Period Estimation For Incomplete Time Series

1st Lin Zhang
Department of Computer Science
University at AlbanySUNY
Albany, U.S.A
lzhang22@albany.edu

2nd Petko Bogdanov
Department of Computer Science
University at AlbanySUNY
Albany, U.S.A
pbogdanov@albany.edu

***Abstract*—Natural and human-engineered systems often exhibit periodic behavior. Examples include the climate system, migration of animals in the wild, consumption of electricity in the power grid and others. The behavior of such systems, however, is not perfectly periodic. The time series we collect from them are often noisy and incomplete due to limitations of data collection and transmission, or due to sensor malfunction and outages. In addition, there are often multiple periods, for example, air temperature and pressure oscillates daily and yearly with the seasons. Hence, accurate and robust period estimation from raw time series is a fundamental task often employed in downstream applications such as traffic prediction and anomaly detection.**

In this paper, we study the period estimation problem in noisy time series with multiple periods and missing values. We propose a method based on a Ramanujan periodic dictionary and a vector completion model to estimate missing values. To account for the block structure in the Ramanujan periodic dictionary, we introduce a graph Laplacian group lasso regularization which enables robust and efficient period learning in the presence of missing observations. In our extensive experiments on datasets from diverse domains, our proposed methodology outperforms state-of-art baselines in terms of accuracy of period estimation.

***Index Terms*—Period Estimation; Nested Periodic Matrices; Missing Value Imputation; Graph Laplacian Group Lasso; Alternating Direction Method of Multipliers**

I. Introduction

Periodicity is a common pattern in time series from many natural and human-engineered systems: tandem repeats in DNA [1], seasonal animal migration [2], weather [3], electrocardiograms [4] and music composition [5] all exhibit periodic behavior. Identifying the natural periods in observed signals is a key step in understand complex system behavior and solving important prediction and outlier detection tasks [6]–[8]. As a result, the problem of period learning from signals has received a lot of attention in the fields of signal processing [9], data mining [10], databases [11], and bioinformatics [12].

The major challenges in period estimation are three-fold: (i) multiplicity (or complexity) of the underlying periods [9], (ii) noise [10], and (iii) missing values [13]. Real-world signals often exhibit more than one natural period (multiplicity) [3]. For example, electricity consumption over time is a mixture of daily, weekly and season-specific cycles [14]. In addition, the measured signals are often contaminated by measurement noise [15]. The issues of multiplicity and noise have been previously addressed by employing multi-scale basis for reconstruction (e.g. Fourier [16], wavelets [17] and other periodic bases [9]), however, the issue of missing values due to sensor malfunction [10] or communication errors in data transmission [13] has received limited attention. Incomplete data poses a key challenge for traditional period estimation methods based on Fourier Transform (FT) [16], AutoCorrelation Functions (ACF) [18] and periodograms [19], as they are all designed with the expectation of regularly sampled time series. In addition, these traditional methods often detect a large number of spurious periods, and thus, require non-trivial thresholding to determine the predicted periods [9]. These limitations make period detection in noisy signals with missing values especially challenging.

Period detection has recently been approached within a periodic dictionary framework [20]–[22]. The main idea is to approximate signals as a linear combinations of atoms in periodic dictionaries. Vaidyanathan et al. [22] proposed an efficient dictionary construction method, called *Farey dictionary*, in which periods are explicitly represented within a set of nested periodic matrices. This approach was further generalized to a family of periodic dictionaries including the *Ramanujan periodic dictionary* [9], in which period estimation is formulated as a convex optimization problem enabling optimal results and efficient inference. The effectiveness of the Ramanujan basis was demonstrated in diverse applications such as tandem repeats in DNA [1], protein repeats [12], music analysis [5] and community detection [23].

Both traditional methods and periodic dictionary-based methods do not explicitly handle missing observations. As a result, when employed on time series with missing values (imputed using various imputation techniques), they fail to recover the underlying true periods. Figure 1 demonstrates this challenge for a real-world dataset tracking daily beer consumption [24]. We omit 30% of the observations in the time series and detect the period using our proposed approach PIE, and baselines based on Fast Fourier Transform (FFT+) [16] and Nested Periodic Matrices (NPM+) [9], where missing values for baselines are imputed using spline interpolation. Our proposed method PIE not only obtains the ground truth period of 7 days, but also reports no other spurious periods in its periodogram, while alternatives are sensitive to missing values and report multiple spurious periods.

In this paper we propose a period estimation framework, called PIE, for noisy and incomplete time series. The optimization objective in our framework unifies the estimation of missing values and period estimation and is motivated

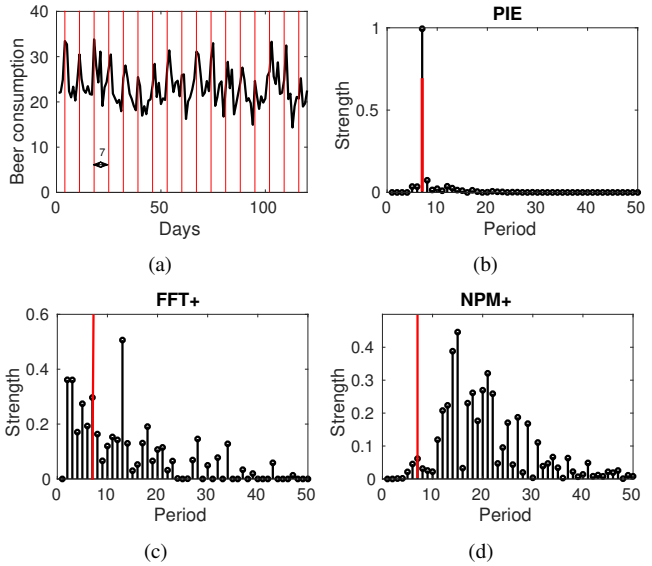


Fig. 1. Periodogram for daily beer consumption in the Czech republic [24] with 30% missing values estimated by our method PIE and baselines including NPM [9] and FFT [16]. The expected period of 7 days (or a week) is the strongest period detected by PIE, while alternatives extract multiple periods which cannot be readily interpreted.

by the co-dependence of the two tasks. Namely, a good estimation of missing observations can aid accurate period learning and vice versa. A key to the accuracy of our method is a graph Laplacian group lasso regularization designed to take advantage of the block structure in the Ramanujan periodic dictionary to improve the selectivity and robustness of our framework.

Our contributions in this paper are as follows:

(1) Novelty: To the best of our knowledge our method, called PIE, is the first framework for estimating periods from incomplete univariate time series.

(2) Robustness: We propose a novel group structure regularization for nested periodic dictionaries, leading to improved robustness and efficiency.

(3) Applicability: Extensive experiments on synthetic data and real-world datasets demonstrate that PIE exhibits better accuracy than state-of-the-art period learning methods while maintaining superior scalability.

II. Related work

Period estimation. Traditional period estimation employs Fourier transform [2], [25] to represent time series in the frequency domain and determine the periods based on the magnitude of individual frequencies. The key drawback of such approaches is the detection of a large number of spurious periods [9]. Auto-correlation is another traditional solution for period estimation [26], [27]. In addition, hybrid methods that combine Fourier transformation and auto-correlation have also been proposed for the problem [28]. All approaches from this group often rely on a predefined threshold for selecting dominant periods.

The *matrix profile* [29], [30] is another powerful tool for period estimation as it detects similarities in time series segments. The framework, however, requires reference time series for comparison and depends on prior knowledge of the size of comparison time window, which can be critical for period estimation. As reference time series and comparison window sizes are not always readily available, matrix profile methods cannot be readily applied to time series of unknown period and missing values, a setting we focus on in this paper.

The period estimation problem has been recently approached by a periodic dictionary framework by Tenneti and colleagues [9] who proposed a family of periodic dictionaries and a unified convex model for estimating periods. This approach offers significant improvements compared to earlier work as it employs orthogonality of the dictionary atoms, resulting in unique solutions [31].

