Learning Periods from Incomplete Multivariate Time Series

Lin Zhang
Computer Science
University at Albany—SUNY
Albany, U.S.A
lzhang22@albany.edu

Alexander Gorovits
Computer Science
University at Albany—SUNY
Albany, U.S.A
agorovits@albany.edu

Wenyu Zhang
Statistics and Data Science
Cornell University
Ithaca, U.S.A
wz258@cornell.edu

Petko Bogdanov
Computer Science
University at Albany—SUNY
Albany, U.S.A
pbogdanov@albany.edu

Abstract—Time series often exhibit seasonal variations whose modeling is essential for accurate analysis, prediction and anomaly detection. For example, the increase in consumer product sales during the holiday season recurs yearly, and similarly household electricity usage has daily, weekly and yearly cycles. The period in real-world time series, however, may be obfuscated by noise and missing values arising in data acquisition. How can one learn the natural periodicity from incomplete multivariate time series?

We propose a robust framework for multivariate period detection, called LAPIS. It encodes incomplete and noisy data as a sparse representation via a Ramanujan periodic dictionary. LAPIS can accurately detect a mixture of multiple periods in the same time series even when 70% of the observations are missing. A key innovation of our framework is that it exploits shared periods across individual time series even when they time series are not correlated or in-phase. Beyond detecting periods, LAPIS enables improvements in downstream applications such as forecasting, missing value imputation and clustering. At the same time our approach scales to large real-world data executing within seconds on datasets of length up to half a million time points.

Index Terms—Period learning; Multivariate time series; Missing data imputation; Alternating Optimization;

I. Introduction

Multivariate time series data arise in many domains including sensor networks [47], social media [29], finance [19], databases [40] and others [14]. Key application tasks include prediction [15], control [5] and anomaly detection [47]. Understanding what constitutes "normal" seasonal behavior is critical for such down-stream applications. For example, accurate auto-regression-based forecasting models often include seasonality adjustment as a preliminary step [11]. In addition, periodicity learning has enabled the understanding of animal migration [24], effective anomaly detection in web applications [44] and in ECG readings [36] and periodic community detection [27].

Period learning in real-world time series is challenging due to i) missing values [9], (ii) noise [23] (iii) and multiplicity of periods (e.g., a combination of daily and seasonal oscillations) [38]. Missing data and noise typically arise due to measurement errors and outages within sensor networks such as urban air quality monitoring systems. Moreover there may be multiple periods in the air quality over time due to diverse contributing factors: transportation, agriculture, energy production, and others, which are all governed by similarly diverse periods: time of day, workweek, annual holidays, etc. The above challenges are common to many real-world systems and obscure the inherent periodicity in data [33].

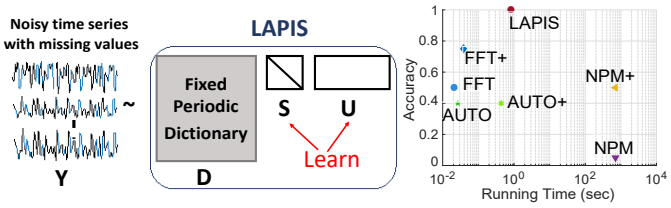


Fig. 1: LAPIS encodes noisy periodic time series with missing values, Y, via a fixed periodic dictionary D by learning a scale S and an encoding matrix U. Its accuracy in detecting complex ground truth periods in the data exceeds that of alternative approaches without incurring a large computational cost (right).

The problem we address in this paper is the detection of dominant periods from a multivariate time series with missing observations at random instances of time. Classical methodologies for period detection are based on the Fourier Transform (FT) [34], auto-correlation [13] and combinations of the two [42]. The topic has been pursued in multiple research communities: signal processing [38], data mining [23], databases [17]. and bioinformatics [39]. However, to the best of our knowledge, no prior work considers noise, missing values and multiplicity of periods jointly for multivariate timeseries.

We propose LAPIS, an effective framework to learn periods from multivariate time series which addresses all three design challenges: noise, missing values and period multiplicity (Figure 1(left). To deal with period multiplicity, we formulate period learning as a sparse encoding problem via a fixed Ramanujan periodic dictionary. While prior work has also employed Ramanujan-based encoding [38], our framework introduces a key innovation, namely, it sparsifies whole period selection by exploiting the group structure in the Ramanujan dictionary as opposed to sparsifying individual dictionary atom selection. This modeling decision improves robustness to

noise and missing values and promotes period sharing among individual time series. As a result LAPIS dominates state-of-the-art baselines in term of accuracy of ground truth period detection while ranking among the fastest alternatives in terms of scalability (Figure 1(right).

Our main contributions in this work are as follows:

Significance: To the best of our knowledge, LAPIS is the first model that can perform period learning on multivariate time series in the presence of missing values, multiple periods and noise.

Robustness: Our method is robust to noise and missing values thanks to learning common periods across time series by exploiting the group structure in the Ramanujan periodic dictionary.

Applicability: Extensive experiments on synthetic data and multiple real-world datasets demonstrate the superior quality and scalability of LAPIS.

II. Related work

Period Learning: Traditional period learning approaches in time series rely on Fourier transform [22], auto-correlations [13] or combinations of the two [42]. Methods in this group are sensitive to noise and often require prior information. These challenges were addressed in a recent periodic dictionary framework [38]. The general idea behind this framework was first introduced in the context of periodicity transforms [35] employed to project signals into a periodic subspace. This original method is limited to signals lengths that are multiples of a given period subspace. Tenneti et al. [38] improved this idea by developing a family of flexible periodic dictionaries. Our method generalizes periodic dictionaries to (i) multivariate time series where we impose a group structure of shared periodicity and (ii) to scenarios with missing values. We demonstrate experimentally that LAPIS is superior to basic periodic dictionary alternatives. Periodic behavior in discrete event sequences, as opposed to general time series, have also been considered in the literature. Li et al. [24] introduced a clustering-based solution to estimate the dominant period in event sequences. Yuan et al. [46] proposed an alternative probabilistic model for period detection. Methods from this group are effective for event data, however, they are not applicable to general time series data.

Missing Data Imputation: Although our primary goal is not to perform missing value imputation, a part of our model approximates missing values to recovery the periodic patterns in data. Therefore, we give a brief introduction on related work of missing value imputation. Missing data estimation in general matrices has wide applications in data mining and analytics [43]. The general idea is to model missing data under some regularity assumption for the full data, e.g., low-rank approximations [8], minimal description length [43], matrix profiles [4] and general matrix factorization [45]. More recently deep learning approaches for missing data imputation have also gained popularity including adversarial imitation learning [10] and structured multi-resolution via adversarial models [28]. These deep learning methods have a significant high complexity and require sufficient data to train models. For time series data, commonly used methods for missing value imputation are summarized [30], such as spline interpolation.

III. Preliminaries and notation

We denote with A_i, A^j, A_{ij} the i-th column, j-th row, and the i;j-th element of a matrix A respectively. In our development we will use the Frobenius norm kAk_F, the $L_{2;1}$ norm: $kAk_{2;1} = \sum_i \sqrt{\sum_j A_{ij}^2} = \sum_i kA_ik_2$, and the nuclear norm: $kAk_* = \sum_i$, where $_i$ is the ith singular value of A. We denote element-wise multiplication and division by $\cdot$ and $/$ respectively. I denotes the identity matrix.

