PERCeIDs: PERiodic CommunIty Detection

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

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

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

Abstract*—Many complex networked systems, both natural and human-made, exhibit periodic behavior driven by underlying seasonal processes: election cycles and regular sporting events in social networks, cell cycle phases in gene networks, and load variation in infrastructure networks due to weather or daylight patterns. The “natural” periodicity may vary across network communities. At the same time this periodic community behavior is central to (i) understating the overall system dynamics and (ii) for detection of the communities themselves. The predominant approach to dynamic community detection first detects communities and then as a second step quantify seasonality in their activity. *How to jointly detect communities and their inherent periodicity, while also accounting for non-periodic one-off events?

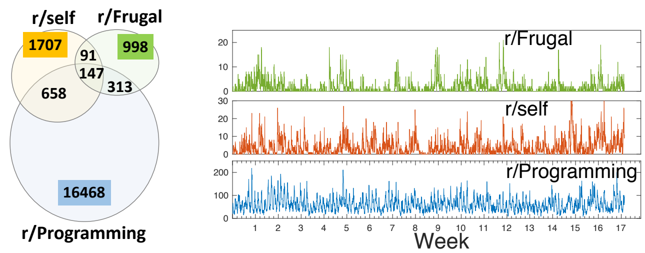
We propose PERCeIDs, a framework for periodic overlapping community detection from temporal interaction data. We model observed pairwise interaction activity as a mixture of periodic and outlier (non-periodic) components. We explicitly enforce periodic structure within our model by learning a succinct *Ramanujan basis dictionary* for community behaviors. By explicitly modeling periodicity, PERCeIDs outperforms baselines on both detecting highly overlapping communities with up to 2-fold improvement in NMI compared to state-of-the-art baselines, while offering an interpretable temporal structure for discovered communities in the dataset. Implementation of our method is available for download [64].

***Index Terms*—Dynamic networks, Community detection, Periodicity detection, Tensor factorization, Optimization.**

I. Introduction

Periodicity is a common pattern in natural networks: animal mobility [37], neuron activation in the brain [10] and pathway expression during the cell cycle [50] all exhibit periodic behavior. Similar patterns have been observed in social [34], [35] and infrastructure [30] networks as well. This periodic behavior arises in connected subnetworks which share (i) functions in biological networks: gene or neural pathways; (ii) interests in social networks: fans of a football team or a celebrity; or (iii) geographical locality in infrastructure networks: neighborhoods, towns etc. In light of this, *How can we identify communities in dynamic networks which share both membership and periodic temporal behavior?*

Community detection in static networks has a long research history [20], [54], [62], [63], however, methods in this category do not consider the temporal information of interactions. Evolutionary clustering [9], [31], [46] considers the evolution of the community membership itself and methods in this group exhaustively partition every snapshot into non-overlapping clusters. More recent approaches for



(a) Overlap (b) Time series

Fig. 1. (a) Venn diagram of the sets of Reddit users in three subreddit channels and (b) hourly interaction time series within the subredit channels spanning January-April, 2009.

dynamic networks assume stationary community membership and persistent activity [3], [7], [17], [25], [39], [49], enabling better detection of known ground truth communities due to the additional information encoded in the temporal behavior. We generalize the definition employed by the latter group of methods by modeling *dynamic behavioral communities* as a persistent set of nodes whose interaction activity is a mixture of (i) periodic “oscillations” and (ii) one-off (outlier) periods of increased activity.

Consider, as an example, three subreddit channels in Reddit - *r/Frugal*, *r/self* and *r/Programming* - whose user base and activity are summarized in Fig. 1. The three subreddits vary in both the number of participating users and activity levels, however, their user bases overlap significantly 1(a). While *r/Programming* observes daily oscillations with occasional larger bursts, the patterns in *r/Frugal* and *r/self*, which are subreddits on frugal living or advice and discussion on personal issues, respectively, are quite different 1(b). The overlap of members makes separating these communities challenging, however, community-specific temporal interaction patterns, including differences in periodicity as well as occurrences of bursts, can be exploited in addition to network separability to improve community detection. Our goal in this work is to infer such *behavioral communities* from raw interaction data without a priori knowledge of the community delineation (e.g. subreddits within which the interaction occurred).

We propose PERCeIDs: a framework for periodic community detection which models *behavioral communities* as groups of nodes with high within-group interactions and group-specific periodic patterns of activity levels which occasionally observe bursts. The objective in PERCeIDs seamlessly incorporates (i) network locality based on tensor decomposition, (ii) parsimonious per-community period estimation employing a *Ramanujan basis* dictionary and the theory of *Nested Periodic Matrices (NPM)*, and (iii) sparse outlier burst detection. Unlike the majority of existing work, we do not require prior knowledge of natural seasonality in the data, but estimate those directly from raw interaction events. PERCeIDs is flexible in that it handles communities with different periodicity, including communities which are not periodic within the same model. While this flexibility comes with the computational cost of accounting for large numbers of candidate periods, we propose an effective and efficient procedure for estimating the maximal period in a given dataset. Our framework outperforms state-of-the-art baselines on both synthetic and real-world datasets: an increase in NMI from 0.2 to 0.4, and a decrease in Jensen-Shannon divergence from 0.3 to nearly 0. Beyond community detection PERCeIDs characterizes the temporal community behavior as a mixture of periodic and outlier components which is useful on its own in explanatory analytics and modeling and prediction tasks.

Our contributions in this paper are as follows:

- **Novelty:** To the best of our knowledge, PERCeIDs is the first dynamic community detector which learns temporal periodicity directly from network interactions without prior knowledge.
- **Accuracy and robustness:** PERCeIDs exploits the co-dependence of network locality and regular temporal activity, making it robust to noise, community overlap, bursty activity and varying temporal observation windows. It consistently outperforms state-of-the-art alternatives enabling up to two-fold increase in NMI on synthetic data and up to 23% on real-world data.
- **Practical efficiency:** While we explicitly optimize both community membership and periodicity fit, our optimization framework does not incur significant computational overhead compared to simpler and less-accurate alternatives, making it applicable to real world datasets.

II. Related work

Community detection. Community detection is a fundamental network analysis task with a long history of research in the case of static networks [6], [20], [36], [55], [62], [63]. Additional consideration of timing of node interactions has been shown to improve the quality of community detection [23], [25], [49]. A key distinction among models for the dynamic setting is the expected temporal behavior of communities. One group of methods focus on long-term evolution and postulate that the community membership changes over time [7], [9], [31], [40], [45], [53], [54], with provisions for local smoothing [47]. Alternative models for short-term evolution assume a stationary membership and varying activity (number of interactions) over time. In this second group, some methods assume persistent activity in a single active interval [3], [16], [22], [39], [52], others expect recurrent user-provided consistency [49], while a different group of methods learns data-driven community dynamics [41] with proposals for regularization to enforce constant "on/off" periods of activation, and thus, robustness to the temporal resolution [25]. Our work similarly assumes stationary community membership, however, it models the temporal activity as a mixture of periodic and outlier (one-off) behavior components learned directly from temporal interaction data.

Period estimation. Period estimation has been classically approached by Fourier methods [29], [38], where time series are transformed into frequency domain and natural periods in the data are determined based on high-amplitude frequency bins. Such approaches often predict a large number of spurious periods [57]. Auto-correlation is another approach for period estimation [14], however, it cannot determine periods automatically, but typically requires a threshold for dominant periods, post-processing or joint consideration with the Fourier spectrogram [59]. Recently Tenneti and colleagues proposed an efficient parameter-free and sparse solution for period estimation based on a periodic dictionary [57], improving on earlier work which suffered solution ambiguity due to non-orthogonality of the employed dictionary [51]. The key idea in the former is the adoption of Nested periodic matrices (NPM) which enable significant improvements over Fourier and other bases in the context of timeseries [57]. We model the temporal community activity as a mixture of periodic (based on NPMs) and bursty (outlier) patterns while simultaneously detecting the community membership which gives rise to this activity. Hence, our methodology can be viewed as a generalization of NPM-based models (i) applied to dyadic interaction data and (ii) also allowing for occasional outlier (non-periodic) behavior.