Period learning from incomplete data has been limited to event sequences as opposed to general time series [32], [33]. Li et al. [32] proposed period learning in the presence of missing events by partitioning the data into short segments of predefined length and casting the period detection as an optimal alignment length for segments. Yuan et al. [33] proposed a probabilistic model for learning multiple periods in event sequences with missing values. However, these methods cannot be readily applied to general time series as they are designed for discrete binary event sequences. In addition, they assume a single period (as opposed to multiple) and rely on prior information about the event sequences, e.g., a segmentation threshold in [32].

Missing value imputation. Our primary goal is not to perform missing value imputation, a part of our model approximates missing values to recover the periodic patterns in data. General solutions for missing data imputation have been proposed to deal with incomplete matrices, such as Low-rank based matrix estimation [34], the generalized Winberg algorithm [35], and matrix profiles [36]. Matrix completion methods do not explicitly model rows/columns as temporal signals (i.e. permuting rows and columns would produce the same results from these methods) and further take advantage of multiple signals. Liu and colleagues [37] address the missing value imputation in multivariate time series using deep learning, however, they take advantage of relations among multiple time series. This is different from our task in which we focus on uni-variate time series. Another relevant method by Xie et al. [38] partitions a signal and stack segments in a matrix which is then factorized to estimate missing data. This method requires knowledge of the periodicity to partition the signal and has an underlying assumption that the signal has only one period. In [39], a non-negative Hidden Markov Model is proposed to estimate missing data by employing dictionaries learned from training data without missing values. As this method is supervised, it is not suitable for our task in which no training is available. In summary, missing value imputation techniques either rely on commonalities among multiple signals or supervision in the form of complete data, but do not address directly period learning and data imputation in univariate time series.

Deep learning methods for missing data imputation have also been proposed including recurrent architectures [40] and

adversarial learning [37]. Beyond their high complexity, these methods require large quantity of training data to learn the imputation models. Instead, we exploit periodicity directly in time series with missing values without the need for additional training data.

III. Preliminaries and Notation

In this section we introduce preliminaries and notation used in the remainder of the paper.

Definition 3.1: ($Periodic\ time\ series$) A time series $x(t)$ is *k-periodic* if there exists an integer g such that $|x(n+k) - x(n)| \leq \epsilon, \forall n$, where k is the smallest integer satisfying the above inequality and ϵ is a small real value.

Instead of using conventional methods for period estimation, such as FT, we consider a periodic dictionary based framework using *Nested Periodic Matrices (NPM)* [9].

Definition 3.2: ($Nested\ Periodic\ Matrices$) Let the integers $\{d_1, d_2, ...d_K\}$ be the divisors of d sorted in increasing order. The Nested Periodic Matrices (NPM) for a period of length d are defined as:

$$\mathbf{A}_d = [\mathbf{P}_{d_1}, \mathbf{P}_{d_2}, ...\mathbf{P}_{d_K}], \tag{1}$$

where each $\mathbf{P}_{d_i} \in \mathbb{R}^{d \times \phi(d_i)}$ is a periodic basis matrix for a component period d_i and $d = \sum_i \phi(d_i)$ where $\phi(d_i)$ denotes the Euler totient function evaluated at d_i, i.e. the number of integers between 1 and d_i that are co-prime to d_i. Columns of $\mathbf{P}_{d_i}$ are signals with period d_i. All basis matrices in the above definition are constructed based on the *Ramanujan sum*:

$$C_{d_i}(g) = \sum_{k=1, gcd(k,d_i)=1}^{d_i} e^{j2\pi kg/d_i}, \tag{2}$$

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

$$\mathbf{P}_{d_i} = \begin{bmatrix} C_{d_i}(0) & C_{d_i}(g-1) & ... & C_{d_i}(1) \\ C_{d_i}(1) & C_{d_i}(0) & ... & C_{d_i}(2) \\ ... & ... & ... & ... \\ C_{d_i}(g-1) & C_{d_i}(g-2) & ... & C_{d_i}(0) \end{bmatrix}. \tag{3}$$

The final dictionary $\mathbf{A}$ with maximum period P_{max} is formed by concatenating basis periodic matrices with periods from 1 to P_{max} as $\mathbf{A} = [\mathbf{A}_1, .., \mathbf{A}_{P_{max}}]$. A small periodic dictionary example A based on the above construction is presented in Fig. 2. It includes nested matrices for three periods $(2, 3, 5)$. The columns of this dictionary are periodically extended to the length of the input signal which in this example is 6. Further details related to the above definitions are available in [9].

The model for period estimation in complete (no missing data) signals proposed by Tenneti and colleagues [9] aims to obtain a succinct representation of an input signal through the NPM basis:

$$\underset{\mathbf{b}}{\operatorname{argmin}} \|\mathbf{Hb}\|_{Ln}, s.t.\ \mathbf{x} = \mathbf{Ab}, \tag{4}$$

where $\mathbf{x}$ is the input signal; $\mathbf{b}$ contains the coefficients specifying the representation in NPM basis; $\mathbf{H}$ is a diagonal matrix encouraging representation through small periods ($\mathbf{H}_{ii} = p^2$, where p is the period of the i-th column in $\mathbf{A}$); and Ln is a norm basis. We consider the L1 and L2 norms proposed in past work as baselines and refer to the corresponding period estimators as NPM-L1 and NPM-L2, respectively.

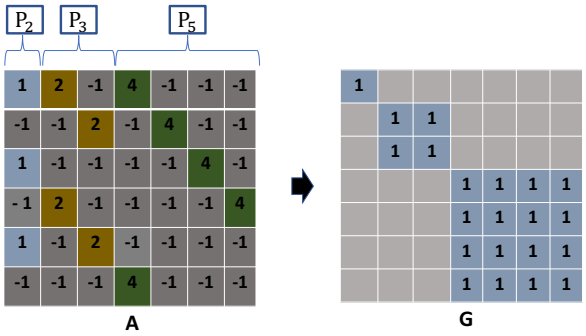


Fig. 2. An example of a Ramanujan periodic dictionary $\mathbf{A}$ (left), which has three periods as $(2, 3, 5)$ and assumes a length of the time series to encode of 6. On the right we show a block diagonal adjacency matrix $\mathbf{G}$ we employ to enforce the group structure in the dictionary. Positions corresponding to atoms of the same period have edges of weight 1 in $\mathbf{G}$ and 0 otherwise.

IV. Problem Formulation

The input for our problem is an incomplete time series $\mathbf{x} \in \mathbb{R}^T$ and a corresponding binary mask vector $\mathbf{w}$ denoting the observed time points in $\mathbf{x}$, i.e., $\mathbf{w}_i = 1$ if $\mathbf{x}_i$ is observed, and $\mathbf{w}_i = 0$ otherwise. Our goal is to estimate the periods in $\mathbf{x}$.

A. Block-aware period estimation for complete signals

We adopt the Ramanujan periodic dictionary for period estimation using a novel block-aware sparse coding model. The NPM basis and in particular the Ramanujan periodic dictionary have a group structure, namely each period is represented by $\phi(d_i)$ column atoms. Columns from the same block can collectively represent an arbitrary temporal pattern with a corresponding period of length d_i, and this block-wise dependence should be enforced when searching for a sparse reconstruction of the given signal. Note, that prior NPM models seek a sparse representation without considering this block structure.