Our framework will employ the Ramanujan basis within the nested periodic matrices (NPM) framework as a periodic dictionary [38]. An NPM for period g is defined as:

$$_g = [P_{d_1}; P_{d_2}; :::P_{d_K}]; \quad (1)$$

where $fd_1; d_2; :::d_Kg$ are the divisors of the period g sorted in an increasing order; $P_{d_i} \in R^{g(d_i)}$ is a period basis matrix for period d_i, where (d_i) denotes the Euler totient function evaluated at d_i. The basis matrix here is constructed based on the Ramanujan sum [38]:

$$C_{d_i}(g) = \sum_{k=1;gcd(k;d_i)=1}^{d_i} e^{j2kg=d_i}; \quad (2)$$

where $gcd(k; d_i)$ denotes the greatest common divisor of k and d_i. The Ramanujan basis is constructed as a circulant matrix of Ramanujan sums as follows:

$$P_{d_i} = \begin{bmatrix} C_{d_i}(0) & C_{d_i}(g-1) & ::: & C_{d_i}(1) \\ C_{d_i}(1) & C_{d_i}(0) & ::: & C_{d_i}(2) \\ ::: & ::: & ::: & ::: \\ C_{d_i}(g-1) & C_{d_i}(g-2) & ::: & C_{d_i}(0) \end{bmatrix} \quad (3)$$

The complete dictionary R for periods up to g_{max} is constructed by stacking basis matrices with periods from 2 to g_{max} as $R = [_1; ::; _{g_{max}}]$. Note that columns in R should be periodically extended to the same length as the time points of input time series. For example, the atoms for periods $f2; 3; 4g$ are defined as follows:

$$P_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}; P_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \\ -1 & -1 \end{bmatrix}; P_4 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \\ 2 & 0 \\ 0 & 2 \end{bmatrix}; \quad (4)$$

and are periodically extended for signals of length $T = 5$ as follows:

$$R = \left[\begin{array}{cc|cc|cc} 1 & -1 & 2 & -1 & 2 & 0 \\ 1 & -1 & -1 & 2 & 0 & 2 \\ 1 & -1 & -1 & -1 & 2 & 0 \\ 1 & -1 & 2 & -1 & 0 & 2 \\ 1 & -1 & -1 & 2 & 2 & 0 \end{array}\right] \quad (5)$$

A recent model for period estimation in univariate and complete (no missing data) signals proposed by Tenneti and colleagues [38] aims to obtain a succinct representation of the input signal through the NPM basis as follows:

$$\min_b kHbk_1; s:t: x = Rb; \quad (6)$$

where x is the input signal, b is the learned representation in NPM basis and H is a diagonal matrix encouraging representation through short periods defined as $H_{ii} = p^2$, where p is the period of the i-th column in R. One important choice in this framework is the maximal candidate period g_{max} as it determines the dimensionality of R. While we are not aware of an automated approach to set g_{max}, small values relative to T work well in practice since the framework explicitly encourages fit through short simple periods, e.g., a period of 20 could be represented as a mixture of its factors 2; 5.

IV. Problem formulation

The input in our problem settings is a partially observed multivariate time series matrix $Y \in R^{T \times N}$ with N time series of length T in its columns. We define an indicator (or a mask) matrix W of the same size as Y with entries $W_{ij} = 1$ if Y_{ij} is observed, and 0 otherwise. Our goal is to learn the inherent periodicity of time series in Y in a manner that is aware of the missing data. We formalize this objective by employing the Ramanujan periodic dictionary framework as follows:

$$\underset{S,U}{\operatorname{argmin}} \frac{1}{2} \| Y - DSU \|_F^2 + J_1(U) + J_2(Y); \qquad (7)$$

where $D = RH^{-1/2} \in R^{T \times L}$ is a "biased" periodic dictionary which promotes the selection of short-period atoms via a diagonal penalty matrix H^{-1} with elements $H_{ii}^{-1} = 1/p^2$, where p is the length of the period containing the i-th atom (column) in the Ramanujan basis dictionary R. The number of atoms L is determined by the number of columns in the Ramanujan dictionary. U is the period mixture matrix specifying the reconstruction of the time series as a linear combination of dictionary basis atoms.

Since individual time series might have different magnitude, we introduce a diagonal matrix $S \in R^{L \times L}$ to "absorb" the scale of the periodic reconstruction, thus, ensuring that loadings in U are of similar magnitude across time series. Note that this allows us to impose sparsity on U (discussed below) without bias to high-magnitude time series. We also incorporate two regularization terms: one promoting sparsity and grouping by period for the mixing matrix $J_1(U)$ and a second one handling missing values in the input $J_2(Y)$.

Sparsity and grouping by period in $J_1(U)$: Since time series typically have only a few inherent periods, we promote a sparse representation in the dictionary loadings through an L1 norm $\|U\|_1$ similar to other methods based on NPMs. This regularization allows for robustness to noise.

In addition to a sparse representation, we also expect our input time series Y to arise from the same system, and thus, share periods. Such shared periodicity is typical in both natural and engineered systems. For example, brain waves (or neural oscillations) govern the neural activity of all brain regions; daily oscillations of weather parameters such as pressure and temperature are across locations, and energy consumption within the same region of the energy grid observes periodic patterns with the same periods: days and seasons. Note that, we do not expect correlated behavior across the time series, but simply shared periods. Thus, we seek to enforce grouping of time series based on sharing a few periods in their periodic representation.

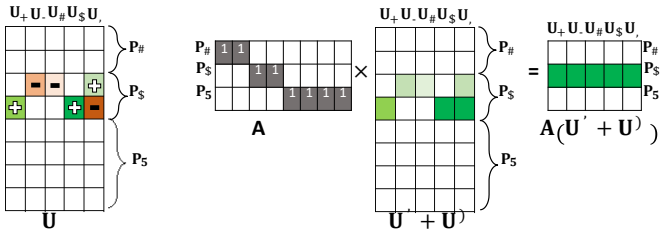


Fig. 2: A sketch of the expected "shape" of a periodic coefficient matrix U for the reconstruction of time series sharing a period (left). To impose a banded structure for period-specific atoms we employ a rank norm on an aggregation matrix $\|A(U^+ + U^-)\|_*$, where loadings of atoms corresponding to the same period are aggregated by absolute value (right).

The structure of the mixture matrix U aligned with the shared period assumption discussed above is presented in Fig. 2 (left). Intuitively we expect that mixture vectors U_i corresponding to individual time series have non-zero elements in bands corresponding to their periods, i.e. corresponding period-specific sub-matrices P_i. To promote such grouping banded structure we can employ a rank norm penalty. However, imposing a low-rank constraint directly on U will disregard the the period-specific band structure and treat selection of individual atoms from the dictionary independently. Instead, we need to impose a low-rank constraint at the period (band) level, by a product with a period aggregation matrix $A \in R^{P \times L}$ depicted in Fig. 2(right), which encodes the indices of atoms associated with a period in each of its rows.