Anomaly and change point detection in networks. Another relevant set of problems is anomaly detection in networks [2], [5], particularly in the case of community structure in dynamic graphs. Many of the existing techniques define anomalies in the changes on vertices/edges [4], global snapshots [33] and in between [11], [42], [43], [60]. These methods are either focused on consecutive snapshot distance scores or longer timeseries, but restricted to nodes/edges or statistics of the full graph. Different from those, our model focuses on deviations from periodic behavior at the community level. Closer to our approach are tensor-based community and change point detectors [23], [25], [32]. These methods discover change points as a post-processing step to the factorization, while we learn a mixture of seasonal and outlier components jointly with the community membership, resulting in better quality of our method on a variety of tasks as we demonstrate in the evaluation.

III. Preliminaries and Notation

Before we formalize our problem of periodic community detection, we introduce some necessary preliminaries and the notation we adopt throughout the paper (summarized in Tbl. I). A *dynamic network* $\mathcal{G}$ is a sequence of network *snapshots* (or instances) $\mathcal{G} = \{G_1, G_2, ..., G_T\}$ over a fixed finite set of nodes $|V| = m$ and exhibiting varying connections among them over time. Each snapshot $G_t = \{V, \mathbf{W}_t\}$ is characterized by a symmetric adjacency matrix $\mathbf{W}_t \in \mathbb{R}^{m \times m}$ whose elements denote the number of interactions among a pair of nodes at time t. Collectively the snapshot adjacency matrices form a 3-way node×node×time tensor $\boldsymbol{\mathcal{W}} \in^{m \times m \times T}$.

A time series (or a signal) $x(t)$ is *g-periodic* if there exists an integer g such that $|x(n+g) - x(n)| \leq \epsilon, \forall n$, where g is the smallest among all integers that satisfy the inequality and ϵ is a small real value. Different from traditional Fourier methods for period estimation, we employ a sparse representation (SR) to learn periods in community activity by employing *Nested Periodic Matrices (NPM)* [57]. Let the integers $\{d_1, d_2, ...d_K\}$ be the divisors of g sorted in increasing order. The Nested Periodic Matrix (NPM) for period g is defined as:

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

where each $\mathbf{P}_{d_i} \in \mathbb{R}^{g \times \phi(d_i)}$ is a period basis matrix for period d_i. Columns of $\mathbf{P}_{d_i}$ are time series with period d_i and $\phi(d_i)$ 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. Note that since $g = \sum_i \phi(d_i)$, $\mathbf{\Phi}_g \in \mathbb{R}^{g \times g}$ is a square matrix [26]. For example, a 15-periodic signal ($g = 15$) can be well represented as a combination of two 3- and 5-periodic symbols (the non-trivial divisors of 15), resulting in the NPM: $\mathbf{\Phi}_{15} = [\mathbf{P}_3, \mathbf{P}_5]$. While there are different bases to construct the $\mathbf{P}_i$ matrices in an NPM, we employ a *Ramanujan basis dictionary (RBD)* that is a common family for constructing NPMs. The RBD is based on the Ramanujan sum [57]:

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

where j is the imaginary unit and gcd denotes the greatest common divisor function, i.e. the values of k and d_i are co-prime. For a divisor d_i of g, the Ramanujan basis dictionary matrix $\mathbf{P}_{d_i}$ is comprised of a series of down-shifted sequences of Ramanujan sums with the following shape:

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

It can be shown that the columns of $\mathbf{P}_{d_i}$ are orthogonal to each other and that the resulting $\mathbf{P}_{d_i}$ is full-rank [57].

IV. Problem Formulation

Given a dynamic network $\mathcal{G}$ with a corresponding adjacency tensor $\boldsymbol{\mathcal{W}}$ our aim is to detect *periodic behavioral communities*: overlapping sets of nodes exhibiting a mixture of periodic and outlier interaction activity levels. With this design goal in mind, we next formalize an optimization objective grounded in tensor factorization and time series period estimation.

$\mathcal{G}$	A set of implicit dynamic network instances
G_t	Network snapshot at time t
$\mathbf{B}_{p,q}$	The $l_{p,q}$-norm of $\mathbf{B}$, $\mathbf{B}_{p,q} = \left(\sum_{i=1}^m \|\mathbf{B}_i\|_p^q\right)^{\frac{1}{q}}$
$\|\mathbf{B}\|_*$	$\|\mathbf{B}\|_* = \sum_i |\sigma_i|$ nuclear norm
σ_i	the i-th largest singular value of $\mathbf{B}$
$\mathbf{B}_i, \mathbf{B}^i$	the i-th row and column of $\mathbf{B}$
$\boldsymbol{\mathcal{W}} \in^{m \times m \times T}$	adjacency tensor
$\mathbf{W}_t \in \mathbb{R}^{m \times m}$	adjacency matrix at time t
$\mathbf{W}_{(i)}$	The mode-i matrix of $\boldsymbol{\mathcal{W}}$.
$\mathbf{\Phi} \in^{\mathbb{R}} T \times N$	Nested periodic matrix
$\mathbf{U} \in \mathbb{R}^{m \times K}$	Community matrix
$\mathbf{X} \in \mathbb{R}^{T \times K}$	Temporal profile matrix of communities
$\mathbf{Y} \in \mathbb{R}^{N \times K}$	Sparse coefficients of periods
$\mathbf{H} \in \mathbb{R}^{N \times N}$	The penalty matrix for the dictionary
K	Number of communities
m	Number of nodes in the network
T	Number of time points
N	Number of columns of the dictionary $\mathbf{\Phi}$
$\odot$	The Khatri-Rao product

TABLE I
KEY NOTATION USED THROUGHOUT THE PAPER

Community membership via tensor factorization. A natural and widely-adopted approach to grouping nodes in dynamic networks is to factorize the adjacency tensor $\boldsymbol{\mathcal{W}}$, yielding a coupling of nodes to factors which can be interpreted as soft community membership and temporal activity profiles for each of the factors. We adopt a non-negative CP decomposition in which the tensor is decomposed into a sum of rank-one tensors as $\boldsymbol{\mathcal{W}} = \sum_k \mathbf{u}_k \circ \mathbf{u}_k \circ \mathbf{x}_k$ [23], where the factors $\mathbf{u}_k$ corresponding to the node modes of the tensor are the same due to the symmetry of temporal snapshots. Stacking the factor vectors $\mathbf{u}_k$ and $\mathbf{x}_k$ in matrices $\mathbf{U}$ and $\mathbf{X}$ yields a concise matrix CP objective:

$$\Psi(\mathbf{U}, \mathbf{X}) = \|\boldsymbol{\mathcal{W}} - [\mathbf{U}, \mathbf{X}]\|_F^2, \tag{4}$$

where $[\mathbf{U}, \mathbf{X}]$ is a shortcut for the three-way product of the factor matrices leading to the k-rank reconstruction of $\boldsymbol{\mathcal{W}}$.

Period activity estimation. To learn periodicity in communities' activity, we further impose structure on the temporal profiles within $\mathbf{X}$ via a sparse periodic+outlier decomposition. More specifically, we cast the period estimation problem as a sparse representation via a Ramanujan basis dictionary (RBD) within the Nested Periodic Matrices (NPM) framework. The RBD, among other possible basis families, is advantageous due to its orthogonality ensuring solution uniqueness. Intuitively, we seek to represent the temporal activity of a community as a linear combination of short-period time series, called basis. We also enforce a sparse selection of the basis and allow for non-periodic one-off components leading to a sparse low-error reconstruction.