We propose to model this dependence through a Graph Laplacian group lasso regularization on the reconstruction coefficients $\mathbf{b}$:

$$\underset{\mathbf{b}}{\operatorname{argmin}} \frac{1}{2} \sum_i \sum_j (\mathbf{b}_i - \mathbf{b}_j)^2 \mathbf{G}_{ij} = \mathbf{b}^T \mathbf{L} \mathbf{b}, \tag{5}$$

where $\mathbf{G}$ is a block matrix with elements $\mathbf{G}_{ij} = 1$ if the i-th and j-th dictionary atoms (columns) are part of the same NPM block, as shown in Fig 2. $\mathbf{L}$ is the combinatorial Laplacian matrix of $\mathbf{G}$ defined as $\mathbf{L} = \mathbf{D} - \mathbf{G}$, where $\mathbf{D}$ is a diagonal degree matrix. The block-aware objective for complete signals we propose is as follows:

$$\underset{\mathbf{b}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{Ab}\|_2^2 + \lambda_1 \|\mathbf{Hb}\|_1 + \lambda_2 \mathbf{b}^T \mathbf{L} \mathbf{b}, \tag{6}$$

where λ_1 and λ_2 are balance parameters. The first term in the objective is a data fitting function, the second term enforces sparse representation and the third term imposes the Graph

Laplacian group structure penalty. The $\mathbf{L}$ is defined same as in Eq. 4

B. Handling missing observations

When the time series is partially observed the objective from Eq. 6 is not directly applicable. To perform period estimation we introduce a complete signal proxy $\mathbf{z}$, as follows:

$$\begin{aligned}&\underset{\mathbf{z},\mathbf{b}}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{z}-\mathbf{A}\mathbf{b}\right\|_2^2+\lambda_0\left\|\mathbf{w}\odot(\mathbf{x}-\mathbf{z})\right\|_2^2\\&+\lambda_1\left\|\mathbf{H}\mathbf{b}\right\|_1+\lambda_2\mathbf{b}^T\mathbf{L}\mathbf{b},\end{aligned} \tag{7}$$

where $\odot$ is the Hadamard element-wise product. Minimizing the above objective allows us to simultaneously learn the periods and obtain estimates for the periodic component of missing values in $\mathbf{z}$. The objective also incorporates the inter-dependence between missing value imputation and period estimation: good missing value imputation will allow recovery of the true periods and similarly knowledge of the period will enable accurate missing value imputation.

V. Optimization

Since $\mathbf{H}$ is invertible, we convert the objective into standard Lasso format by introducing $\hat{\mathbf{A}}=\mathbf{A}\mathbf{H}^{-1}$ and $\boldsymbol{\beta}=\mathbf{H}\mathbf{b}$. Therefore, we can rewrite Eq. 7 as follows:

$$\begin{aligned}&\underset{\mathbf{z},\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{z}-\hat{\mathbf{A}}\boldsymbol{\beta}\right\|_2^2+\lambda_0\left\|\mathbf{w}\odot(\mathbf{x}-\mathbf{z})\right\|_2^2\\&+\lambda_1\left\|\boldsymbol{\beta}\right\|_1+\lambda_2\boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta}.\end{aligned} \tag{8}$$

Note that $\mathbf{H}$ is constant and has the same value in each block, therefore, the minimizing $\boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta}$ is equivalent to minimizing $\mathbf{b}^T\mathbf{L}\mathbf{b}$.

Since the objective in Eq. 8 is non-differentiable, it is hard to develop an optimization approach that updates $(\boldsymbol{\beta},\mathbf{z})$ simultaneously. We propose an alternating minimization method to optimize the variables in turn. To this end we partition Eq. 8 into two sub-problems w.r.t $(\boldsymbol{\beta},\mathbf{z})$, specified below:

$$\begin{cases}\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\ \frac{1}{2}\left\|\mathbf{z}-\hat{\mathbf{A}}\boldsymbol{\beta}\right\|_2^2+\lambda_1\left\|\boldsymbol{\beta}\right\|_1+\lambda_2\boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta} & (a)\\ \underset{\mathbf{z}}{\operatorname{argmin}}\ \lambda_0\left\|\mathbf{w}\odot(\mathbf{x}-\mathbf{z})\right\|_2^2+\frac{1}{2}\left\|\mathbf{z}-\hat{\mathbf{A}}\boldsymbol{\beta}\right\|_2^2 & (b)\end{cases} \tag{9}$$

Update for $\boldsymbol{\beta}$: Because $\left\|\boldsymbol{\beta}\right\|_1$ is non-differentiable, we employ the alternating direction method of multipliers (ADMM) [41] to optimize the subproblem in Eq. 9(a). We introduce an auxiliary variable $\boldsymbol{\alpha}$ and reformulate the problem as follows:

$$\begin{aligned}&\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{z}-\hat{\mathbf{A}}\boldsymbol{\beta}\right\|_2^2+\lambda_2\boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta}+\lambda_1\left\|\boldsymbol{\alpha}\right\|_1\\&s.t\ \boldsymbol{\alpha}=\boldsymbol{\beta}\end{aligned} \tag{10}$$

Following the ADMM method, we formulate the augmented Lagrangian form:

$$\begin{aligned}&L\left(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{y}\right)=\frac{1}{2}\left\|\mathbf{z}-\hat{\mathbf{A}}\boldsymbol{\beta}\right\|_2^2+\lambda_2\boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta}+\lambda_1\left\|\boldsymbol{\alpha}\right\|_1\\&+\left\langle\mathbf{y},\boldsymbol{\beta}-\boldsymbol{\alpha}\right\rangle+\frac{\rho}{2}\left\|\boldsymbol{\beta}-\boldsymbol{\alpha}\right\|_2^2,\end{aligned} \tag{11}$$

where $\mathbf{y}$ holds the corresponding Lagrangian multipliers and ρ is a penalty parameter. The updates for $(\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{y})$ at the $(k+1)$-th iteration are:

$$\begin{cases}\boldsymbol{\beta}^{k+1}=\operatorname{argmin}\ L\left(\boldsymbol{\beta},\boldsymbol{\alpha}^k,\mathbf{y}^k\right)\\ \boldsymbol{\alpha}^{k+1}=\operatorname{argmin}\ L\left(\boldsymbol{\beta}^{k+1},\boldsymbol{\alpha},\mathbf{y}^k\right)\\ \mathbf{y}^{k+1}=\mathbf{y}^k+\rho\left(\boldsymbol{\beta}^{k+1}-\boldsymbol{\alpha}^{k+1}\right)\end{cases} \tag{12}$$

Next, we present the solutions to the subproblems from Eq. 12.

$$\begin{aligned}\boldsymbol{\beta}^{k+1}&=\operatorname{argmin}\ L\left(\boldsymbol{\beta},\boldsymbol{\alpha}^k,\mathbf{y}^k\right)\\&=\operatorname{argmin}\frac{1}{2}\left\|\mathbf{z}-\hat{\mathbf{A}}\boldsymbol{\beta}\right\|_2^2+\lambda_2\boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta}\\&+\left\langle\mathbf{y}^k,\boldsymbol{\beta}-\boldsymbol{\alpha}^k\right\rangle+\frac{\rho}{2}\left\|\boldsymbol{\beta}-\boldsymbol{\alpha}^k\right\|_2^2\\&=\operatorname{argmin}\frac{1}{2}\left\|\mathbf{z}-\hat{\mathbf{A}}\boldsymbol{\beta}\right\|_2^2+\lambda_2\boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta}+\frac{\rho}{2}\left\|\boldsymbol{\beta}-\boldsymbol{\alpha}^k+\frac{\mathbf{y}^k}{\rho}\right\|_2^2\end{aligned} \tag{13}$$

By taking the gradient of $L\left(\boldsymbol{\beta},\boldsymbol{\alpha}^k,\mathbf{y}^k\right)$ w.r.t $\boldsymbol{\beta}$, we get:

$$\hat{\mathbf{A}}^T\left(\hat{\mathbf{A}}\boldsymbol{\beta}-\mathbf{z}\right)+2\lambda_2\mathbf{L}\boldsymbol{\beta}+\rho(\boldsymbol{\beta}-\boldsymbol{\alpha}^k+\frac{\mathbf{y}^k}{\rho})=0 \tag{14}$$