If we directly aggregate mixture coefficients in U within a period, they may cancel out as in general elements of U can take both positive and negative values. Thus, we "separate" positive and negative values in U by defining two component matrices $U = U^+ - U^-$, where $U^+ = (|U| + U)/2$ and $U^- = (|U| - U)/2$. We then form the aggregate product $A(U^+ + U^-)$ Fig. 2(right) and impose a low rank penalty on it. The overall sparsity and period grouping regularizer $J_1(U)$ is defined as follows:

$$J_1(U) = \|U\|_1 + \|A(U^+ + U^-)\|_*; \qquad (8)$$

Note that we employ the nuclear norm for the grouping term as it is the convex relaxation of the rank norm.

Missing data imputation: Recall that Y is partially observed due to missing values which prevent us from minimizing directly the reconstruction of Y as outlined in Eq. 7. To handle this, we define a proxy full data matrix X with all values imputed which we represent by the periodic dictionary and also maintain close to the input Y for observed values. Formally, we define the missing value regularizer $J_2(X)$ as:

$$J_2(X) = \|W \odot (X - Y)\|_1; \qquad (9)$$

where W is a mask matrix with elements of 1 in positions in which values in Y are observed and 0 otherwise; and $\odot$ denotes the element-wise product.

Our overall objective for the problem of Period learning with missing values is as follows:

$$\begin{aligned}\underset{X,S\geq 0,U}{\operatorname{argmin}} \; & \frac{1}{2}\|X - DSU\|_F^2 + \alpha_1\|U\|_1 \\ & + \alpha_2\|A - U^+ + U\|_F^2 + \alpha_3\|W \odot (X - Y)\|_1,\end{aligned} \tag{10}$$

where α_1, α_2 and α_3 are balance parameters. Note that we have replaced Y in Eq. 7 with X, i.e. we have integrated the missing data imputation model within the period learning objective.

V. Optimization

Next we derive the optimization algorithm for our problem formulation in Eq. 10. Since variables $\{X, U, S\}$ are not jointly convex in this objective, it can't be solved by gradient based methods. Instead, we develop an efficient solver based on alternating optimization. More specifically, we decompose Eq. 10 into three variable-specific subproblems and update them alternatively as follows:

$$\begin{cases} \underset{U}{\operatorname{argmin}} \; \frac{1}{2}\|X - DSU\|_F^2 + \alpha_1\|U\|_1 + \alpha_2\|A - U^+ + U\|_F^2 \\ \underset{X}{\operatorname{argmin}} \; \frac{1}{2}\|X - DSU\|_F^2 + \alpha_3\|W \odot (X - Y)\|_1 \\ \underset{S\geq 0}{\operatorname{argmin}} \; \frac{1}{2}\|X - DSU\|_F^2 \end{cases}$$

A. Optimization of U.

We solve the subproblem for U by the Alternating Direction Method of Multipliers (ADMM) [6]. We first introduce proxy variables $\{P_1, P_2\}$ and rewrite the problem as follows:

$$\begin{aligned}\underset{U,P_1,P_2}{\operatorname{argmin}} \; & \frac{1}{2}\|X - DSU\|_F^2 + \alpha_1\|P_1\|_1 + \frac{1}{2}\alpha_2\|P_2\|_F^2 \\ & \text{s.t. } P_1 = U, P_2 = A - U^+ + U\end{aligned}$$

We construct the corresponding Lagrangian function:

$$\begin{aligned}R(U, P_1, P_2, \Lambda_1, \Lambda_2) = & \frac{1}{2}\|X - DSU\|_F^2 \\ & + \alpha_1\|P_1\|_1 + \langle \Lambda_1, P_1 - U\rangle + \frac{\beta_1}{2}\|P_1 - U\|_F^2 \\ & + \alpha_2\|P_2\|_F^2 + \langle \Lambda_2, P_2 - A + U^+ - U\rangle \\ & + \frac{\beta_2}{2}\|P_2 - A + U^+ - U\|_F^2,\end{aligned} \tag{11}$$

where Λ_1 and Λ_2 are Lagrange multipliers; β_1 and β_2 are penalty parameters. We perform the optimization of each variable in R alternatively and derive the update rules in the remainder of this subsection.

Update of U: The subproblem w.r.t. U is:

$$\underset{U}{\operatorname{argmin}} \; \frac{1}{2}\|X - DSU\|_F^2 + \frac{\beta_1}{2}\|P_1 - U + \frac{\Lambda_1}{\beta_1}\|_F^2 \tag{12}$$

We set its gradient w.r.t. U to zero and obtain:

$$\hat{D}^T\hat{D}U - X + \beta_1(U - P_1 - \Lambda_1/\beta_1) = 0; \tag{13}$$

where $\hat{D} = DS$, with a closed-form solution for U:

$$U = \left(\hat{D}^T\hat{D} + \beta_1 I\right)^{-1}\left[\hat{D}^T X + \beta_1(P_1 + \Lambda_1/\beta_1)\right].$$

Update for P_1: The P_1 subproblem is as follows:

$$\underset{P_1}{\operatorname{argmin}} \; \alpha_1\|P_1\|_1 + \frac{\beta_1}{2}\|P_1 - U + \Lambda_1/\beta_1\|_F^2 \tag{14}$$

Problems of this from have a closed form solution due to [26] specified in the following Lemma:

Lemma 1: The objective $\operatorname{argmin}_S \frac{1}{2}\|S - B\|_F^2 + \lambda\|S\|_1$, where λ is a positive scalar has a closed-form solution: $S_{ij} = \operatorname{sign}(B_{ij})\max(|B_{ij}| - \lambda, 0)$, where $\operatorname{sign}(t)$ is the signum function.

Thus, by letting $F_1 = U - \Lambda_1/\beta_1$, we have the following element-wise update rule for P_1:

$$[P_1]_{ij} = \operatorname{sign}([F_1]_{ij})\max(|[F_1]_{ij}| - \alpha_1/\beta_1, 0)$$

Update for P_2: The P_2 subproblem is as follows:

$$\underset{P_2}{\operatorname{argmin}} \; \alpha_2\|P_2\|_F^2 + \frac{\beta_2}{2}\|P_2 - A + U^+ - U + \Lambda_2/\beta_2\|_F^2$$

This problem can also be readily solved by employing the singular value thresholding (SVT) method [7]. Let $F_2 = A - (U^+ - U) - \Lambda_2/\beta_2$ and let $F_2 = LSR^T$ be the singular value decomposition (SVD) of F_2. Then following the SVT approach, we update P_2 as $P_2 = LD(S)R^T$, where $D(\cdot)$ is a thresholding function that is defined as $D(S) = \operatorname{diag}\left[\max(S_i - \frac{2\alpha_2}{\beta_2}, 0)\right]$, where S_i denotes the ith singular value of F_2.