We formulate the period estimation for the temporal profile of the i-th community $\mathbf{X}_i \in \mathbb{R}^{T \times 1}$ as estimating a sparse mixture fit $\mathbf{y}$ of the bases from the NPM:

$$\underset{\mathbf{y},\mathbf{o}}{\operatorname{argmin}} \|\mathbf{y}\|_1, \; s.t. \; \mathbf{X}_i = \mathbf{\Phi}\mathbf{y}_i + \mathbf{o}_i + \mathbf{e}, \tag{5}$$

where $\mathbf{\Phi} \in \mathbb{R}^{T \times N}$ denotes a Ramanujan basis NPM with columns periodic individual bases extended to the length of

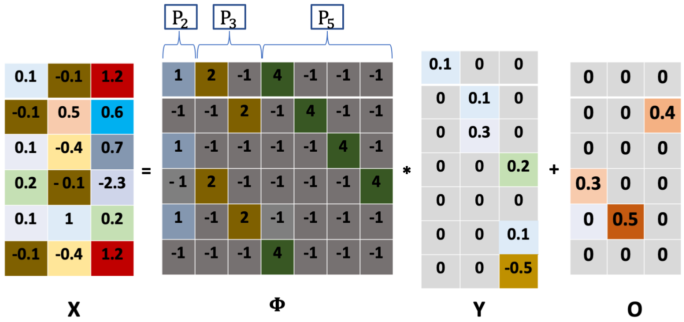


Fig. 2. An illustration of the proposed period estimation model: The columns of $\mathbf{X}$ represents signals with periods of 2, 3, 5, respectively. In addition, each column has a outlier. We define a periodic dictionary $\mathbf{\Phi} = [\mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_5]$ for learning the periods of $\mathbf{X}$. Each column can be expressed as a linear combination of basis in the $\mathbf{\Phi}$ and outliers, where $\mathbf{Y}$ contains the sparse coefficients of these combinations.

the input signal T, $\mathbf{o}_i$ is a sparse time series modeling large one-off deviations (outliers) from the periodic behavior and $\mathbf{e}$ is a low-amplitude Gaussian noise in the temporal activity profile. We introduce the outlier term $\mathbf{o}_i$ to account for natural bursty time intervals which are typical in real-world data. This also adds representational flexibility as communities which do not exhibit any periodicity can still be fit as a sequence of outlier time intervals in $\mathbf{o}_i$ and a near-zero period mixture $\mathbf{y}_i$.

The number of columns N of $\mathbf{\Phi}$ is determined by the maximum candidate period g_{max} of interest. Recall that in the basic NPM model $\mathbf{\Phi}$ is composed of stacked submatrices corresponding to the divisors of the maximal period of interest (Eq. 1). While a default $g_{max} = T$ would exhaustively cover all possible periods in the data, it comes at a high computational cost. We allow a user to define $g_{max} = N \leq T$ and also propose an efficient and effective data-driven estimator for this parameter directly from data in Sec. V-B which works well in practice.

An illustration of the decomposition from Eq. 5 is presented in Fig. 2. The temporal profiles are in the columns of $\mathbf{X}$ and a specific profile $\mathbf{X}_2$ is reconstructed as a sum of a periodic component $\mathbf{\Phi}\mathbf{y}_2$ and a bursty (outlier) component $\mathbf{o}_2$. In this example the NPM of the Ramanujan basis $\mathbf{\Phi} = [\mathbf{P_2 P_3 P_5}]$ contains only the nested matrices for periods 2 (first column) and 3 (columns 2-3) and 5 (columns 4-7) extended to the length of the temporal profile $\mathbf{X}$. The fit $\mathbf{y}_2$ for $\mathbf{X}_2$, has non-zero elements for the second and third basis vectors (corresponding to period 3), while its outlier profile contains an outlier burst in the 5-th time step of $\mathbf{o}_2$ which cannot be explain by the learned periodicity.

Our goal is to characterize faithfully and concisely the natural *behavioral communities* in the data both in terms of membership and periods. Hence, in order to improve the explanatory power of our model, we seek to encourage fits with a sparse basis of short periods representing the overall periodicity of a given activity profile. Without any constraints, the objective Eq. 5 will not select such minimum-basis representation. Hence, we introduce a penalty for selecting a large-period basis through a diagonal cost matrix of monotonically increasing elements: $\mathbf{H} \in \mathbb{R}^{N \times N}, \mathbf{H}_{ii} = p^2$, where p is the period of the i-th column in $\mathbf{\Phi}$. In addition, we add weighting parameters λ_1 and λ_2 to balance the fit between periodic and outlier components. Combining the fit for all communities we obtain the following activity objective in matrix form:

$$\underset{\mathbf{Y},\mathbf{O}}{\operatorname{argmin}} \left\|\mathbf{X} - \mathbf{\Phi}\mathbf{Y} - \mathbf{O}\right\|_F^2 + \lambda_1 \left\|\mathbf{H}\mathbf{Y}\right\|_1 + \lambda_2 \left\|\mathbf{O}\right\|_1, \tag{6}$$

where $\mathbf{y}_i$ fits are stacked in the columns of $\mathbf{Y}$, the term $\|\mathbf{HY}\|_1$ sparsifies the selected periodicity fits by progressively penalizing larger periods, and $\|\mathbf{O}\|_1$ is comprised of the outlier component fits for each community in its columns. Note that we have incorporated the constraint from Eq. 5 as a reconstruction penalty term excluding the Gaussian error.

Periodic community detection. We integrate the community membership and periodic activity objectives above into a unified periodic community objective as follows:

$$\begin{aligned} \underset{\mathbf{U},\mathbf{X},\mathbf{Y},\mathbf{O}}{\operatorname{argmin}} \; & \frac{1}{2}\left\|\boldsymbol{\mathcal{W}} - [\mathbf{U}, \mathbf{X}]\right\|_F^2 + \lambda_0 \left\|\mathbf{X} - \mathbf{\Phi}\mathbf{Y} - \mathbf{O}\right\|_F^2 \\ & + \lambda_1 \left\|\mathbf{H}\mathbf{Y}\right\|_1 + \lambda_2 \left\|\mathbf{O}\right\|_1, \end{aligned} \tag{7}$$

where λ_0 is a balance parameter to control the relative importance of the periodic activity fit and the tensor reconstruction error. Minimizing this objective will enforce the selected communities to have pronounced periodicity since, if present, they will lead to a small fitting error in the activity term. At the same time, the model provides flexibility to fit both periodic and outlier behavior specific to each community. By setting a proper combination of λ_1 and λ_2, a user can promote periodic or bursty behavior in the fit.

V. Optimization solution

A. *PERCeIDs: a periodic community detector*

We adopt an Alternating Optimization (AO) approach for the objective from of Eq. 7, i.e., to minimize the objective function we minimize over each individual variable in turn. We first separate Eq. 7 into subproblems involving individual variables, obtaining the following:

$$\begin{cases} \mathbf{U}: \underset{\mathbf{U}\geq 0}{\operatorname{argmin}} \; \frac{1}{2}\left\|\boldsymbol{\mathcal{W}} - [\mathbf{U}, \mathbf{X}]\right\|_F^2 & \text{(a)} \\ \mathbf{X}: \underset{\mathbf{X}\geq 0}{\operatorname{argmin}} \; \frac{1}{2}\left\|\boldsymbol{\mathcal{W}} - [\mathbf{U}, \mathbf{X}]\right\|_F^2 + \lambda_0 \left\|\mathbf{X} - \mathbf{\Phi}\mathbf{Y} - \mathbf{O}\right\|_F^2 & \text{(b)} \\ \mathbf{Y}: \underset{\mathbf{Y}}{\operatorname{argmin}} \; \lambda_0 \left\|\mathbf{X} - \mathbf{\Phi}\mathbf{Y} - \mathbf{O}\right\|_F^2 + \lambda_1 \left\|\mathbf{H}\mathbf{Y}\right\|_1 & \text{(c)} \\ \mathbf{O}: \underset{\mathbf{O}}{\operatorname{argmin}} \; \lambda_0 \left\|\mathbf{X} - \mathbf{\Phi}\mathbf{Y} - \mathbf{O}\right\|_F^2 + \lambda_2 \left\|\mathbf{O}\right\|_1 & \text{(d)} \end{cases} \tag{8}$$

We iteratively optimize the sub-problems until a convergence stopping criteria is met. We next discuss the optimization of individual sub-problems and then combine those in our overall optimization scheme PERCeIDs.