We, thus, obtain a closed-form solution:

$$\boldsymbol{\beta}^{k+1}=\left(\hat{\mathbf{A}}^T\hat{\mathbf{A}}+2\lambda_2\mathbf{L}+\rho\mathbf{I}\right)^{-1}\left(\hat{\mathbf{A}}^T\mathbf{z}+\rho\boldsymbol{\alpha}^k-\mathbf{y}^k\right) \tag{15}$$

The update of $\boldsymbol{\alpha}^{k+1}$ is the following problem:

$$\begin{aligned}\boldsymbol{\alpha}^{k+1}&=\operatorname{argmin}\ L\left(\boldsymbol{\beta}^{k+1},\boldsymbol{\alpha},\mathbf{y}^k\right)\\&=\operatorname{argmin}\lambda_1\left\|\boldsymbol{\alpha}\right\|_1+\left\langle\mathbf{y},\boldsymbol{\beta}^{k+1}-\boldsymbol{\alpha}\right\rangle+\frac{\rho}{2}\left\|\boldsymbol{\beta}-\boldsymbol{\alpha}\right\|_2^2\\&=\operatorname{argmin}\lambda_1\left\|\boldsymbol{\alpha}\right\|_1+\frac{\rho}{2}\left\|\boldsymbol{\beta}^{k+1}-\boldsymbol{\alpha}+\frac{\mathbf{y}^k}{\rho}\right\|_2^2\end{aligned} \tag{16}$$

This subproblem can be solved by soft thresholding [42] based on the following Lemma 1.

Lemma 1: If $\lambda>0$, then

$$\operatorname{argmin}_{\mathbf{u}}\frac{1}{2}\left\|\mathbf{u}-\mathbf{v}\right\|_2^2+\lambda\left\|\mathbf{u}\right\|_1$$

has an element-wise closed form solution:

$$\mathbf{u}_i=max\left\{|\mathbf{v}_i|-\lambda,0\right\}\times sign\left(\mathbf{v}_i\right) \tag{17}$$

where $sign\left(r\right)$ is the signum function, defined as

$$sign\left(r\right)=\begin{cases}+1 & \text{if } r>0\\0 & \text{if } r=0\\-1 & \text{if } r<0\end{cases} \tag{18}$$

As a result, the i-th element of $\boldsymbol{\alpha}$ has a closed-form solution

$$\boldsymbol{\alpha}_i^{t+1}=max\left\{|\mathbf{v}_i|-\frac{\lambda_1}{\rho},0\right\}\cdot sign\left(\mathbf{v}_i\right),$$

where $\mathbf{v}=\boldsymbol{\beta}^{k+1}+\frac{\mathbf{y}^k}{\rho}$.

Update for $\mathbf{z}$: Taking the gradient of Eq. 9(b) w.r.t $\mathbf{z}$ and setting it to 0, we obtain:

$$2\lambda_0\mathbf{w}\odot(\mathbf{z}-\mathbf{x})+\left(\mathbf{z}-\hat{\mathbf{A}}\boldsymbol{\beta}\right)=0. \tag{19}$$

Thus, we have a closed-form solution for $\mathbf{z}$:

$$\mathbf{z} = \left(2\lambda_0 \mathbf{w} \odot \mathbf{x} + \hat{\mathbf{A}}\boldsymbol{\beta}\right) \oslash (\mathbf{1} + 2\lambda_0 \mathbf{w}), \tag{20}$$

where $\mathbf{1} \in \mathbb{R}^T$ is a vector of all ones and $\oslash$ denotes element-wise division.

Algorithm 1: PIE

1 **Input:** A time series $\mathbf{x}$ and parameters $(\lambda_0, \lambda_1, \lambda_2)$
2 **Initialize:** $\mathbf{z} = \mathbf{x}, \mathbf{y} = 0, \rho = 1$
3 $\mathbf{Q} = \left(\hat{\mathbf{A}}^T\hat{\mathbf{A}} + 2\lambda_2\mathbf{L} + \rho\mathbf{I}\right)^{-1}$
4 **while** *not converged* **do**
5 **while** *not converged* **do**
6 $\boldsymbol{\beta}^{k+1} = \mathbf{Q}\left(\hat{\mathbf{A}}^T\mathbf{z} + \rho\boldsymbol{\alpha}^k - \mathbf{y}^k\right)$
7 $\boldsymbol{\alpha}_i^{k+1} = max\left\{|\mathbf{v}_i| - \frac{\lambda_1}{\rho}, 0\right\} \cdot sign\left(\mathbf{v}_i\right)$
8 $\mathbf{y}^{k+1} = \mathbf{y}^k + (\boldsymbol{\beta}^{k+1} - \boldsymbol{\alpha}^{k+1})$;
9 $\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^k$
10 **end**
11 $\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^{k+1}$;
12 $\mathbf{z}^{t+1} = \left(2\lambda_0\mathbf{w} \odot \mathbf{x} + \hat{\mathbf{A}}\boldsymbol{\beta}^{t+1}\right) \oslash (\mathbf{1} + 2\lambda_0\mathbf{w})$;
13 $t \leftarrow t + 1$;
14 **end**
15 $\mathbf{b} = \mathbf{H}^{-1}\boldsymbol{\alpha}$;
16 **Output:** Complete time series $\mathbf{z}$, sparse coefficients $\mathbf{b}$

A. Overall algorithm and complexity

We summarize the overall algorithm of our method for *Period Estimation and Imputation (PIE)* in Algorithm. 1. The steps of the algorithm alternate between updating the two variables $\boldsymbol{\beta}$ and $\mathbf{z}$, based on the updates derived in the previous subsection. Finally, to obtain predicted periods, we add the coefficients learned in $\mathbf{b}$ in the columns corresponding to dictionary atoms representing a given period. Top periods rank highest in terms of aggregate dictionary loadings.

There are two main loops in the algorithm: an inner loop to update $\boldsymbol{\beta}$ and an outer loop which iterates between the two variables. The elements of $\boldsymbol{\beta}$ are updated in steps 6-9 and the complexity of these steps is dominated by step 6 since Steps 7-9 are all linear in the size of the input T. Updating $\boldsymbol{\beta}$ (Step 6) involves a matrix multiplication, which has complexity of $\Theta(gT)$, where g is the total number of columns of $\hat{A}$. The complete signal $\mathbf{z}$ is updated at step 12 and has a complexity of $\Theta(gT)$. Note that the complexity of step 3 is $O(g^3)$, but this inversion is computed only once and re-used after that. In addition, this step is data agnostic and can be pre-computed and re-used for the analysis of different time series. Thus, the overall complexity of one iteration of the main loop is $\Theta(gT)$.

VI. Experimental Evaluation

We evaluate our method PIE on 3 different tasks: 1) period estimation, 2) missing value imputation and 3) future value prediction. We use state-of-the-art baselines for comparison in each of them and perform experiments on both synthetic and real-world datasets.

A. Datasets

We use the following datasets for evaluation:

Synthetic data: We generate periodic time series by following the experimental setup in [9]. We set the series length to $T = 200$ and ground truth (GT) periods to $[3, 7, 11]$. Then, we add random noise and remove observations at random times (missing values).

Web traffic [43]: This dataset tracks the number of daily views of Wikipedia articles between 7/2015 and 12/2016. We follow the process in [44] to aggregate data at a daily time scale based on the languages of articles (there are 7 languages in the dataset). We report performance on the German time series, however, all languages exhibit similar performance. For this data, we expect "natural" periods of a week, or a month.

Bike rentals [45]: This dataset contains time series of rental logs of bikes from 2011 to 2012 in the Capital Bikeshare system, Washington D.C., USA. Bike rentals are likely to be correlated with weather conditions as well as weekly patterns of human movement. For instance, temperature, precipitation, season, day of the week, and hour in the day are likely to affect rental behaviors. In particular, we expect a weekly periodicity due to work-week patterns, such as riding to work or primarily for leisure when time is available on weekends.