Updates for the Lagrange multipliers Λ_1, Λ_2: We update the Lagrange multipliers as follows:

$$\begin{cases} \Lambda_1^{t+1} = \Lambda_1^t + \beta_1[P_1^t - U^t] \\ \Lambda_2^{t+1} = \Lambda_2^t + \beta_2[P_2 - A + (U^+ + U)^t] \end{cases} \tag{15}$$

B. Optimization of X.

Since X is involved in an element-wise product term, its elements cannot be optimized simultaneously. To address this, we adopt ADMM approach similar to that for U. We first introduce a proxy variable E as follows:

$$\underset{X,E}{\operatorname{argmin}} \; \frac{1}{2}\|X - DU\|_F^2 + \alpha_3\|W \odot E\|_1 \; \text{s.t. } E = X - Y$$

Next we form the corresponding Lagrangian form:

$$\begin{aligned}L(X, E, M, \beta) = & \frac{1}{2}\|X - DU\|_F^2 + \alpha_3\|W \odot E\|_1 + \\ & \langle M, E - (X - Y)\rangle + \frac{\beta}{2}\|E - (X - Y)\|_F^2,\end{aligned}$$

where M is a Lagrange multiplier and β is a penalty coefficient. We perform the optimization of each variable in $L(X, E, M)$ alternatively as follows:

$$\begin{cases} X^{k+1} = \underset{X}{\operatorname{argmin}} \; L(X, E^k, M^k, \beta^k) & (a) \\ E^{k+1} = \underset{E}{\operatorname{argmin}} \; L(X^{k+1}, E, M^k, \beta^k) & (b) \\ M^{k+1} = M^k + \beta^k(E^{k+1} - X^{k+1} + Y) & (c) \end{cases}$$

The explicit minimization for X^{k+1} is follows:

$$X^{k+1} = \operatorname{argmin}_X \frac{1}{2}\|X - DU\|_F^2 + \frac{k}{2}\|E^k - (X - Y) + M^k/k\|_F^2$$

We set the gradient w.r.t X to zero:

$$X - DU + {}^k(X - H) = 0;$$

where $H = E^k + Y + \frac{M^k}{k}$, and obtain:

$$X^{k+1} = DU + {}^kH = 1 + {}^k \quad (16)$$

The minimization for E^{k+1} is:

$$E^{k+1} = \operatorname{argmin}_E {}_3\|W \quad E\|_1 + \frac{k}{2}\|E - X^{k+1} + Y + M^k/k\|_F^2$$

Here we again employ Lemma 1 and let $T = X^{k+1} - Y - \frac{M^k}{k}$, resulting in:

$$E^{k+1}_{ij} = \operatorname{sign}(T_{ij}) \max |T_{ij}| - {}_3 W_{ij}/k; 0$$

C. Optimization of S.

We introduce a Lagrangian multiplier to ensure non-negativity of S' entries and obtain the following Lagrangian function:

$$G(S;) = 1=2\|X - DSU\|_F^2 - \operatorname{tr}(S^T) \quad (17)$$

Taking the gradient of G w.r.t S, we get:

$$@G=@S = D^T DSUU^T - D^T XU^T : \quad (18)$$

Based on the KKT conditions, we also have the system:

$$@G=@S = 0; \quad S = 0 \quad (19)$$

By combining Eq. 18 and Eq. 19, we obtain:

$$D^T DSUU^T - D^T XU^T S = 0: \quad (20)$$

We define the following simplifying variables: $L = D^T XU^T$, $G = D^T D$ and $H = UU^T$. Since D, U and X are all mix-signed, we split each of the new variables into positive and negative summands:

$$\begin{cases} H = H^+ - H \\ G = G^+ - G \\ L = L^+ - L ; \end{cases} \quad (21)$$

similar to the definitions of U^+ and U. Then, we rewrite $C = HSG - L$ in terms of the two-part definitions for the new variables as follows:

$$\begin{aligned} C &= H^+ - H \; S \; G^+ - G - L^+ - L \\ &= H^+SG^+ + H \; SG \; + L \\ &\quad H \; SG^+ + H^+SG \; + L^+ \end{aligned}$$

Algorithm 1 Optimization of LAPIS

Require: The incomplete data Y and parameters $({}_1, {}_3)$.
Ensure: The estimated data X and period coefficients U

1: Initialize: X Y, S I, ${}_1 = 0$, ${}_2 = 0$ and $M = 0$ 2:
while X, U and S have not converged do
3: while U has not converged do
4: $U = (D^T D + {}_1 I - {}_1 H)^{-1} (D^T X + {}_1 P_1 + {}_1 \frac{}{1})$
5: $U^+ = (|U| + U)/2$
6: $U = (|U| - U)/2$
7: $[P_1]_{ij} = \operatorname{sign}([F_1]_{ij}) \max([F_1]_{ij} - \frac{1}{1}; 0$ 8:
$F_2 = A \; U^+ + U \; {}_2 \frac{}{2}$
9: $(L; ; R) = \operatorname{svd}(F_2)$
10: $P_2 = LD() R^T$
11: ${}^{t+1}_1 = {}^t_1 + {}_1 P^t_1 (U^t$
12: ${}^{t+1}_2 = {}^t_2 + {}_2 P^t_2 (A \; U^+ + U \; {}^t$ 13:
end while
14: while X and E have not converged do
15: $X^{k+1} = DU + \frac{k}{2}H - 1 + \frac{k}{2}$
16: $E^{k+1}_{ij} = \operatorname{sign}(T_{ij}) \max |T_{ij}| - {}_3 W_{ij} \frac{}{k}; 0$
17: $M^{k+1} = M^k + {}^k E^{k+1} - X^{k+1} - Y$
18: ${}^{k+1} = {}^k$
19: end while
20: $S_{ij} \quad S_{ij} \sqrt{\frac{[H \; SG^+ + H^+SG \; + L^+]_{ij}}{[H^+SG^+ + H \; SG \; + L \;]_{ij}}}$
21: end while
22: return $\{U; X\}$

Finally, we obtain an element-wise update for S following the approach in [18]:

$$S_{ij} \quad S_{ij} \sqrt{\frac{[H \; SG^+ + H^+SG \; + L^+]_{ij}}{[H^+SG^+ + H \; SG \; + L \;]_{ij}}} \quad (22)$$

D. Overall algorithm and complexity.

We list the steps of our method LAPIS in Algorithm 1. We repeat the sequential updates for U; X and S, Steps 3-18 until convergence. The major running time cost comes from Steps 4 and 9 while the remaining steps have a linear complexity or involve sparse matrix multiplication. Step 4 involves an inversion of a quadratic matrix, which is in general $O(L^3)$. However, due to the sparsity in D, this complexity can be reduced to $O(LS_{nz})$ following the analysis in [48], where S_{nz} is the number of non-zero elements of $D^T D + {}_1 I$. In Step 9, the SVD operation has complexity of $O(\min(LN^2; L^2N))$. The overall complexity is dominated by $O(t_S t_U LS_{nz} + t_S t_U \min(LN^2; L^2N))$, where t_S and t_U are the number of iterations of the loops starting in Steps 2 and 3 respectively. In practice, when employed for time series of half a million time points, LAPIS takes a few seconds on a standard desktop machine. The implementation of LAPIS will be made available with this camera-ready paper.