Updates for U: The tensor factorization objective Eq. 8(a) can be rewritten based on the modes of $\boldsymbol{\mathcal{W}}$ as follows:

$$\underset{\mathbf{U}\geq 0}{\operatorname{argmin}} \frac{1}{2} \left\| \mathbf{W}_{(i)} - \mathbf{U}_{(i)} \left(\mathbf{X} \odot \mathbf{U}_{(j\neq i)}\right)^T \right\|_F^2 \tag{9}$$

where $\mathbf{W}_{(i)}$ is the mode-i matrix obtained by linearizing all indices of the tensor except the i-th. In our case, we will consider the two (equivalent) linearizations for the node modes, i.e., $i \in (1,2)$, and not the one for the temporal dimension. Due to the non-negativity constraint on $\mathbf{U}$, this objective is not the original unconstrained CANDECOMP/PARAFAC (CP) tensor decomposition [27], and thus cannot be directly solved by Alternating Least Squares (ALS) [12]. Instead, we employ Alternating Optimization Alternating Direction Method of Multipliers (AOADMM) [28], to handle this subproblem. The $0-\infty$ penalty function $\mathcal{D}(\mathbf{U})$ is employed to enforce the non-negativity constraint:

$$[\mathcal{D}(\mathbf{U})]_{ab} = \begin{cases} 0 & \text{if } \mathbf{U}_{ab} \geq 0 \\ +\infty & otherwise, \end{cases} \tag{10}$$

leading to the following revised Eq. 9 for mode $i = 1$ ($i = 2$ is handled analogously):

$$\begin{aligned} &\underset{\mathbf{U}_{(1)},\grave{\mathbf{U}}_{(1)}}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{W}_{(1)}^T - \mathbf{R}\grave{\mathbf{U}}_{(1)}\right\|_F^2 + \mathcal{D}(\mathbf{U}_{(1)}), \\ &s.t\ \mathbf{U}_{(1)} = \grave{\mathbf{U}}_{(1)}^T, \mathbf{U} \geq 0, \end{aligned} \tag{11}$$

where $\mathbf{R} = \mathbf{X} \odot \mathbf{U}_{(2)}$, $\eta = tr\left(\mathbf{R}^T\mathbf{R}\right)/K$, and $\grave{\mathbf{U}}_{(1)}$ is an auxiliary ADMM variable. We employ the ADMM algorithm and iterate over the following updates:

$$\begin{cases} \grave{\mathbf{U}}_{(1)} \leftarrow \left(\mathbf{R}^T\mathbf{R} + \eta\mathbf{I}\right)^{-1}\left[\mathbf{R}^T\mathbf{W}_{(1)}^T + \eta\left(\mathbf{U}_{(1)} + \mathbf{Q}_{(1)}\right)^T\right] & (a) \\ \mathbf{U}_{(1)} \leftarrow \underset{\mathbf{U}_{(1)}}{\operatorname{argmin}}\ \mathcal{D}(\mathbf{U}_{(1)}) + \frac{\eta}{2}\left\|\mathbf{U}_{(1)} - \grave{\mathbf{U}}_{(1)}^T + \mathbf{Q}_{(1)}\right\|_F^2 & (b) \\ \mathbf{Q}_{(1)} \leftarrow \mathbf{Q}_{(1)} + \mathbf{U}_{(1)} - \grave{\mathbf{U}}_{(1)}^T & (c), \end{cases} \tag{12}$$

where $\mathbf{Q}_{(1)}$ is an intermediate variable. Since $\mathcal{D}(\mathbf{U}_{(1)})$ is an element-wise indicator function, the update of $\mathbf{U}_{(1)}$ can be obtained by thresholding at zero, where $\hat{\mathbf{U}}_{(1)} = \grave{\mathbf{U}}_{(1)}^T - \mathbf{Q}_{(1)}$. The final community matrix $\mathbf{U}$ is then computed as the average of the two node factors $\mathbf{U} = \frac{1}{2}\left[\mathbf{U}_{(1)} + \mathbf{U}_{(2)}\right]$ due to the adjacency tensor symmetry.

Updates for $\mathbf{X}$: The update of the temporal factor $\mathbf{X}$ in Eq. 8(b) is also not a standard tensor factorization objective due to the periodic decomposition term. However, similar to the updates of $\mathbf{U}$, we can employ AOADMM, reformulating the objective as follows:

$$\begin{aligned} &\underset{\mathbf{X},\grave{\mathbf{X}}}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{W}_{(3)}^T - \mathbf{V}\grave{\mathbf{X}}\right\|_F^2 + \lambda_0\left\|\mathbf{X} - \mathbf{\Phi}\mathbf{Y} - \mathbf{O}\right\|_F^2 + \mathcal{D}(\mathbf{X}) \\ &\qquad s.t\ \mathbf{X} = \grave{\mathbf{X}}^T, \mathbf{X} \geq 0, \end{aligned} \tag{13}$$

where $\mathbf{W}_{(3)}$ is the mode-3 (temporal) unfolding matrix of $\boldsymbol{\mathcal{W}}$, $\mathbf{V} = \mathbf{U} \odot \mathbf{U}$, and $\grave{\mathbf{X}}$ is an auxiliary ADMM variable. The optimization via ADMM iterates over the following updates,

$$\begin{cases} \grave{\mathbf{X}} = \left(\mathbf{V}^T\mathbf{V} + \rho\mathbf{I}\right)^{-1}\left[\mathbf{V}^T\mathbf{W}_{(3)}^T + \rho\left(\mathbf{X} + \mathbf{S}\right)^T\right] \\ \mathbf{X} = \underset{\mathbf{X}}{\operatorname{argmin}}\ \lambda_0\left\|\mathbf{X} - \mathbf{\Phi}\mathbf{Y} - \mathbf{O}\right\|_F^2 + \frac{\rho}{2}\left\|\mathbf{X} - \grave{\mathbf{X}}^T + \mathbf{S}\right\|_F^2 + \mathcal{D}(\mathbf{X}) \\ \mathbf{S} = \mathbf{S} + \mathbf{X} - \grave{\mathbf{X}}^T \end{cases} \tag{14}$$

where $\rho = tr\left(\mathbf{V}^T\mathbf{V}\right)/K$ and $\mathbf{S}$ is an intermediate variable. To update $\mathbf{X}$ (second update), we obtain the result from the first two terms and then threshold at zero. We solve the minimization for $\mathbf{X}$ by setting the gradient of the first two terms (w.r.t $\mathbf{X}$) to zero, leading to the closed-form solution:

$$\mathbf{X} = \left[2\lambda_0(\mathbf{\Phi}\mathbf{Y} + \mathbf{O}) + \rho(\grave{\mathbf{X}}^T - \mathbf{S})\right] / \left(2\lambda_0 + \rho\right) \tag{15}$$

Updates for $\mathbf{Y}$: The objective in Eq. 8(c) is a generalized lasso regression w.r.t. $\mathbf{Y}$ [58], thus, it can be reduced to a standard lasso problem as follows:

$$\underset{\mathbf{T}}{\operatorname{argmin}}\ \lambda_0\left\|\mathbf{F} - \mathbf{\Phi}\mathbf{H}^{-1}\mathbf{T}\right\|_F^2 + \lambda_1\left\|\mathbf{T}\right\|_1, \tag{16}$$

where $\mathbf{F} = \mathbf{X} - \mathbf{O}$ and $\mathbf{T} = \mathbf{H}\mathbf{Y}$. Note, that $\mathbf{H}$ is invertible since it is a diagonal matrix with non-zero diagonal elements. We solve the standard lasso problem in Eq. 16 by the least-angle regression (LARS) algorithm [19] (details omitted due to space constraints). Given a solution for $\mathbf{T}$, the periodicity mixture matrix $\mathbf{Y}$ is computed as $\mathbf{Y} = \mathbf{H}^{-1}\mathbf{T}$.

Updates for O: By substituting $\mathbf{Z} = \mathbf{X} - \mathbf{\Phi}\mathbf{Y}$ and $\alpha = \frac{\lambda_2}{2\lambda_0}$, we can rewrite Eq. 8(d) as $\underset{\mathbf{A}}{\operatorname{argmin}}\ \frac{1}{2}\left\|\mathbf{O} - \mathbf{Z}\right\|_F^2 + \alpha\left\|\mathbf{O}\right\|_1$. This has a similar form to the soft-thresholding problem [8], and therefore, $\mathbf{O}$ can be optimzed based on the following lemma.

Lemma 1: $\mathbf{A}_{ij} = sign\left(\mathbf{B}_{ij}\right) \times max\left(\left|\mathbf{B}_{ij}\right| - \alpha, 0\right)$ is a closed-form solution for $\underset{\mathbf{A}}{\operatorname{argmin}}\ \frac{1}{2}\left\|\mathbf{A} - \mathbf{B}\right\|_F^2 + \alpha\left\|\mathbf{A}\right\|_1$, if $\alpha > 0$.

Lemma 1 gives us an element-wise update for $\mathbf{O}$:

$$\mathbf{O}_{ij} = sign\left(\mathbf{Z}_{ij}\right) \times max\left(\left|\mathbf{Z}_{ij}\right| - \frac{\lambda_2}{2\lambda_0}, 0\right). \tag{17}$$

Overall algorithm: PERCeIDs. We summarize all optimization steps discussed above in Alg. 1. The input to PERCeIDs is the adjacency tensor, the maximal candidate period to be considered, number of communities to extract and the balance parameters. Each of the variables is updated in turn while holding the remaining fixed in the main loop (Steps 4-28). Note that we have omitted the iteration super-index q in the steps for cleaner notation. First the node factors $\mathbf{U}$ of the tensor are fit in Steps 5-13. While at convergence we expect them to be close to each other, we still need to update them independently and only average them after convergence (Step 29). Next we update the temporal profile $\mathbf{X}$ (Steps 14-21), the period matrix $\mathbf{Y}$ (Steps 22-24) and the outlier time steps $\mathbf{O}$ (Steps 25-27). Finally, we check for convergence of each of the variables and the global objective (Steps 28-29).