Sunspots [46]: This dataset contains the records of sunspot activity between 1/1749 and 8/2017. A sunspot is a phenomenon in which a region of the sun's photo-sphere becomes darker than its surrounding area. The mechanism behind sunspot formation is driven by concentrations of magnetic field flux leading to a periodic pattern which repeats approximately every 11 years (132 months) according to the solar cycle.

Beer consumption [24]: This dataset includes beer consumption from the Czech Republic. Beer consumption is often related to daily or weekly activity, for instance after-work socialization or larger social events on weekend days. As such, we expect a weekly period.

Air Quality [47]: This hourly dataset contains the PM2.5 measurements for five cities in China spanning the period between 2010 and 2015. PM2.5 levels measure small (less than 2.5nm) airborne particulate matter, 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) [48]. In this experiment, we report the results based on data from Jingan, Shangai, China.

Hourly weather [49]: This data includes hourly weather measurements for 5 years from multiple cities. We evaluate period estimation on the pressure time series as it is directly affected by the solar position and one can expect a half day (12 hours) or full day (24 hours) GT periods in this dataset. We report in experiments the time series for Atalanta city, however, results from other cities are similar.

A note on expected periods: Explicit ground truth periods are not available for many of the above datasets. Since some represent human activity we intuitively expect "natural" periods - days, weeks, months, years. Other hard-to-explain

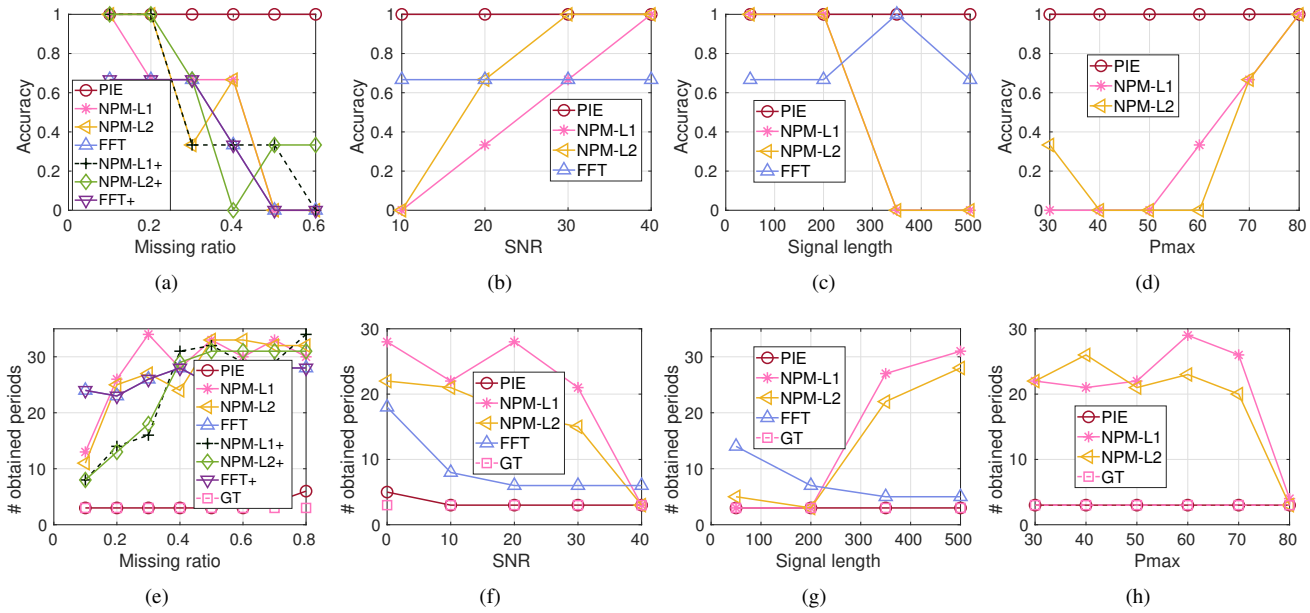


Fig. 3. Period estimation comparison of the competing techniques on synthetic datasets. In the first column, we show results for varying missing ratio; the second column shows the results for varying SNR; the third column shows the results for varying time series length; the last column shows the results of varying Pmax for LP, NPM-L1, and NPM-L2.

periods detected by competing techniques are likely spurious as they cannot be interpreted as driven by phenomena from the respective domains.

B. Experimental settings

Baselines: Since we consider two different tasks, period estimation and data imputation, we adopt two corresponding groups of baselines for comparison. For period estimation, we compare PIE to NPM-L1 [9], NPM-L2 [9] and FFT [16]. For this evaluation, all missing values are filled in with zeros. For missing value imputation, we compare to linear interpolation [50], spline interpolation [50], and LMF [38] using the actual ground truth periods for partitioning the time series. In addition, we combine period estimation methods with spline interpolation, which is the best missing value imputation method among baselines, to obtain NPM-L1+, NPM-L2+ and FFT+. This is to reduce the negative impact of the missing values for these period estimation methods.

Evaluation Metrics: We compute the accuracy of detecting ground truth (GT) periods for the period estimation task. As methods often detect multiple periods, we sort the obtained periods according to their magnitudes and use the top-k ranked periods, where k is the number of GT periods. We use *Sum of Squared Errors (SSE)* to compare alternative approaches on missing value estimation, forecasting and parameter evaluation.

Implementation: All methods are implemented and executed in Matlab 2018b. All reported running times are based on single-core execution on an Intel(R) Core i7-8700 CPU @ 3.20GHz processor in a Dell desktop.

C. Period estimation

1) Performance on synthetic data: We compare PIE with other period estimation baselines on synthetic data by varying three parameters: signal-to-noise ratio (SNR), missing value ratio (the fraction of unavailable observations) and the signal length T in Fig. 3. PIE has perfect (100%) accuracy for all missing ratios between 0 and 0.6 (Fig. 3(a)). NPM-L1 and NPM-L2 achieve 100% only when the missing ratio is below 0.1 and 0.2, respectively. FFT is hihgly sensitive to missing values and is incapable of detecting the true period even when a small fracion of the measurements are missing. In addition, two-step methods: NPM-L1+, NPM-L2+ and FFT+, which use spline interpolation and then estimation of periods, do not have significantly better performance then their no-interpolation counterparts since the imputation of missing values is period-agnostic and does not preserve the periodicity in signals. In contrast, PIE maintains a high quality as it jointly imputes missing values and estimates the period, allowing it to consistently uncover the true periodicity in data.

In Fig. 3(b), we show performance for time series with no missing values and varying SNR. PIE similarly achieves 100% accuracy at all SNR levels while FFT peaks at 67%. Both NPM methods are sensitive to noise and as a result perform optimally only at high SNR, namely NPM-L1 and NPM-L2 reach 100% at $SNR = 30$ and above. PIE is more robust to noise compared to other Ramanujan dictionary methods (NPM methods) as it models explicitly the block structure in the periodic dictionary via the graph Laplacian group lasso. Both NPM methods, however, neglect this structure.

Next we evaluate period estimation for varying the observation length of the time series. Ideally, a method should be able to detect the periodicity employing as little data as possible.

