values in the interval $[0, 1]$.

For the case when missing data are simulated by using sampling without replacement, the scores are calculated based on [18, Tables I-XX] and they are presented in [18, Fig. 1]. DLU(Oracle) is the best amongst all the methods applied to the original time series. At the same time, Trend+DLU(Oracle) is much better than any other method. This demonstrates that DLU has a great potential when it is applied to the transformed data. However, the only possibility to turn it into a practical method is to use an IT criterion. It is worth mentioning that the normalized errors computed for the second best method are always close to those produced by Trend+DLU(Oracle) (see [18, Fig. 2]).

We eliminate from the competition the methods that involve Oracle and recompute the scores for the remaining methods. The methodology for calculating the scores is the same as above and the results are exhibited in Fig. 1. Note that, for all values of ρ, Trend+DLU(BIC) and Trend+DLU(EBIC) are superior to Trend+imputeTS.

The scores earned when the Polya urn model is used in simulations are displayed in [18, Fig. 3] (see also [18, Tables XXI-XL]). Additionally, we compute the scores for the subset of the practical methods that do not rely on Oracle and show them in Fig. 2. The similarity between the ranking of the imputation methods in Fig. 1 and Fig. 2 is evident.

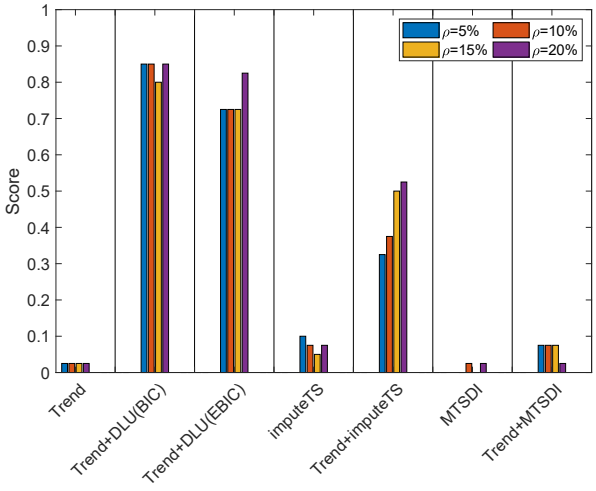

Fig. 1: Scores for the imputation methods which do not involve Oracle. The missing data are generated by sampling without replacement. The percentages of the missing data are given in the legend.

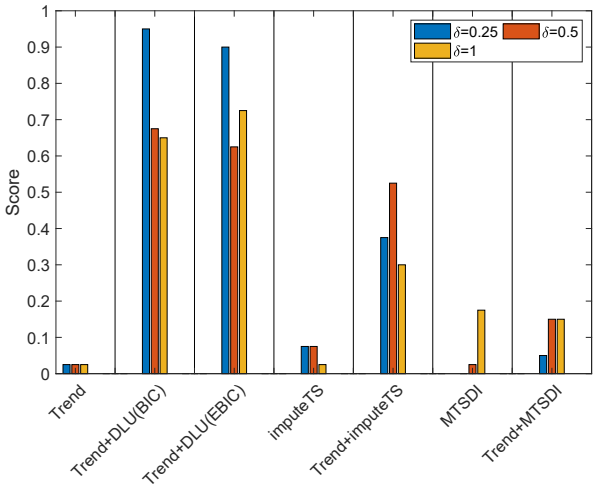

Fig. 2: Results similar to those reported in Fig. 1. The main difference is that the positions of the missing data are selected by applying the Polya urn model (with parameter $M = 5$). The values of the parameter δ for the model are given in the legend. The percentage of the missing data is $\rho = 5\%$.

V. DL IMPUTATION METHOD FOR MULTIVARIATE TIME SERIES (DLM) - EXPERIMENTAL EVALUATION

Dataset: In this experiment, we consider $K = 3$ time series that are deemed to measure a multivariate process. They represent the monthly number of tourists in three states of Australia: New South Wales (NSW), Victoria (VIC) and South Australia (SA). The data are available at the address https://robjhyndman.com/data/TourismData_v3.csv and comprise measurements from January 1998 to December 2016. It follows that the number of measurements for each time series

is $T = 228$. Note that the value of T is much smaller than in the case of the time series that have been analyzed in Sec. IV.

Missing data: The sampling without replacement is used in order to select the indexes of the missing data that represent $\rho = 5\%$ of T.

IT criteria for DLM: We employ again BIC and EBIC, but there is a major difference in comparison with using them for DLU. In DLU, the IT criteria are instrumental in finding the size n and the sparsity s of the dictionary $\boldsymbol{D}$, as well as in selecting the dictionary initialization that have produced the "best" estimates. In DLM, the IT criteria should perform the additional task of choosing the sizes $n_1, \ldots, n_K, n_{K+1}$ of the blocks $\boldsymbol{D}_1, \ldots, \boldsymbol{D}_K, \boldsymbol{D}_{K+1}$ of $\boldsymbol{D}$. For ease of computation, we assume that $n_1 = n_2 = \cdots = n_K$ and they are equal to n_d. Furthermore, we define $n_u = n_d + n_{K+1}$. It is clear that, for each time series in the dataset, n_u represents the number of atoms used in its representation. In analogy with the DLU case, n_u is taken to be one of the entries of the set $\{2m, \ldots, 8m\}$. For each value of n_u, we have that $n_{K+1} \in \{m, 2m, \ldots, n_u - m\}$.