Updating the tensor factors, $\mathbf{U}_{(i)}$ in Step 9 and $\mathbf{X}$ in Step 17, are the most expensive operations. Since the Step 9 and 11 involve a matrix inversion, the worst-case complexity is $O\left(K^3\right)$. In particular, computing $\mathbf{R}^T\mathbf{W}_{(i)}^T$ and $\mathbf{V}^T\mathbf{W}_{(3)}$ in these steps requires a large amount of memory due to the high dimensionality of $\mathbf{W}_{(i)}$. Instead of computing those directly, we make use of the MTTKRP operation [25] which exploits sparsity in the matrix when available and has a complexity of $O\left(mK\right)$, where m is the number of nonzero elements in $\boldsymbol{\mathcal{W}}$. The update of $\mathbf{T}$ in Step 23 has a complexity of $O\left(T^2K + T^3\right)$ [19]. Therefore,

Algorithm 1: PERCeIDs: Periodic Community Detection

Input: Observations $\boldsymbol{\mathcal{W}}$, maximum period g_{max}, number of communities K, and balance parameters $(\lambda_0, \lambda_1, \lambda_2)$
Output: Community membership $\mathbf{U}$, temporal profile $\mathbf{X}$, community periods $\mathbf{Y}$ and outlier time steps $\mathbf{O}$
Initialize: $\theta = 10^{-3}$, $q = 0$.
while *not converged* **do**
  // Update $\mathbf{U}_{(1)}^{q+1}$ and $\mathbf{U}_{(2)}^{q+1}$:
  for *i = 1:2* **do**
    $\mathbf{R} = \mathbf{X} \odot \mathbf{U}_{(j \neq i, j \in (1,2))}$ and $\eta = tr\left(\mathbf{R}^T\mathbf{R}\right)/K$;
    while *not converged* **do**
      $\grave{\mathbf{U}}_{(i)} = \left(\mathbf{R}^T\mathbf{R} + \eta\mathbf{I}\right)^{-1}\left[\mathbf{R}^T\mathbf{W}_{(i)}^T + \eta\left(\mathbf{U}_{(i)} + \mathbf{Q}\right)^T\right]$;
      $\mathbf{U}_{(i)} = max\left[\grave{\mathbf{U}}_{(i)}^T - \mathbf{Q}_{(i)}, 0\right]$;
      $\mathbf{Q}_{(i)} = \mathbf{Q}_{(i)} + \mathbf{U}_{(i)} - \grave{\mathbf{U}}_{(i)}^T$;
    end
  end
  // Update $\mathbf{X}^{q+1}$:
  $\mathbf{V} = \mathbf{U}_{(1)}^{q+1} \odot \mathbf{U}_{(2)}^{q+1}$ and $\rho = tr\left(\mathbf{V}^T\mathbf{V}\right)/K$;
  while $\mathbf{X}^{q+1}$ *has not converged* **do**
    $\grave{\mathbf{X}} = \left(\mathbf{V}^T\mathbf{V} + \rho\mathbf{I}\right)^{-1}\left[\mathbf{V}^T\mathbf{W}_{(3)} + \rho\left(\mathbf{X} + \mathbf{S}\right)^T\right]$;
    $\mathbf{X} = \left[2\lambda_0(\mathbf{\Phi}\mathbf{Y} + \mathbf{O}) + \rho(\grave{\mathbf{X}}^T - \mathbf{S})\right]/(2\lambda_0 + \rho)$;
    $\mathbf{X} = max\left(\mathbf{X}, 0\right)$;
    $\mathbf{S} = \mathbf{S} + \mathbf{X} - \grave{\mathbf{X}}^T$;
  end
  // Update $\mathbf{Y}^{q+1}$:
  Update $\mathbf{T}$ using the LARS algorithm Eq. 16;
  $\mathbf{Y} = \mathbf{H}^{-1}\mathbf{T}$;
  // Update $\mathbf{O}^{q+1}$:
  $\mathbf{Z} = \mathbf{X}^{q+1} - \mathbf{\Phi}\mathbf{Y}^{q+1}$;
  $\mathbf{O}_{ij} = max\left(|\mathbf{Z}_{ij}| - \frac{\lambda_2}{2\lambda_0}, 0\right) \cdot sign\left(\mathbf{Z}_{ij}\right)$;
  // Check the convergence:
  $\left\|\mathbf{U}^{q+1} - \mathbf{U}^q\right\|_\infty \leq \theta$, $\left\|\mathbf{X}^{q+1} - \mathbf{X}^q\right\|_\infty \leq \theta$, $\left\|\mathbf{Y}^{q+1} - \mathbf{Y}^q\right\|_\infty \leq \theta$, $\left\|\mathbf{O}^{q+1} - \mathbf{O}^q\right\|_\infty \leq \theta$ and $\left|\frac{f^{q+1} - f^q}{f^q}\right| \leq \theta$, where f^q is the objective value of Eq. 7;
  $q = q + 1$;
end
$\mathbf{U} = \frac{1}{2}\left[\mathbf{U}_{(1)} + \mathbf{U}_{(2)}\right]$;

the overall complexity of each iteration of the proposed method is $O\left[mK + 2t_{\mathbf{U}_{(i)}}K^3 + t_{\mathbf{X}}K^3 + t_{\mathbf{T}}(T^2K + T^3)\right]$, where $t_{\mathbf{U}_{(i)}}$, $t_{\mathbf{X}}$ and $t_{\mathbf{T}}$ denote the number of iterations of the sub-scripted variables. As the number of factors is typically a constant with respect to the input size, the overall complexity of an iteration can be simplified to $O(m + t_{\mathbf{T}}T^3)$.

B. Estimating the maximum candidate period g_{max}

One important parameter for PERCeIDs is the maximum candidate period g_{max} which determines the width N of the periodic dictionary matrix $\mathbf{\Phi}$. On one hand, we would like to consider as many potential periods as possible (i.e. $g_{max} = T$), while on the other artificially high g_{max} may incur unnecessary computational overhead as our basis ensures that large periods are "decomposed" into a minimal set of co-prime divisors [56]. Hence, we develop an efficient estimator for g_{max} based on the intuition that the temporal factors of $\boldsymbol{\mathcal{W}}$, even if not corresponding to accurate communities, will contain noisy versions of the periods. We can then, "de-noise" these observations for a reliable global estimate for g_{max}.

Algorithm 2: Estimate g_{max}

1 **Input:** Observations $\boldsymbol{\mathcal{W}}$
2 **Output:** Maximum period g_{max}
3 Compute $\mathbf{X}$ by NTF on $\boldsymbol{\mathcal{W}}$;
4 Compute covariance matrix $\mathbf{E} = \mathbf{X}\mathbf{X}^T$;
5 Separate signal from noise covariance: $[\mathbf{E}_s, \mathbf{E}_n]$ = RPCA($\mathbf{E}$);
6 $\mathbf{E}_s = \mathbf{M}\mathbf{\Lambda}\mathbf{M}^T$;
7 $\mathbf{\Pi}_{opt} = \mathbf{M}\mathbf{\Lambda}\left(\mathbf{\Lambda} + \mu\mathbf{M}^T\mathbf{E}_n\mathbf{M}\right)^{-1}\mathbf{M}^T$;
8 $\mathbf{X}_s = \mathbf{\Pi}_{opt}\mathbf{X}$;
9 P_{range} = autocorr($\mathbf{X}_s$);
10 $g_{max} = max(P_{range})$.