	Statistics		PIE			NPM-L1 [9]		NPM-L2 [9]		FFT [16]	
Dataset	*Length*	*GT Periods*	Results	Time	$[\lambda_0, \lambda_1, \lambda_2]$	Results	Time	Results	Time	Results	Time
Web traffic	550	[**7**] days	[**7**]	0.3s	$[1, 1e^{-3}, 1e^{-3}]$	[18]	52s	[18]	0.2s	[49]	0.04s
Bike rental	731	[**7**] days	[**7**]	0.2s	$[1, 1e^{-3}, 1e^{-3}]$	[20]	31s	[2]	0.2s	[26]	0.04s
Sunspots	2820	[**132**] months	[**132**]	8.2s	$[1, 1e^{-4}, 1e^{-2}]$	[150]	1467.7s	[99]	4.3s	[128]	0.04s
Beer consumption	365	[**7**]days	[**7**]	0.3s	$[1, 1e^{-3}, 1e^{-2}]$	[18]	15.3s	[15]	0.2s	[**7**]	0.04s
Air quality	25k	[**12**, **24**] hours	[**12**, **24**]	2.2s	$[1, 1e^{-4}, 1e^{-4}]$	[1,2]	3.87 hours	[3, 4]	156.3s	[46, 49]	0.05s
Hourly weather	45k	[**12**, **24**] hours	[**12**, **24**]	4.5s	$[1, 1e^{-4}, 1e^{-4}]$	/	/	/	/	[7, 31]	0.08s

TABLE I

REAL-WORLD DATASET STATISTICS, COMPARISON OF PERIODS ESTIMATION AND RUNNING TIME FOR OUR METHOD AND COMPETITORS.

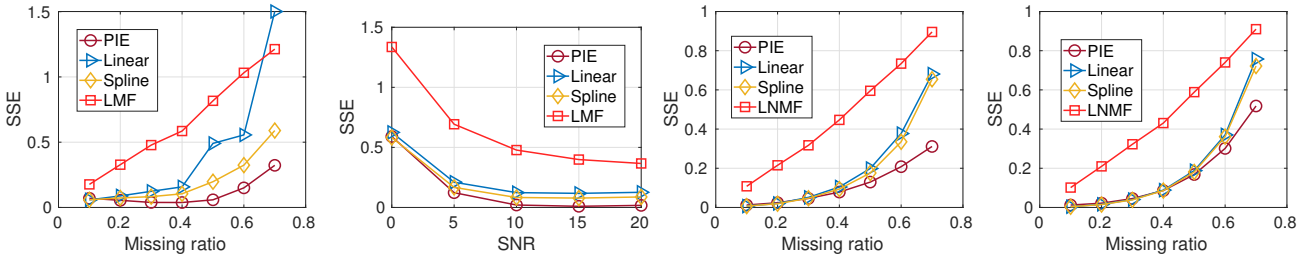


Fig. 4. Evaluation of the estimated missing values: (a) and (b) present the SSE for varying missing ratio and SNR on synthetic data; (c) and (d) present the results for varying missing ratio on real-world datasets: Beer consumption and Web traffic data.

PIE retains 100% accuracy when varying time series length from 100 to 500. FFT also maintains consistent, although lower than PIE's, accuracy when varying the length of the time series. The performance of both NPM methods drops from 100% to 0 when increasing the observation length since they are sensitive to the choice of $Pmax$ (i.e. the largest period in the dictionary), and thus cause identifiability issues when the observation length of the time series varies. In other words, for time series of fixed periods, NPM methods predict different periods for different observation lengths.

We also investigate the impact of varying the maximum candidate period $Pmax$ for Ramanujan dictionary methods in Fig. 3(d) (observation length is fixed to 200). PIE is not sensitive to the selected value of $Pmax$ and consistently obtains 100% accuracy for estimating GT periods. However, NPM-L1 and NPM-L2 achieve 100% accuracy only when the maximum considered period is relatively high $Pmax = 80$ (i.e. they require larger dictionary, and thus, higher computational cost). The advantage of PIE is due to the proposed Graph Laplacian group lasso regularization that overcomes the identifiability issue. As a result our Laplacian regularization not only increases the accuracy, but also reduces the computational cost. The periodic dictionary contains 278 columns when $Pmax = 30$, and 1966 when $Pmax = 80$. Hence, the ability to obtain high quality with smaller $Pmax$ significantly reduces the computational cost for detection.

In all synthetic experiments, PIE not only exhibits better accuracy, but also predicts fewer periods overall, making it highly specific. To quantify the number of spurious periods for all methods, we normalize the obtained period weight vector of each method to one and set a threshold of 0.05 as a cut-off for detected periods. In Figs. 3(e), 3(g), 3(f) and 3(h), we present the number of detected periods for each setting. Typically PIE predicts the same number of periods as those in the GT, while baselines predict multiple spurious periods. This property is very important when the number of GT periods is unknown because one can determine true periods simply based on the results of PIE. Without certain prior information to use in post-processing, we can't directly use the results of other period estimation methods.

2) Performance on real-world data: We also demonstrate the performance of PIE on a variety of real-world datasets. We summarize period detection results in Table. I. PIE is able to detect periods matching the expected true periods in each of the datasets. In comparison, competing methods mostly detect difficult to justify or incorrect periods. While FFT is the fastest method in terms of running time, it correctly detects the GT period in the Beer dataset only. Both NPM-L1 and NPM-L2 miss the GT periods because real-world datasets are noisy, and thus, not perfectly periodic. To make sure that NPM-L1 and NPM-L2 have optimal performance, we set $Pmax = 90$ for this evaluation. Nevertheless, both methods fail to detect the true periodicity while also incurring high computational cost.Note that some results for NPM-L1 and NPM-L2 are missing for the Air quality and Hourly weather datasets because NPM methods do not scale to such large datasets.

D. Missing value estimation study

Apart from period estimation, we can also employ PIE to estimate missing values in periodic time series. In this section we report the performance on missing value estimation measured in terms of SSE. We present the performance for varying SNR and missing ratio for synthetic data in Fig. 4(a) and Fig. 4(b). PIE outperforms competitors on value imputation as it considers periodicity in the time series when

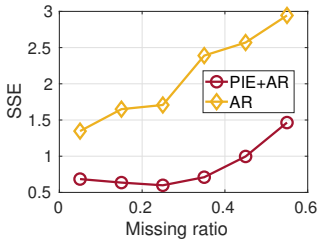
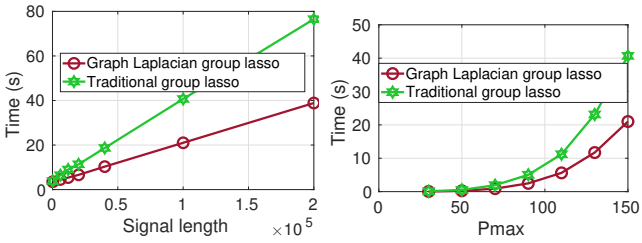


Fig. 5. Total SSE for forecasting 2000 future time points after training on 8000 employing Autoregression (AR) and Autoregression with periodicity adjustment via PIE (PIE+AR).



(a) (b)

Fig. 6. Evaluation of the running time of Graph Laplacian group lasso and the traditional group lasso solver by varying the length of the signal (a) and the maximum candidate period Pmax (b). We fix the $Pmax = 150$ for (a) and the signal's length to $100k$ for (b).

predicting missing values. LMF partitions the signal at a fixed window length and has the worst prediction in these multiple-period datasets. Linear interpolation degrades faster than all alternatives with increasing ratio of missing values as it has access to diminishing local temporal information. Spline interpolation is the closest competitor to PIE for missing value imputation, however, it does not take advantage of periodic behavior.

We also fix the missing value ratio at 0.3, and evaluate the competing methods at varying SNRs in Fig. 4(b). PIE still outperforms alternatives because of its noise robustness due to the proposed group regularization. Linear and spline interpolation obtain similar performance as at this level of missing values they are able to recover the signal based on local approximation. In particular, the performance of all methods tend to be stable when SNR is higher than 10. This is because the amount of missing values becomes a significant factor in estimation accuracy when the noise is low. LMF again has the worst performance among competitors since noise compromises its low-rank assumption.