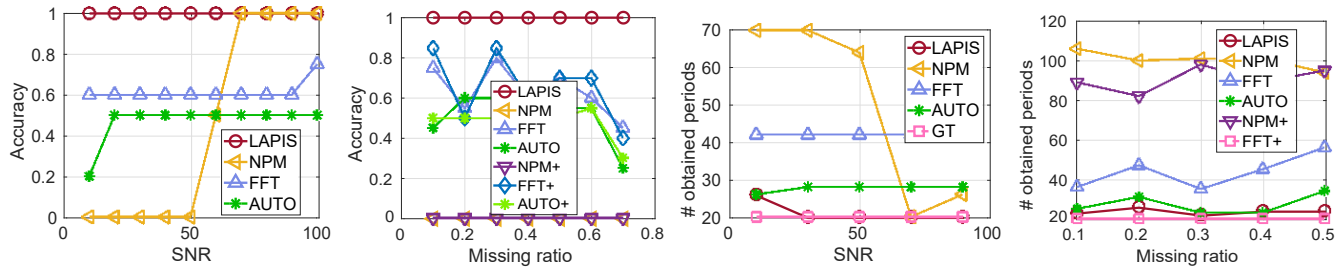
VI. Experimental Evaluation

A. Datasets

Synthetic data: We generate 10 periodic 800-step-long time series following the protocol in [38]. The time series form 3 groups sharing the same within-group ground truth (GT)

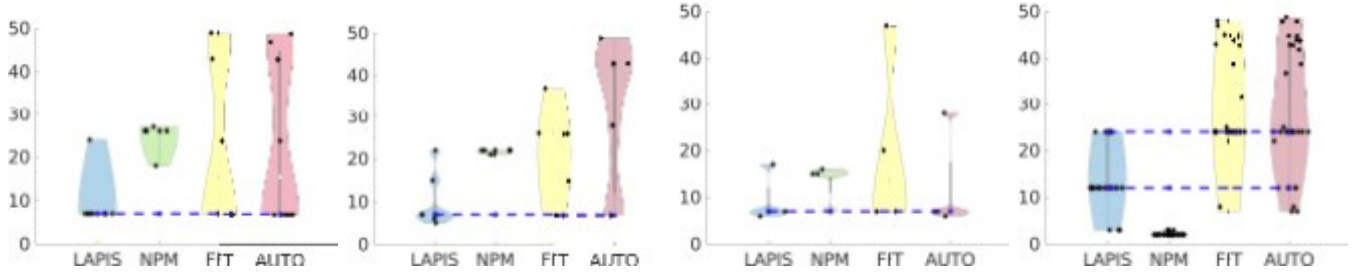
	Statistics			LAPIS		NPM		FFT		AUTO	
Dataset	Length	N variate	GT period	Periods	Time	Periods	Time	Periods	Time	Periods	Time
Web traffic [2]	550	8	7 days	7	0.3s	26	4.1s	7	0.1s	7	0.03s
Bike rentals [16]	731	16	7 days	7	0.5s	22	4.8s	26	0.1s	43	0.02
Beer consumption [3]	365	4	7 days	7	0.8s	15	0.6s	7	0.6s	7	0.04s
Historical weather [1]	45k	36	12; 24 hours	12; 24	8.3s	2; 3	12600s	24; 45	4s	24; 45	0.6s
Air Quality [12]	52k	10	12; 24 hours	12; 24	25.4s	3,5	32817s	24,49	0.3s	24,49	0.2s

TABLE I: Real-world dataset statistics and comparison of period detection and running times of competiting techniques.



(a) Accuracy with SNR (b) Accuracy with missing% (c) number of periods with SNR (d) number of periods with missing%

Fig. 3: Period learning comparison of competing techniques on synthetic data (20 ground truth periods)



(a) Web traffic (b) Bike rental (c) Beer consumption (d) Historical weather

Fig. 4: Period learning on real-world datasets

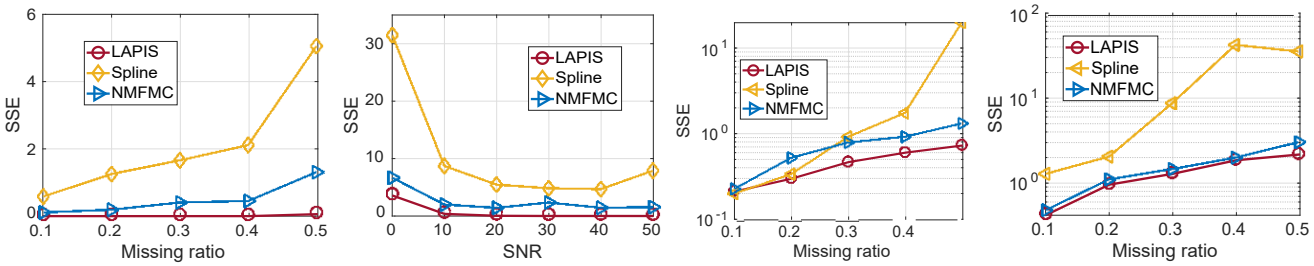
periods, where each group has 2 periods. We add Gaussian noise of varying strength in all time series by using the same implementation as in [38]. We then remove observations independently in each series and at random steps.

Real-world data: We evaluate our model on datasets from several domains summarized in Tbl. I. Note that all datasets are multivariate time series, we use N variate to denote the number of components in the Table 1. The Web traffic dataset [2] tracks the daily views of Wikipedia articles aggregated per language between 06/2015 and 12/2016. We expect that seven days (one week) is a natural ground truth period in this dataset. Our Bike rental dataset [16] is a two-year daily log of the number of bike rentals at different stations in Washington D.C. Since rental bikes are used for daily commutes in urban areas, we expect that the GT period in this data will also be driven by the work week. The Beer consumption dataset [3] tracks the daily beer consumption in the Czech Republic. We expect that consumption will vary with people's weekly routines and also expect a weekly period in this data. The Historical Weather dataset [1] consists of 5 years of hourly weather measurements. We use the pressure time series at 36 US cities. Since the pressure is directly affected by the solar position we expect half-day or daily GT periods in this dataset. Finally, the Air Quality dataset [25] contains the PM2.5 measurements for five cities in China between 1/2010 and 12/2015. PM2.5 levels measure small (less than 2.5nm) airborne particles, predominantly generated by burning of fossil fuel. Therefore, the PM2.5 index is related to automobile emissions, industrial activity, and construction work. These activities are likely to follow human behavioral patterns, which will often lead to periods of half (12 hours) or full days (24 hours) [21]. In this experiment, we report the results based on data from Jingan, Shangai, China.

A note on expected periods: While ground truth periods are not usually available for many of the above datasets, since these datasets represent human activity we expect intuitively "natural" periods - a (half) day, a week, a month, a year. Otherwise, the detected periods are more difficult to interpret as driven by phenomena from the respective domains.

B. Experimental setup

Baselines: The key goal of our model is period learning but it can also be employed to perform seasonality adjustment and impute missing data or project future behavior. For period learning, we compare our method to three baselines: NPM [38] which is the state-of-art period learning method, FFT [22], and AUTO [24] which combines auto-correlation and Fourier transform. In particular, these three baselines are not designed



(a) Synthetic data (b) Synthetic data (c) Beer consumption (d) Air quality

Fig. 5: Evaluation of missing value imputation.

for multivariate time series directly, therefore we apply them on each univariate time series and combine the results. We also compare LAPIS to two period-agnostic baselines for missing data imputation: Spline interpolation [30] and matrix completion NMFMC [45]. We do not compare to deep learning methods because they have significantly higher complexity and require large amount of data for training.

To test if the period learning baselines are mainly affected by missing data, we also attempt to impute missing values and then learn periods, resulting in no-missing value variants: NPM+, FFT+ and AUTO+, where we combine period-agnostic missing data imputation with period learning baselines. Here, we employ NMFMC to impute values before employing NPM, FFT and AUTO because it outperforms the Spline method in all datasets. We also quantify the utility of seasonality adjustment by LAPIS for improving auto-regressive prediction models.