The steps of our proposed g_{max} estimation approach are listed in Alg. 2. We first obtain a temporal factor $\mathbf{X}$ by standard NTF on $\boldsymbol{\mathcal{W}}$ (Step 3). In [15], the maximum period is estimated by applying auto-correlation, however, such an approach neglects the inherent noise in the data. To address this, we first reduce the noise in our observed temporal factors $\mathbf{X}$ via a de-noising linear projection [48], inferred based on the covariance matrices of the noise-free signal $\mathbf{X}_s$ and that of the noise $\mathbf{X}_n$, denoted $\mathbf{E}_s$ and $\mathbf{E}_n$ respectively. We formulate the separation of the noise-free and noise covariances as a matrix decomposition problem assuming an additive model for the observed covariance: $\mathbf{E} = \mathbf{E}_s + \mathbf{E}_n$. We have the following objective:

$$\underset{\mathbf{E}_s, \mathbf{E}_n}{\operatorname{argmin}} \left\|\mathbf{E} - \mathbf{E}_s - \mathbf{E}_n\right\|_F^2 + \gamma_1\left\|\mathbf{E}_x\right\|_* + \gamma_2\left\|\mathbf{E}_n\right\|_1, \qquad (18)$$

where we have imposed a low-rank regularizer on the noise-free covariance estimate and sparsity regularizer on the noise covariance with γ_1 and γ_2 serving as balance parameters. This can be solved by Robust principal component analysis (RPCA) [13] (Step 5, where we have skipped the optimization details for brevity). We then perform an eigen decomposition of the estimated noise-free covariance (Step 6) and compute the optimal noise reduction projector for $\mathbf{X}$ (Step 7) as:

$$\mathbf{\Pi}_{opt} = \mathbf{M}\mathbf{\Lambda}\left(\mathbf{\Lambda} + \mu\mathbf{M}^T\mathbf{E}_n\mathbf{M}\right)^{-1}\mathbf{M}^T, \qquad (19)$$

where $\mathbf{M}$ and $\mathbf{\Lambda}$ are the eigenvectors and diagonal eigenvalue matrix of $\mathbf{E}_s$. Given the estimated projector $\mathbf{\Pi}_{opt}$, we obtain the de-noised temporal factor $\mathbf{X}_s$ (Step 8) and employ a scalable auto-correlation approach to analyze $\mathbf{X}_s$ (Step 9) and estimate hidden periods. We estimate the maximum period as the largest period detected by auto-correlation which survives a threshold of one standard deviation. Finally, g_{max} is estimated as the maximum period from each factor (Step 10).

Among the most expensive steps of Alg. 2 are the EVD in Step 6 and the matrix inversion in Step 7, both with cubic worst case complexity $O(T^3)$ (assuming $T > m$). However, this cost is incurred only once before PERCeIDs, and is comparable with each iteration of our main algorithm.

VI. Experiments

A. Datasets

We summarize the datasets we use for validation in Table II.

	Statistics			PERCeIDs				LARC [25]			NTF [23]		
Dataset	$\|\mathcal{V}\|$	T	K	$\hat{g}_{max}$	DIV	NMI	Time	DIV	NMI	Time	DIV	NMI	Time
Synthetic	150	200	5	17	**0.03**	**0.98**	5	0.30	0.87	3	0.28	0.82	3
Football	115	1243	12	26	**0**	**1**	5	0.008	0.91	3	0.14	0.77	3
Reality Min.	94	8636	7	23	**0.55**	**0.21**	20	0.65	0.17	40	0.80	0.06	7
Reddit-Episode	242	8636	7	31	**0.80**	**0.004**	15	0.88	0.003	30	0.94	0	10
Reddit-TVshows	3538	1641	6	39	**0.81**	0	76	0.82	0	70	0.96	0	48

TABLE II
DATASET STATISTICS, ESTIMATES OF MAXIMUM PERIOD $\hat{g}_{max}$ PER DATASET BASED ON ALG. 2, AND COMPARISON OF COMMUNITY DETECTION QUALITY (DIV AND NMI) AND RUNNING TIME IN SECONDS FOR PERCEIDS AND COMPETITORS.

Synthetic data: The synthetic data generator has two major components: ground truth soft communities and their temporal profiles. We follow the methodology in [44] to sample temporal interaction events within the communities, namely members in the same community are randomly connected over time, proportional to their matching strength of associations to each of the communities. The temporal community profiles are generated as a periodic signal by following experimental setup by Tenneti et al. [57]. In addition, we injected non-periodic outlier bursts in time of random magnitude and random noisy individual edges among arbitrary pairs (regardless of community membership).

Real-world data: We also evaluate our techniques on three real-world datasets. *Reality Mining* [18] includes proximity-based interactions between roughly 100 friends, lab-mates, and other colleagues at MIT over the course of nine months. This dataset provides high resolution temporal information, which we aggregate at the hourly level. We use unweighted lab group membership as our community ground truth.

Reddit [1] is a set of datasets derived from a dump of all public reddit comments between 2008 and the first half of 2015. We construct datasets from groups of subreddits, which we use as ground truth communities; links are determined by (undirected) replies to comments or top-level posts. *Reddit-Episode* is a smaller dataset of tv shows from the first half of 2009; *Reddit-TVshow* is a set of more developed similar subreddits from the last two months of 2010. Resolution is hourly in all cases.

Football provides a ground truth community membership for a semi-synthetic dataset with generated temporal behavior.

B. Experimental setup

Baselines: We compare against two dynamic community detectors based on tensor factorization: LARC and NTF. LARC [25] is a state-of-the-art overlapping community detection method that combines data reconstruction and smooth temporal activation over time. Non-negative Tensor Factorization (NTF) [24] is a tensor factorization-based method for overlapping community detection employing CP decomposition.

Optimal hyper-parameters for all competing techniques were estimated by grid search for each experimental setting.

Metrics: When datasets have ground truth (GT), we measure the level of agreement between GT and the obtained communities by all methods using the following metrics:

1) Normalized Mutual Information (NMI) [21] adopts the criterion used in information theory to compare the detected communities and the ground-truth communities. This version of NMI is adapted to overlapping communities.

2) Jansen-Shannon divergence (DIV) [61] measures the correspondence between ground-truth and detected results represented as probability distributions over nodes.

We also calculate the area under the ROC (AUC) for recovering ground truth injected bursts in the temporal profile. Implementation of our method in MATLAB is available for download [64].

C. Community detection on synthetic data

In the synthetic datasets, we evaluate the performance of our methods and competitors under a variety of conditions, including different-length observation windows, varying noise, random bursts, and increasing community membership overlap. All comparison results are averages of 5 runs with random per-run initialization presented to each competing technique.

We first vary the observation window to evaluate how much of the timeline needs to be observed for accurate periodic community detection. Due to the existence of periodicity in activity, our method should be able to detect communities without access to the full time length of the data. In the first column of Fig. 3, we demonstrate that the performance of PERCeIDs is more stable and consistently better than alternatives even with a small number of observed time points. However, since both LARC and NTF do not model periods, the increasing observation window gives minimal extra information to improve their results.

We next consider the impact of noise on periodic community detection since noise can interfere with the natural periodicity in the temporal profiles of communities. Results of this experiment are presented in the second column of Fig. 3. To demonstrate our method's robustness to noise, we add random ("salt-and-pepper") noise to the adjacency tensor. In this experiment PERCeIDs similarly has consistently better accuracy than baselines due to the noise robustness of our periodic decomposition of the temporal community profiles.

When increasing the bursty behavior in the activity, we show that our proposed method's accuracy has a significant separation from alternatives as well. This is due to the flexibility introduced by the additive outlier component $\mathbf{O}$, allowing it to faithfully capture outlier time periods while maintaining a high performance in community detection.