We also present value imputation results on real-world datasets, including Beer consumption and Web traffic data, in Fig. 4(c) and Fig. 4(d). Similar to the observed behavior on the synthetic data, our method dominates alternatives for the entire range of missing value ratios. In both datasets, LMF results in the highest SSE, while linear and spline interpolation achieve similar quality that of PIE at low missing ratios and diverges when more than 50% of the observations are missing.

E. PIE for forecasting in time series.

The periodic dictionary encoding learned by PIE can be also employed to improve forecasting (or future value prediction) by considering seasonality or periodicity. We set out to test this hypothesis in this subsection. A general time series with trend and periodicity can be decomposed into $x = x_{per} + x_{tr} + \epsilon$, where in x_{per} is its periodic component, x_{tr} is an long-term trend, and ϵ is a noise component. Our hypothesis is that when predictive methods, such as auto-regression (*AR*), are used for forecasting their accuracy critically depends on accounting for periodicity in the data. Hence, we set out to evaluate this hypothesis using AR as an example predictor, though other methods can also be similarly employed.

In this experiment, we generate periodic multivariate time series with total length of $T = 10000$. We use the first 8000 points for training and the remainder for testing. The baseline model we employ is a 100-lag vector *AR* model, which is used to forecast the final 2000 time points directly. In particular, we replace missing values with the mean of the stationary time series. In comparison, we employ PIE to get the periodic component $\mathbf{z}$ as well as the periodic structure coefficient $\boldsymbol{\beta}$. The same *AR* model is then used to fit the trend without seasonality, i.e. perform AR on $\mathbf{x} - \mathbf{z}$. Next, the forecasting from this *AR* model is added to the projected periodic component from $\bar{\mathbf{A}}\boldsymbol{\beta}$, where $\bar{\mathbf{A}}$ is obtained by extending the periodic dictionary to the final 2000 time points that we predict. We repeat this procedure for multiple missing ratios, and present the results in Fig. 5. Removing correctly-identified periodic components from the time series, as detected by PIE, leads to more accurate forecasting even over long prediction horizons, particularly in the presence of missing values. The final results are obtained by averaging 10 runs for each missing ratio.

F. Importance of the graph Laplacian group lasso.

In this section, we demonstrate the efficiency of our proposed graph Laplacian group lasso. Recall that we propose the graph Laplacian group sparse model to enforce the group structure in the dictionary. This allows us to optimize all groups together, while the traditional group lasso model needs to solve one group at a time [41], [51]. To validate the utility of our proposed regularization we compare it with the traditional group lasso by formulating two alternative versions for our model as follows:

$$\begin{cases} \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \ \frac{1}{2}\left\|\mathbf{y} - \hat{\mathbf{A}}\boldsymbol{\beta}\right\|_2^2 + \lambda \sum_i^N \left\|\boldsymbol{\beta}_{G_i}\right\|_2 & (1) \\ \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \ \frac{1}{2}\left\|\mathbf{y} - \hat{\mathbf{A}}\boldsymbol{\beta}\right\|_2^2 + \lambda \boldsymbol{\beta}^T \mathbf{L} \boldsymbol{\beta}, & (2) \end{cases} \tag{21}$$

where we integrate the traditional group lasso and our proposed graph Laplcacian group lasso in Eq.21 (1) and (2), respectively. Note that $\mathbf{y}$ is a time series; N denotes the number of groups in the dictionary $\hat{\mathbf{A}}$; and λ is a balance parameter.

To optimize Eq.21 (1), we employ a state-of-the-art solver of group lasso [41] which employs the ADMM framework. We can obtain a closed form solution for Eq.21 (2) by setting

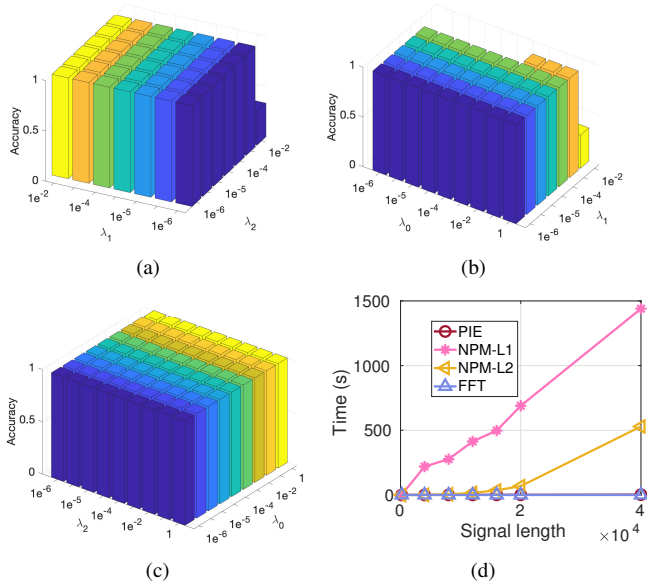


Fig. 7. The sensitivity analysis of the parameters on the synthetic data and running time (in seconds).

$L(\boldsymbol{\beta}) = \frac{1}{2}\left\|\mathbf{y} - \hat{\mathbf{A}}\boldsymbol{\beta}\right\|_2^2 + \lambda\boldsymbol{\beta}^T\mathbf{L}\boldsymbol{\beta}$, we take the gradient of $L(\boldsymbol{\beta})$ w.r.t $\boldsymbol{\beta}$ as zero, therefore, we have

$$\hat{\mathbf{A}}^T\left(\hat{\mathbf{A}}\boldsymbol{\beta} - \mathbf{y}\right) + 2\lambda\mathbf{L}\boldsymbol{\beta} = 0 \tag{22}$$

Then, we can obtain the closed-form solution as

$$\boldsymbol{\beta}^* = \left(\hat{\mathbf{A}}^T\hat{\mathbf{A}} + 2\lambda\mathbf{L}\right)^{-1}\hat{\mathbf{A}}^T\mathbf{y} \tag{23}$$

With our group lasso model, we not only obtain a closed form solution, i.e the optimal solution, but also one that is faster than the traditional group lasso solver. We show the running time of our proposed model and the traditional model in Fig. 6. When increasing the size of $Pmax$ and the signal's length, the running time gap between our approach and traditional lasso is gradually increasing.

G. Scalability and parameter sensitivity analysis

We evaluate the CPU running time of competing period learning methods on the synthetic data in Fig. 7(d). All methods are implemented in MATLAB without parallel execution. Note that the code of NPM-L1 and NPM-L2 are provided by the authors of [9]. For PIE, NPM-L1 and NPM-L2, we fix $Pmax = 50$. For varying signal lengths, the running time of PIE is close to that of FFT, taking only a few seconds to run on a signal of length $T = 40k$ (Fig. 7(d)). Although NPM-L2 has a closed-form solution, it is slower than PIE and FFT. When the length of the signal is $40k$, NPM-L2 needs about 9 minutes and NPM-L1 need about 25 minutes to complete, while PIE requires 2 seconds.

We also investigate the sensitivity of PIE to its parameters on synthetic data, including λ_0, λ_1, and λ_2 for missing value modeling, sparsity and group structure, respectively. In these experiments, we fix one parameter and vary the other two. We report the period detection accuracy in Fig. 7. PIE is not sensitive to the parameters within wide ranges. When λ_1 and λ_2 are set close to 1, the performance of our model decreases significantly. This is because large λ_1 and λ_2 promote the selection of very sparse blocks in the periodic dictionary, limiting the correct reconstruction of the input data and leading to inaccurate period detection. Small λ_1 and λ_2 result in good performance in practice. Note that this is also confirmed across real-world datasets: we report the optimal parameter values in the 6-th column of Table I. In particular, we fix these parameters when varying the missing ratio and the results demonstrate robustness.

VII. Conclusion

In this work, we introduced a robust framework for period estimation in signals with missing values. We formulated the period estimation and missing values imputation as a joint optimization problem. The demonstrated the superior performance of our method over state-of-the-art baselines by conducting extensive experiments on both challenging synthetic datasets and multiple real-world datasets. PIE was able to estimate the true periods with 100% accuracy even when 60% 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] S. V. Tenneti and P. P. Vaidyanathan, "Detecting tandem repeats in dna using ramanujan filter bank," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2016, pp. 21–24.