Metrics: In the period learning task we use the accuracy of detecting ground truth (GT) periods for evaluation. Among all detected periods by competing methods, we employ the top-k periods for computing the accuracy, where k is the number of GT periods. For missing values imputation and forecasting, we adopt Sum of Squared Errors (SSE) for evaluation.

C. Period learning analysis.

We compare LAPIS to baselines on synthetic data for varying signal to noise ratio (SNR) and the percentage of missing values in Fig. 3. In Fig. 3a we compare the meth-ods on synthetic data with no missing values and varying SNR. LAPIS consistently detects all periods at 100%. NPM achieves 100% accuracy only for large SNR indicating that it is sensitive to noise. FFT and AUTO achieve a maximal 70% and 50% accuracy for the lowest noise is regime. To further demonstrate our model's robustness, we quantify the total number of detected periods. We scale the magnitude of the period detection vector reported by each method and threshold it at 0:2. We report the obtained number of periods for all methods in Fig. 3c. LAPIS typically reports close to the ground truth number of periods (20 in this experiment), while competitors tend to detect a large number of spurious periods not in the GT. While both LAPIS and NPM employ Ramanujan dictionaries, our methods is less sensitive to noise since it exploits sharing of periods across time series and further enforces a block structures in the periodic dictionary which NPM neglects.

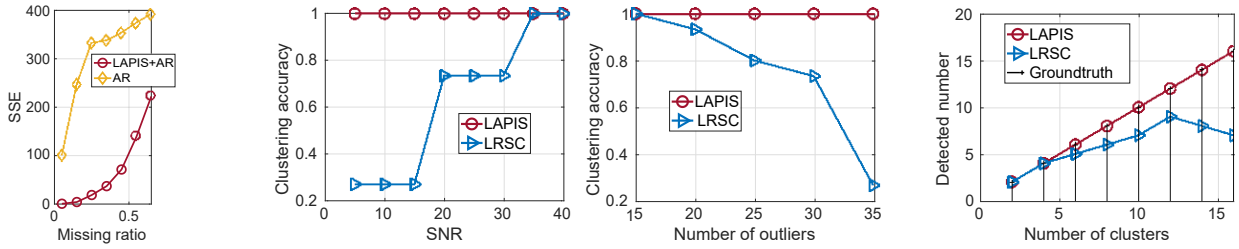
LAPIS retains perfect detection of GT periods for increasing fraction of missing values (up to 70%) due to its explicit modeling of missing values in data (Fig. 3b). In comparison, alternatives (FFT and AUTO) degrade steadily even when missing value imputation is performed as a prepossessing (FFT+ and AUTO+). This suggests that period-agnostic missing value imputation does not help significantly traditional period learning approaches. NPM fails to detect periods in the presence of missing values, i.e., when missing values are filled with a constant (NPM) or by NMFMC (NPM+), as the inconsistency of the imputed values "break" the periodic patterns in the data.

Period learning results in real-world data are reported in Fig. 4 as violin plots of the distribution of predicted periods, where the GT periods are shown as horizontal dashed blue lines. LAPIS's prediction are typically concentrated around the expected GT periods while baselines are less accurate and more scattered. LAPIS enforces period sharing across time series, while others assume independence between time series and tend to overfit observations, yielding a wide range of predictions. We further summarize the period learning results and running times in real-world data in Tbl. I.

D. Missing value imputation.

While LAPIS extract the periodic component from observed time series and is not directly a missing value predictor. However we employ the estimated periodic-based values for missing value imputation. The performance of missing value imputation in synthetic data for varying SNR and varying ratio of missing data is shown in Fig. 5a and Fig. 5b respectively. Though LAPIS is not explicitly designed for missing values imputation, it outperforms period-agnostic baselines based on interpolation Spline and matrix factorization NMFMC. Spline considers temporal smoothness, however, it only models local changes ignoring periodic behavior. As a result , Spline is sensitive to noise and missing. NMFMC estimates missing values based on a low-rank structure in the overall data matrix Y, which leads to robustness to noise, however, it does not exploit temporal dependencies of the values.

We also evaluate missing value imputation on the Beer consumption and Air Quality d datasets in Fig. 5c and Fig. 5d respectively. We vary the missing data ratio by randomly dropping observations. Spline has the worst performance among



(a) Projection (b) Vary SNR (c) Vary number of outliers (d) Vary number of cluster

Fig. 6: (a): Total SSE for projection over 100 time points after training on 700, with (LAPIS +AR) and without (AR) seasonality adjustment via LAPIS.; Evaluation of clustering accuracy (b) varying SNR; (c) varying the number of outliers. Figure (d) shows the detected number of clusters when varying the number of clusters in the data.

competitors due to the inherent noise in real-world data to which the method is sensitive. The performance of LAPIS and NMFMC are closer with the advantage of LAPIS-based imputation increasing with the number of missing observations. We believe that better trend estimation, instead of the simple moving average we adopt, can further improve period-aware missing value imputation, however, such evaluation is beyond the scope of the current paper and plan to explore it as future work.

Note that real world time series often exhibit patterns beyond periodic, such as trends. It is worthy to consider more complex methods for estimating trend after seasonality adjustment can be employed to improve the prediction, however, this is beyond the scope of our paper whose focus is on estimating the periodic behavior.

E. LAPIS for forecasting in time series.

To demonstrate another direct application of accurately learned periodicity, we present the benefit of accounting for LAPIS-derived periodic behavior in predicting the future values of a stationary time series. A time series with a trend and periodic components can be represented as $x = x_{per} + x_{tr} +$, capturing recurring periodic or seasonal components in x_{per} and overall aperiodic trend in x_{tr}, with error term . Account-ing for periodicity in the data enables a method optimized for detecting consistent trends in data, such as auto-regression (AR), to more effectively detect the evolution of the time series as a whole.

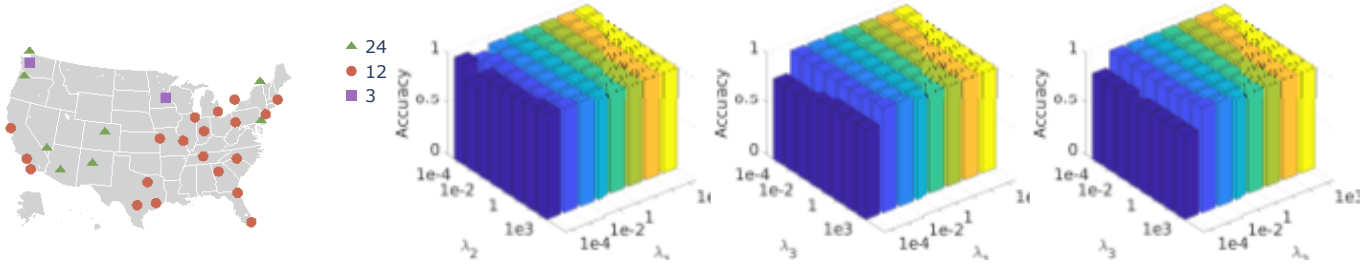
We generate a periodic multivariate time series as described above and train on the first 700 points. A baseline model is a 10-lag vector AR model (which allows for co-movement between series), which is used to forecast the final 100 time points directly. To allow for fit of missing values while preserving possible interactions between series, we replace missing values with the mean of the stationary time series instead of ignoring the time point. Alternatively, we use LAPIS to learn the periodic approximation X as well as the periodic structure U. The same AR model is then fit to Y X, capturing the trend alone, and the forecast from this AR model is then added to the projected periodic component from D S Ũ, where D is̃ the periodic dictionary extended to the final 100 time points which we predict. This procedure is repeated for multiple missing ratios, and results are presented in Fig. 6a. Removing correctly-identified periodic components from the time series, as enabled by LAPIS, leads to more accurate prediction even over long prediction horizons, particularly in the presence of missing values.