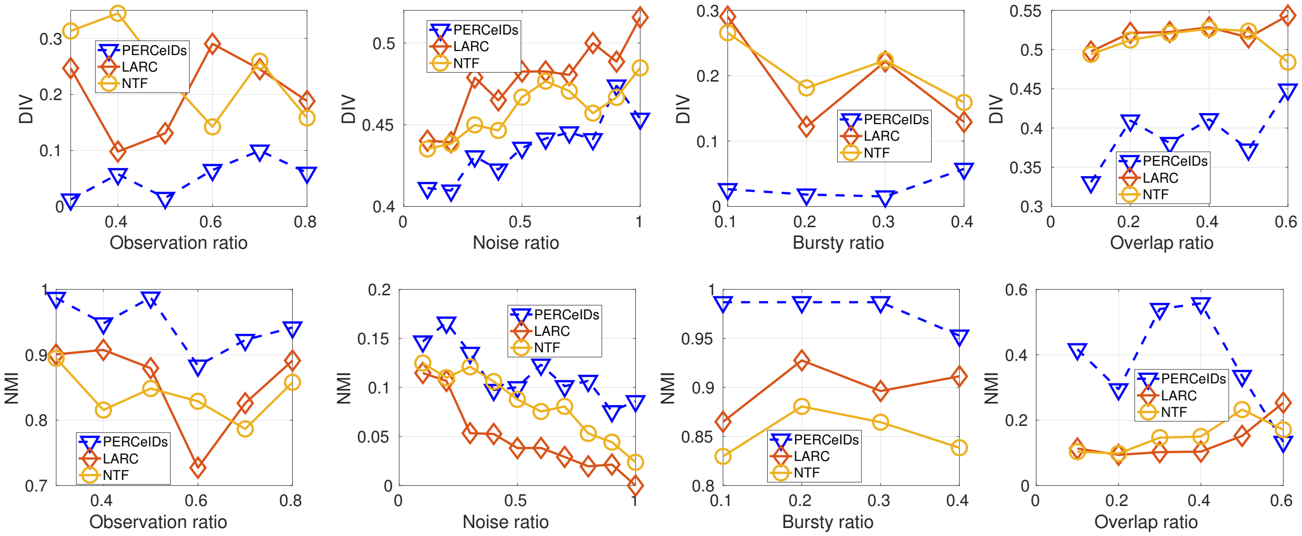


Fig. 3. Community detection comparison of the competing techniques on the synthetic dataset by varying different settings: the first row shows DIV results while the second row shows NMI comparisons. In the first column, we show results for varying length of temporal observation window; in the second column, we show results for increasing random noise added to the adjacency tensor; in the third column, we present the results for varying ratio of bursty outliers added to the data; the last column shows the results for increasing overlap in community membership.

In the last column of Fig. 3, we present the results of varying degrees of overlap in the ground truth communities. Increasing overlap of community members decreases the separability of communities, particularly when the temporal behavior is not properly accounted for. With the presence of (non-periodic) global noise obscuring communities, our method maintains a significant performance improvement over baselines through more than 50% overlap between adjacent communities. The competing methods are unable to effectively separate ground truth communities from noise at any level.

D. Experiments on real world datasets

We also demonstrate the performance of PERCeIDs on a variety of real and modified real/semi-synthetic datasets described above. In table II we show that PERCeIDs outperforms baselines on all presented datasets, improving community detection both in high resolution settings (*Reality Mining*) and larger and sparser datasets (*Reddit-TVshows*).

E. Period analysis

Beyond the analysis of community detection quality, we also evaluate the competing techniques on their ability to estimate the periods of community activity profiles. While our model explicitly estimates the periods in activity, competitors lack this ability. For a fair comparison, we apply the same NPM period estimation method on the temporal profiles obtained by baselines as a post-processing step and term them LARC++ and NTF++.

In Fig. 4, we present the estimated period for all methods of a community in the synthetic data with ground truth hidden periods of 11 and 13. The result from LARC's temporal profile contains multiple erroneous periods, which leads to a significantly large spurious period (obtained as product of the smaller ones). Though NTF finds the ground truth (GT)

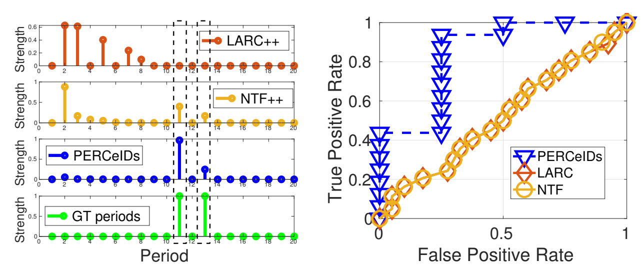


Fig. 4. (a): Period estimation comparison. LARC++ and NTF ++ are the corresponding baseline methods equipped with a post-processing period estimation based on NPM analysis of their temporal factors, akin to the one which we incorporate in PERCeIDs. (b): ROC comparison of known burst (outlier) detection injected in synthetic data.

periods, their strength is weaker than its highest-amplitude period which is spurious. This phenomenon is due to both LARC's and NTF's inability to directly consider periodicity in their models, leading to "wrong" community fits compared to GT and thus inaccurate activity profiles with mixed periods. In comparison, our method successfully detects the ground truth periods of 11 and 13 and they appear as the only significant loadings, i.e. no obvious spurious periods are detected due to the efficient Ramanujan subspace estimation and simultaneous periodicity and membership fitting.

F. Burst (outlier) detection

Our method accounts for intermittent and irregular activity bursts in the temporal profile. To evaluate its effectiveness in detecting such deviations from periodic behavior, we inject bursts (spikes) into the community activity level and calculate the area under the ROC (AUC) for identifying the ground truth burst positions when detecting communities. The results from this evaluation are presented in Fig. 4(b) (this results shows

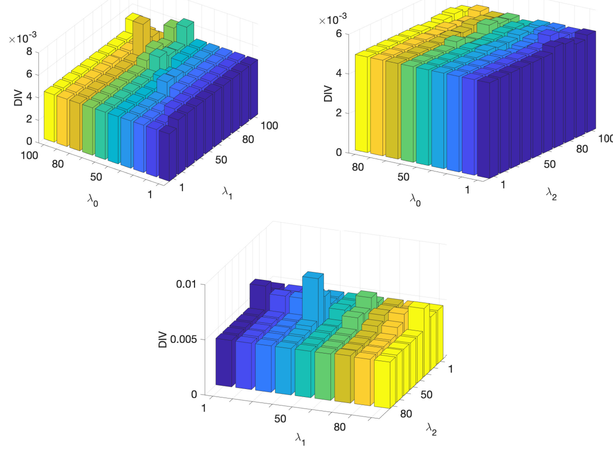


Fig. 5. Sensitivity study of the parameters.

the performance on one community, while other communities have similar behavior).

Since LARC and NTF don't model bursts directly, we rank their detected temporal profiles based on thresholding the temporal profiles they estimate, i.e. high levels are treated as predictions for bursty periods. LARC and NTF perform similarly to each other and not markedly better than random. LARC enforces smoothness in the temporal activity over time, in fact seeking to "smooth" bursts. NTF has no assumption on the temporal activity, but promotes minimum reconstruction error, thus failing to accurately detect outlier-like bursts. Compared to the baselines, PERCeIDs explicitly models pronounced outliers and as a result its corresponding TPR grows at a much faster rate than that of alternatives achieving close to optimal TPR at FPR=0.25.

G. Parameter sensitivity

We also study the sensitivity of PERCeIDs to its hyper-parameters, which include λ_0, λ_1 and λ_2 for temporal profile fitting, periodicity learning and outlier modeling. We fix one parameter and vary the other two and evaluate the DIV in the synthetic data, presented in Fig. 5. It is clear from this evaluation that our method is not sensitive to its parameters within reasonably large ranges of their values $[1, 100]$. For example, the maximal DIV change is less than 0.005 when varying λ_0 and λ_1, with the DIV close to optimal through out this range. We observe similar robustness to parameters in real world data experiments as well.

VII. Conclusion

We proposed a novel algorithm for periodic community detection, termed PERCeIDs. We model the activity of communities as a mixture of hidden periodic signal and outlier bursts over time. Our model is rooted in parsimonious period estimation based on nested periodic matrices with Ramanujan basis, enabling PERCeIDs to detect jointly and faithfully the community membership and periods of activity from raw network interactions without prior knowledge. We demonstrated that our method is robust to noise, outlier non-periodic behavior, and significant membership overlap in ground truth communities via an extensive evaluation in both synthetic and real-world datasets. It consistently outperformed state-of-the-art alternatives with up to two-fold improvement in NMI of ground truth community detection on synthetic data and up to 23% improvement in real-world datasets. In addition, PERCeIDs dominated alternatives on period estimation as well as irregular outlier activity bursts detection and exhibited robustness to its hyper parameters.

VIII. Acknowledgements

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

References

[1] Reddit comments crawl https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment/.

[2] C. C. Aggarwal, Y. Zhao, and P. S. Yu. Outlier detection in graph streams. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ICDE '11, pages 399–409, Washington, DC, USA, 2011. IEEE Computer Society.

[3] R. Ahmed and G. Karypis. Algorithms for mining the evolution of conserved relational states in dynamic networks. In *ICDM*, 2011.

[4] L. Akoglu, M. McGlohon, and C. Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 410–421. Springer, 2010.

[5] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29(3):626–688, 2015.