[2] 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*. ACM, 2010, pp. 1099–1108.

[3] Q. Yuan, J. Shang, X. Cao, C. Zhang, X. Geng, and J. Han, "Detecting multiple periods and periodic patterns in event time sequences," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: ACM, 2017, pp. 617–626. [Online]. Available: http://doi.acm.org/10.1145/3132847.3133027

[4] T. Tsalaile, R. Sameni, S. Sanei, C. Jutten, J. Chambers *et al.*, "Sequential blind source extraction for quasi-periodic signals with time-varying period," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 3, pp. 646–655, 2008.

[5] H. Liao and L. Su, "Monaural source separation using ramanujan subspace dictionaries," *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1156–1160, Aug 2018.

[6] 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*, ser. KDD '10. New York, NY, USA: ACM, 2010, pp. 1099–1108. [Online]. Available: http://doi.acm.org/10.1145/1835804.1835942

[7] T. G. Andersen and T. Bollerslev, "Intraday periodicity and volatility persistence in financial markets," *Journal of Empirical Finance*, vol. 4, no. 2, pp. 115 – 158, 1997, high Frequency Data in Finance, Part 1. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0927539897000042

[8] T. Takahashi, B. Hooi, and C. Faloutsos, "Autocyclone: Automatic mining of cyclic online activities with robust tensor factorization," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2017, pp. 213–221. [Online]. Available: https://doi.org/10.1145/3038912.3052595

[9] S. V. Tenneti and P. P. Vaidyanathan, "Nested periodic matrices and dictionaries: New signal representations for period estimation," *IEEE Trans. Signal Processing*, vol. 63, no. 14, pp. 3736–3750, 2015. [Online]. Available: https://doi.org/10.1109/TSP.2015.2434318

[10] Z. Li and J. Han, *Mining Periodicity from Dynamic and Incomplete Spatiotemporal Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 41–81. [Online]. Available: https://doi.org/10.1007/978-3-642-40837-3_2

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

[12] 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*, Nov 2016, pp. 343–348.

[13] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "Brits: Bidirectional recurrent imputation for time series," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 6775–6785. [Online]. Available: http://papers.nips.cc/paper/7911-brits-bidirectional-recurrent-imputation-for-time-series.pdf

[14] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International Journal of Forecasting*, vol. 30, no. 4, pp. 1030 – 1081, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207014001083

[15] W. Zhu, S. M. Mousavi, and G. C. Beroza, "Seismic signal denoising and decomposition using deep neural networks," *arXiv preprint arXiv:1811.02695*, 2018.

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

[17] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 674–693, 1989.

[18] C. Berberidis, W. G. Aref, M. Atallah, I. Vlahavas, and A. K. Elmagarmid, "Multiple and partial periodicity mining in time series databases," in *Proceedings of the 15th European Conference on Artificial Intelligence*, ser. ECAI'02. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2002, pp. 370–374. [Online]. Available: http://dl.acm.org/citation.cfm?id=3000905.3000984

[19] N. R. Lomb, "Least-squares frequency analysis of unequally spaced data," *Astrophysics and Space Science*, vol. 39, no. 2, pp. 447–462, Feb 1976. [Online]. Available: https://doi.org/10.1007/BF00648343

[20] M. Nakashizuka, "A sparse decomposition for periodic signal mixtures," 08 2007, pp. 627 – 630.

[21] D. Shi-Wen and J. Han, "Ramanujan subspace pursuit for signal periodic decomposition," *Mechanical Systems and Signal Processing*, vol. 90, 12 2015.

[22] P. P. Vaidyanathan and P. Pal, "The farey-dictionary for sparse representation of periodic signals," 05 2014, pp. 360–364.

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

[24] "Daily beer consumption, howpublished = https://www.kaggle.com/dongeorge/beer-consumption-sao-paulo/downloads/beer-consumption-sao-paulo.zip/2, note = Published: 2019."

[25] P. Indyk, N. Koudas, and S. Muthukrishnan, "Identifying representative trends in massive time series data sets using sketches," in *VLDB*, 2000, pp. 363–372.

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

[27] S. Parthasarathy, S. Mehta, and S. Srinivasan, "Robust periodicity detection algorithms," in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, ser. CIKM 06. New York, NY, USA: Association for Computing Machinery, 2006, p. 874875.

[28] 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*. SIAM, 2005, pp. 449–460.

[29] C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, "Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Dec 2016, pp. 1317–1322.

[30] Y. Zhu, Z. Zimmerman, N. S. Senobari, C. M. Yeh, G. Funning, A. Mueen, P. Brisk, and E. Keogh, "Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Dec 2016, pp. 739–748.

[31] W. Sethares and T. Staley, "Periodicity transforms," *Trans. Sig. Proc.*, vol. 47, no. 11, pp. 2953–2964, Nov. 1999. [Online]. Available: https://doi.org/10.1109/78.796431

[32] Z. Li, J. Wang, and J. Han, "eperiodicity: Mining event periodicity from incomplete observations," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1219–1232, 2015. [Online]. Available: https://doi.org/10.1109/TKDE.2014.2365801

[33] 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*, ser. WSDM '17. New York, NY, USA: ACM, 2017, pp. 263–272.

[34] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, p. 717, Apr 2009. [Online]. Available: https://doi.org/10.1007/s10208-009-9045-5

[35] A. Eriksson and A. V. D. Hengel, "Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l1 norm," 2012.

[36] 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*, vol. 20, no. 1, pp. 40–49, 3 2011.

[37] Y. Liu, R. Yu, S. Zheng, E. Zhan, and Y. Yue, "NAOMI: non-autoregressive multiresolution sequence imputation," *CoRR*, vol. abs/1901.10946, 2019. [Online]. Available: http://arxiv.org/abs/1901.10946

[38] C. Xie, A. Tank, A. Greaves-Tunnell, and E. Fox, "A Unified Framework for Long Range and Cold Start Forecasting of Seasonal Profiles in Time Series," *arXiv e-prints*, p. arXiv:1710.08473, Oct 2017.

[39] J. Han, G. J. Mysore, and B. Pardo, "Audio imputation using the non-negative hidden markov model," in *Lecture Notes in Computer Science: Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2012, pp. 347–355.

[40] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "BRITS: bidirectional recurrent imputation for time series," *CoRR*, vol. abs/1805.10572, 2018. [Online]. Available: http://arxiv.org/abs/1805.10572

[41] 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.*, vol. 3, no. 1, pp. 1–122, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1561/2200000016

[42] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2013.

[43] "Wikipedia Traffic Data Exploration, howpublished = https://www.kaggle.com/c/web-traffic-time-series-forecasting/data, note = Published: 2017."

[44] "Wikipedia Traffic Data Exploration process, howpublished = https://www.kaggle.com/muonneutrino/wikipedia-traffic-data-exploration, note = Published: 2017."

[45] H. Fanaee-T and J. Gama, "Event labeling combining ensemble detectors and background knowledge," *Progress in Artificial Intelligence*, pp. 1–15, 2013. [Online]. Available: http://dx.doi.org/10.1007/s13748-013-0040-3

[46] "Monthly Mean Total Sunspot Number, howpublished = https://www.kaggle.com/robervalt/sunspots, note = Published: 2018."

[47] 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.

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

[49] "Historical Hourly Weather Data 2012-2017, howpublished = https://www.kaggle.com/selfishgene/historical-hourly-weather-data, note = Published: 2017-12-27."

[50] A. Gnauck, "Interpolation and approximation of water quality time series and process identification," *Analytical and bioanalytical chemistry*, vol. 380, pp. 484–92, 11 2004.

[51] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS*, 2013.