F. LAPIS for time series clustering.

LAPIS captures the structure among all time series according to the period of each univariate time series by imposing the low-rank regularization kZk , where $Z = A(U^{+} + U$). We next validate the effectiveness of this learned inter-series structure. Based on the low-rank structure, we can obtain time series clusters by using the linear dependency among the columns in Z, where a column in Z represents the period coefficients of the corresponding univariate time series. Based on the Cauchy-Schwarz inequality [37], one can determine if two columns are linearly dependent if their inner product equals the product of their norms. We employ the Cauchy-Schwarz inequality on Z to determine the clusters of columns, namely columns form a cluster if they are all linearly dependent.

Notice that LAPIS can automatically obtain the number of clusters by estimating the rank of Z. Commonly used clustering methods, such as k-means [20], spectral clustering [31], and k-Multiple-Means [32], often expect the number of clusters as a required input. However, this number is not known apriori in general. For a fair comparison, we focus on a baseline which does not require the number of clusters as an input. Rank minimization-based clustering methods share this advantage, and among them we employ a low-rank subspace clustering (LRSC) [41] as the baseline. Analogous to the approach for our method, we apply Cauchy-Schwarz inequality to get the final clusters for LRSC. We use the clustering accuracy for performance evaluation, which is defined as the fraction of correctly assigned data points out of the total number of data points. Each data point in our setting corresponds to one univariate time series.

We generate a periodic multivariate time series as in our synthetic data protocol. We present results of the clustering for varying SNR and number of outliers in Figs. 6b,c. The results were averaged over 5 independent runs of data generation. LAPIS clusters all data points precisely and consistently because it is insensitive to outliers and noise as demonstrated



(a) Location periods from Weather (b) Sensitivity $_{2;1}$ (c) Sensitivity $_{3;1}$ (d) Sensitivity $_{2;3}$

Fig. 7: (a): Periodicity in weather patterns in US cities. The legend specifies the estimated period in hours. (b-d) Parameter sensitivity analysis for LAPIS.

in all previous experiments. The success of LRSC relies on the quality of reconstruction of the full raw data, therefore, it is sensitive to outliers and noise. As a result, the performance of LRSC is significantly reduced when increasing the number of outliers or noise magnitude.

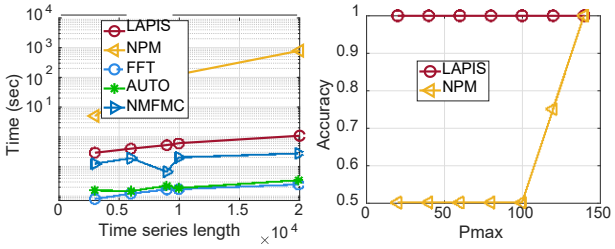
We also present the number of clusters detected by each method in Fig. 6d. We fix the number of instances to 50 and vary the number of clusters from 2 to 16. LAPIS obtains the optimal number of clusters steadily as periodicity can be robustly obtained. At teh same time LRSC increasingly underestimate the number of clusters with increasing number of true cluster. Since the number of instances in each cluster is reduced, the features of instances become more similar aligned with the self-representation approach of LRSC. Therefore, the clustering error grows as the margin among clusters narrows. Note that this is not necessary true in general time series clustering, but holds for periodic time series as the quality of their clustering depends on the accuracy in period detection.

G. Case Study: weather periodicity.

The periods produced by LAPIS are physically meaningful, as demonstrated in Fig. 7a. The data visualized are learned periods from the Historical Weatherdataset. The majority of the time series produce meaningful 24-hour or 12-hour periods, representing daily or morning/evening cycles in atmospheric pressure. Higher periods indicate greater variability in the data, suggesting more complex repeating patterns necessary for a meaningful representation of the series as a whole. As such, we see shorter periods (12 hours) largely in the drier regions of the southwest along with some northern locations; more complicated patterns are seen along the coasts and around lake- and river-"heavy"' regions of the eastern US. The two regions of 3-hour periods are interesting but difficult to interpret as they are. This may be suggestive of particularly stable weather in those cities, or of flaws in the data leading to less reliable results for those regions.

H. Running time and parameter sensitivity analysis.

In Fig. 8a, we present a comparison of the running time of competitors. Thanks to the low-rank constraint in our model, LAPIS is able to converge faster than NPM. While we set the maximal Ramanujan dictionary period $P_{max} = 30$ for both LAPIS and NPM, however, the running time of NPM is



(a) Scalability (b) Sensitivity P_{max}

Fig. 8: (a) The running time on synthetic data; and (b) the periodic dictionary parameter analysis

still much larger than all other competitors. While LAPIS is slightly slower than simpler alternatives for period detection (FFT, AUTO) and missing values imputation (NMFMC), it exhibits better accuracy across tasks and datasets and completes within seconds on all real data, making it practically applicable.

Since both LAPIS and NPM use a periodic dictionary, we also study the impact of the maximal period P_{max}, which controls dictionary size. We present results on synthetic data with no missing values in Fig. 8b. LAPIS achieves 100% accuracy for period learning with a wide range of P_{max}. This robust behavior is due to its consideration of the block structure in the periodic dictionary R. Our model achieves optimal results with P_{max} as low as 20. In comparison NPM needs to consider as many as $P_{max} = 140$ to get comparable accuracy as it treats all dictionary atoms as independent from each other. Note that the corresponding dictionary columns for $P_{max} = 20$ are 128, while for $P_{max} = 140$ this number is 6000. This difference corresponds to a large reduction in computation costs for LAPIS without reduction in accuracy.

We also evaluate the sensitivity of our model to hyperparameters $f_1; _2; _3 g$. We fix one parameter and vary the other two, and present the accuracy of period learning when varying parameters. In this experiment, we use synthetic data with the missing ratio fixed at 50% and $SNR = 5$. The cases of fixing $_3$ and $_2$ are demonstrated in Fig. 7b, 7c and 7d, respectively. Based on these figures, the performance of LAPIS is not sensitive to its parameters within significantly large ranges.

VII. Conclusion

In this paper, we tackled the problem of learning periods from multivariate time series in the presence of noise, missing values, and multiple periods. We developed an efficient convex model and an optimization framework, called LAPIS, to represent and learn the periods in multivariate time series as a sparse approximation via a Ramanujan periodic dictionary. We applied LAPIS to both synthetic and real-world datasets and demonstrated its consistent advantage over state-of-the-art baselines. LAPIS was able to learn the correct periods (100% accuracy) even when 70% of the values were missing.

VIII. Acknowledgements

This work is supported by the National Science Foundation (NSF) Smart and Connected Communities (SC&C) grant CMMI-1831547.