[6] M. Araujo, S. Günnemann, G. Mateos, and C. Faloutsos. Beyond blocks: Hyperbolic community detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 50–65. Springer, 2014.

[7] M. Araujo, S. Papadimitriou, S. Günnemann, C. Faloutsos, P. Basu, A. Swami, E. E. Papalexakis, and D. Koutra. Com2: fast automatic discovery of temporal ('comet') communities. In *PAKDD*, pages 271–283. Springer, 2014.

[8] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Found. Trends Mach. Learn.*, 4(1):1–106, Jan. 2012.

[9] T. Berger-Wolf, C. Tantipathananandh, and D. Kempe. Dynamic community identification. In *Link Mining: Models, Algorithms, and Applications*, pages 307–336. Springer New York, 2010.

[10] J. K. Bizley, K. M. M. Walker, A. J. King, and J. W. H. Schnupp. Neural ensemble codes for stimulus periodicity in auditory cortex. *Journal of Neuroscience*, 30(14):5078–5091, 2010.

[11] P. Bogdanov, M. Mongiovi, and A. K. Singh. Mining heavy subgraphs in time-evolving networks. In *ICDM*, 2011.

[12] R. Bro. PARAFAC. Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38:149–171, 1997.

[13] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011.

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

[15] S. Deng and J. Han. Ramanujan subspace pursuit for signal periodic decomposition. *CoRR*, abs/1512.08112, 2015.

[16] D. J. DiTursi, G. Ghosh, and P. Bogdanov. Local community detection in dynamic networks. In *In Proc. of IEEE ICDM, New Orleans, USA.*, 2017.

[17] D. J. DiTursi, G. A. Katsios, and P. Bogdanov. Network clocks: Detecting the temporal scale of information diffusion. In *Proceedings of the IEEE International Conference on Data Mining (ICDM), New Orleans, USA.*, 2017.

[18] N. Eagle and A. S. Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006.

[19] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 04 2004.

[20] S. Fortunato and D. Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, 2016.

[21] S. Fortunato and A. Lancichinetti. Community detection algorithms: A comparative analysis: Invited presentation, extended abstract. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '09, pages 27:1–27:2, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[22] N. Gaumont, C. Magnien, and M. Latapy. Finding remarkably dense sequences of contacts in link streams. *Social Network Analysis and Mining*, 6(1):87, 2016.

[23] L. Gauvin, A. Panisson, and C. Cattuto. Detecting the community structure and activity patterns of temporal networks: a non-negative tensor factorization approach. *PloS one*, 9(1):e86028, 2014.

[24] L. Gauvin, A. Panisson, and C. Cattuto. Detecting the community structure and activity patterns of temporal networks: A non-negative tensor factorization approach. *PLOS ONE*, 9(1):1–13, 01 2014.

[25] A. Gorovits, E. Gurjal, V. Papalexakis, and P. Bogdanov. Larc: Learning activity-regularized overlapping communities across time. In *ACM International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 2018)*, 2018.

[26] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. Oxford, fourth edition, 1975.

[27] R. A. Harshman. PARAFAC2: Mathematical and technical notes. *UCLA Working Papers in Phonetics*, 22:30–44, 1972b.

[28] K. Huang, N. D. Sidiropoulos, and A. P. Liavas. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing*, 64(19):5052–5065, Oct 2016.

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

[30] T. Jindal, P. Giridhar, L.-A. Tang, J. Li, and J. Han. Spatiotemporal periodical pattern mining in traffic data. In *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing*, page 11. ACM, 2013.

[31] M.-S. Kim and J. Han. A particle-and-density based evolutionary clustering method for dynamic networks. *VLDB Endow.*, 2, 2009.

[32] D. Koutra, E. E. Papalexakis, and C. Faloutsos. Tensorsplat: Spotting latent anomalies in time. In *2012 16th Panhellenic Conference on Informatics*, pages 144–149. IEEE, 2012.

[33] D. Koutra, J. T. Vogelstein, and C. Faloutsos. Deltacon: A principled massive-graph similarity function. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 162–170. SIAM, 2013.

[34] M. Lahiri and T. Y. Berger-Wolf. Mining periodic behavior in dynamic social networks. In *2008 Eighth IEEE International Conference on Data Mining*, pages 373–382. IEEE, 2008.

[35] M. Lahiri and T. Y. Berger-Wolf. Periodic subgraph mining in dynamic networks. *Knowledge and Information Systems*, 24(3):467–497, 2010.

[36] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *WWW*, 2010.

[37] 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*, KDD '10, pages 1099–1108, New York, NY, USA, 2010. ACM.

[38] 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.

[39] S. Liu, S. Wang, and R. Krishnan. Persistent community detection in dynamic social networks. In *PAKDD*, pages 78–89. Springer, 2014.

[40] X. Ma and D. Dong. Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks. *IEEE Transactions on Knowledge and Data Engineering*, 29(5):1045–1058, May 2017.

[41] C. Matias and V. Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society Series B*, 79(4):1119–1141, 2017.

[42] M. Mongiovi, P. Bogdanov, R. Ranca, A. K. Singh, E. Papalexakis, and C. Faloutsos. Netspot: Spotting significant anomalous regions on dynamic networks. In *SDM*, 2013.

[43] M. Mongiovi, P. Bogdanov, and A. K. Singh. Mining evolving network processes. In *ICDM*, 2013.

[44] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review*, E 69(026113), 2004.

[45] N. P. Nguyen, T. N. Dinh, Y. Shen, and M. T. Thai. Dynamic social community detection and its applications. *PLOS ONE*, 9(4):1–18, 04 2014.

[46] G. Palla, A.-L. Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446:664–667, 2007.

[47] Y. Pei, N. Chakraborty, and K. Sycara. Nonnegative matrix tri-factorization with graph regularization for community detection in social networks. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2083–2089. AAAI Press, 2015.

[48] A. Rezayee and S. Gazor. An adaptive klt approach for speech enhancement. *IEEE Trans. Speech Audio Processing*, 9:87–95, 1999.

[49] P. Rozenshtein, N. Tatti, and A. Gionis. Discovering dynamic communities in interaction networks. In *Proceedings of ECML/PKDD*, 2014.

[50] G. Rustici, J. Mata, K. Kivinen, P. Lió, C. J. Penkett, G. Burns, J. Hayles, A. Brazma, P. Nurse, and J. Bähler. Periodic gene expression program of the fission yeast cell cycle. *Nature genetics*, 36(8):809, 2004.

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

[52] N. Shah, D. Koutra, T. Zou, B. Gallagher, and C. Faloutsos. Timecrunch: Interpretable dynamic graph summarization. In *KDD*, pages 1055–1064, 2015.

[53] J. Sun, C. Faloutsos, S. Papadimitriou, and P. Yu. Graphscope: Parameter-free mining of large time-evolving graphs. In *KDD-2007*, pages 687–696, 12 2007.

[54] L. Tang, H. Liu, and J. Zhang. Identifying evolving groups in dynamic multimode networks. *IEEE Trans. on Knowl. and Data Eng.*, 24(1):72–85, Jan. 2012.

[55] W. Tang, Z. Lu, and I. S. Dhillon. Clustering with multiple graphs. In W. Wang, H. Kargupta, S. Ranka, P. S. Yu, and X. Wu, editors, *ICDM*, pages 1016–1021. IEEE Computer Society, 2009.

[56] S. V. Tenneti and P. P. Vaidyanathan. Minimal dictionaries for spanning periodic signals. In *49th Asilomar Conference on Signals, Systems and Computers, ACSSC 2015, Pacific Grove, CA, USA, November 8-11, 2015*, pages 523–527, 2015.

[57] 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.

[58] R. J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *Ann. Statist.*, 39(3):1335–1371, 06 2011.

[59] 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.

[60] Y. Wang, A. Chakrabarti, D. Sivakoff, and S. Parthasarathy. Hierarchical change point detection on dynamic networks. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 171–179. ACM, 2017.

[61] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 261–270, New York, NY, USA, 2010. ACM.

[62] J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys (csur)*, 45(4):43, 2013.

[63] J. Yang and J. Leskovec. Overlapping community detection at scale. In *Proceedings of the sixth ACM international conference on Web search and data mining - WSDM '13*, page 587, New York, New York, USA, 2013. ACM Press.

[64] L. Zhang, A. Gorovits, and P. Bogdanov. MATLAB implementation for PERCeIDs: http://www.cs.albany.edu/%7Epetko/lab/code.html.