Experimental settings: We only mention the settings that are different from those used in Sec. IV. For instance, we take $m = 12$ because we analyze monthly data. As m is smaller than in the DLU case, we constrain the sparsity s to be selected from the set $\{2, 3, 4\}$.

Additional to DLM, the imputation methods presented in the previous section are also evaluated in our experiment. Using the same convention as for Trend+DLU, we dub Trend+DLM the method that applies DLM to the transformed data. The transformations operated to the time series NSW, VIC and SA are explained in [18, Sec. III]. Because of the multivariate framework, Trend+DLM(Oracle) selects the triple (n_u, n_{K+1}, s) that minimizes the average of the normalized errors for $K = 3$ time series. This explains why, according to the results reported in [18, Table XLI], DLM(BIC) and DLM(EBIC) are superior to DLM(Oracle) for NSW. The average of the normalized errors for DLM(Oracle) is indeed smaller than the averages computed for DLM(BIC) and DLM(EBIC).

Results: As we are interested only on those methods which are practical, we exclude from competition the approaches that are based on Oracle. In the class of the practical methods, the smallest average is achieved by Trend+DLM(BIC) and Trend+DLM(EBIC), which are closely followed by MTSDI. This is a consequence of the fact that Trend+DLM(BIC) and Trend+DLM(EBIC) are ranked the best for VIC and second best for NSW and SA. Remarkably, MTSDI is ranked at the first position for the time series SA. It is not surprising that, in the family of the DL methods, DLM is better than DLU when the data are multivariate. It is also interesting that both BIC and EBIC select models with high sparsity for DLM; in general, $s = 2$ according to [18, Table XLI]. The value selected for n_u is either $2m$ or $3m$, even if in the set of possible sizes there are values as large as $8m$. In most of the cases, the value of n_u chosen for DLM is the same as the dictionary size n chosen for DLU.

VI. Final remarks

Based on the empirical results, we can draw the following conclusions: (i) DL can be successfully used in time series imputation, and its performance compares favorably with the the existing methods; (ii) It is recommended to remove the trend/seasonality before applying DL, especially in the case of the univariate time series; (iii) BIC works well for selecting the dictionary size n and the sparsity s for DLU. In the case of DLM, BIC is also recommended for deciding the size of the common block $\boldsymbol{D}_{K+1}$ and the size of the dictionary blocks $\boldsymbol{D}_1, \ldots, \boldsymbol{D}_K$, which are specific for each time series in the dataset.

The experimental results can be reproduced by using the Matlab code available at the address https://www.stat.auckland.ac.nz/%7Ecgiu216/PUBLICATIONS.htm.

References

[1] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

[2] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *SIAM Multiscale Modeling Simulation*, vol. 7, no. 1, pp. 214–241, 2008.

[3] J. Sulam and M. Elad, "Large inpainting of face images with trainlets," *IEEE Signal Processing Letters*, vol. 23, no. 12, pp. 1839–1843, 2016.

[4] S. Li, Q. Cao, Y. Chen, Y. Hu, L. Luo, and C. Toumoulin, "Dictionary learning based sinogram inpainting for CT sparse reconstruction," *Optik*, vol. 125, no. 12, pp. 2862–2867, 2014.

[5] M.G. Jafari and M.D. Plumbley, "Fast dictionary learning for sparse representations of speech signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 1025–1031, 2011.

[6] S. Moritz and S. Gatscha, "Package: imputeTS, Version 3.0," https://CRAN.R-project.org/package=imputeTS.

[7] W.L. Junger and A. Ponce de Leon, "Package: mtsdi, Version: 0.3.5," https://CRAN.R-project.org/package=mtsdi.

[8] S. Moritz and T. Bartz-Beielstein, "imputeTS: Time Series Missing Value Imputation in R," *The R Journal*, vol. 9, no. 1, pp. 207–218, 2017.

[9] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[10] W.L. Junger and A. Ponce de Leon, "Imputation of missing data in time series for air pollutants," *Atmospheric Environment*, vol. 102, pp. 96–104, 2015.

[11] B. Dumitrescu and P. Irofti, *Dictionary Learning Algorithms and Applications*, Springer, 2018.

[12] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020.

[13] F. Alajaji and T. Fuja, "A communication channel modeled on contagion," *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 2035–2041, 1994.

[14] R. Iordache, I. Tabus, and J. Astola, "Robust index assignment using Hadamard transform for vector quantization transmission over finite-memory contagion channels," *Circuits, Systems and Signal Processing*, vol. 21, no. 5, pp. 485–509, 2002.

[15] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.

[16] B. Dumitrescu and C.D. Giurcăneanu, "Adaptive-size dictionary learning using information theoretic criteria," *Algorithms*, vol. 12, no. 9, 2019, 13 pages.

[17] J. Chen and Z. Chen, "Extended Bayesian information criteria for model selection with large model spaces," *Biometrika*, vol. 95, no. 9, pp. 759–771, 2008.

[18] X. Zheng, B. Dumitrescu, J. Liu, and C.D. Giurcăneanu, "Supplementary material to: On the use of dictionary learning in time series imputation," https://www.stat.auckland.ac.nz/%7Ecgiu216/PUBLICATIONS.htm, 2020.