References

[1] Historical Hourly Weather Data 2012-2017, url = https://www.kaggle.com/selfishgene/historical-hourly-weather-data, year=2017.

[2] Wikipedia Traffic Data Exploration, url = https://www.kaggle.com/c/web-traffic-time-series-forecasting/data, year=2017,.

[3] Daily beer consumption, 2019.

[4] M. Azur, E. Stuart, C. Frangakis, and P. Leaf. Multiple imputation by chained equations: What is it and how does it work? International Journal of Methods in Psychiatric Research, 20(1):40–49, 3 2011.

[5] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. Time series analysis: forecasting and control. John Wiley & Sons, 2015.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn., 3(1):1–122, Jan. 2011.

[7] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. SIAM J. on Optimization, 20(4):1956–1982, Mar. 2010.

[8] E. Candès and B. Recht. Exact matrix completion via convex optimization. Commun. ACM, 55(6):111–119, June 2012.

[9] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li. Brits: Bidirectional recurrent imputation for time series. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31, pages 6775–6785. Curran Associates, Inc., 2018.

[10] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li. BRITS: bidirectional recurrent imputation for time series. CoRR, abs/1805.10572, 2018.

[11] C. Chatfield. Time-series forecasting. Chapman and Hall/CRC, 2000.

[12] Q. G. Data. Waves measuring buoys data. 2019.

[13] M. E. P. Davies and M. D. Plumbley. Beat tracking with a two state model. 2005.

[14] P. Esling and C. Agon. Time-series data mining. ACM Computing Surveys (CSUR), 45(1):12, 2012.

[15] C. Faloutsos, J. Gasthaus, T. Januschowski, and Y. Wang. Forecasting big time series: Old and new. Proc. VLDB Endow., 11(12):2102–2105, Aug. 2018.

[16] H. Fanaee-T and J. Gama. Event labeling combining ensemble detectors and background knowledge. Progress in Artificial Intelligence, pages 1–15, 2013.

[17] P. Fournier Viger, C.-W. Lin, Q.-H. Duong, and T.-L. Dam. Phm: Mining periodic high-utility itemsets. volume 9728, pages 64–79, 07 2016.

[18] Q. Gu and J. Zhou. Co-clustering on manifolds. pages 359–368, 06 2009.

[19] K. Gupta and N. Chatterjee. Financial time series clustering. In International Conference on Information and Communication Technology for Intelligent Systems, pages 146–156. Springer, 2017.

[20] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. JSTOR: Applied Statistics, 28(1):100–108, 1979.

[21] H. Kambezidis, R. Tulleken, G. Amanatidis, A. Paliatsos, and D. Asimakopoulos. Statistical evaluation of selected air pollutants in athens, greece. Environmetrics, 6:349 – 361, 07 1995.

[22] Z. Li, B. Ding, J. Han, R. Kays, and P. Nye. Mining periodic behaviors for moving objects. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1099–1108. ACM, 2010.

[23] Z. Li and J. Han. Mining Periodicity from Dynamic and Incomplete Spatiotemporal Data, pages 41–81. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[24] Z. Li, J. Wang, and J. Han. eperiodicity: Mining event periodicity from incomplete observations. IEEE Trans. Knowl. Data Eng., 27(5):1219–1232, 2015.

[25] X. Liang, S. Li, S. Zhang, H. Huang, and S. Xi Chen. Pm2.5 data reliability, consistency and air quality assessment in five chinese cities. Journal of Geophysical Research: Atmospheres, 08 2016.

[26] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. ArXiv, abs/1009.5055, 2013.

[27] A. G. Lin Zhang and P. Bogdanov. Perceids: Periodic community detection. In IEEE International Conference on Data Mining (ICDM), 2019.

[28] Y. Liu, R. Yu, S. Zheng, E. Zhan, and Y. Yue. NAOMI: non-autoregressive multiresolution sequence imputation. CoRR, abs/1901.10946, 2019.

[29] Y. Matsubara, Y. Sakurai, C. Faloutsos, T. Iwata, and M. Yoshikawa. Fast mining and forecasting of complex time-stamped events. In KDD, 2012.

[30] S. Moritz and T. Bartz-Beielstein. imputets: Time series missing value imputation in r. R Journal, 9(1), 2017.

[31] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Anal-ysis and an algorithm. In ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, pages 849–856. MIT Press, 2001.

[32] F. Nie, C.-L. Wang, and X. Li. K-multiple-means: A multiple-means clustering method with specified k clusters. pages 959–967, 07 2019.

[33] R. K. Pearson, H. Lähdesmäki, H. Huttunen, and O. Yli-Harja. Detecting Periodicity in Nonideal Datasets, pages 274–278. 2003.

[34] M. Priestley. Spectral analysis and time series. Probability and mathematical statistics. Elsevier Academic Press, London, 1981 (Rep. 2004). v. 1. Univariate series.– v. 2. Multivariate series, prediction and control.

[35] W. Sethares and T. Staley. Periodicity transforms. Trans. Sig. Proc., 47(11):2953–2964, Nov. 1999.

[36] H. Sivaraks and C. A. Ratanamahatana. Robust and accurate anomaly detection in ecg artifacts using time series motif discovery. Computational and mathematical methods in medicine, 2015, 2015.

[37] J. M. Steele. The Cauchy-Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities. Cambridge University Press, USA, 2004.

[38] S. V. Tenneti and P. P. Vaidyanathan. Nested periodic matrices and dictionaries: New signal representations for period estimation. IEEE Trans. Signal Processing, 63(14):3736–3750, 2015.

[39] S. V. Tenneti and P. P. Vaidyanathan. Detection of protein repeats using the ramanujan filter bank. In 2016 50th Asilomar Conference on Signals, Systems and Computers, pages 343–348, Nov 2016.

[40] D. Van Aken, A. Pavlo, G. J. Gordon, and B. Zhang. Automatic database management system tuning through large-scale machine learning. In Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD '17, pages 1009–1024, New York, NY, USA, 2017. ACM.

[41] R. Vidal and P. Favaro. Low rank subspace clustering (lrsc). Pattern Recognit. Lett., 43:47–61, 2014.

[42] M. Vlachos, P. Yu, and V. Castelli. On periodicity detection and structural periodic similarity. In Proceedings of the 2005 SIAM international conference on data mining, pages 449–460. SIAM, 2005.

[43] J. Vreeken and A. Siebes. Filling in the blanks-krimp minimisation for missing data. In 2008 Eighth IEEE International Conference on Data Mining, pages 1067–1072. IEEE, 2008.

[44] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, and H. Qiao. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In Proceedings of the 2018 World Wide Web Conference, WWW '18, pages 187–196, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[45] Y. Xu, W. Yin, Z. Wen, and Y. Zhang. An alternating direction algorithm for matrix completion with nonnegative factors. Frontiers of Mathematics in China, 7(2):365–384, Apr 2012.

[46] Q. Yuan, W. Zhang, C. Zhang, X. Geng, G. Cong, and J. Han. Pred: Periodic region detection for mobility modeling of social media users. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17, pages 263–272, New York, NY, USA, 2017. ACM.

[47] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. 11 2018.

[48] $_{L}$. Zhang and P. Bogdanov. DSL: discriminative subgraph learning via sparse self-representation. In Proceedings of SIAM International Conference on Data Mining (SDM), 2019.