LARC: Learning Activity-Regularized Overlapping Communities Across Time

Alexander Gorovits*
Department of Computer Science
University at Albany—SUNY
agorovits@albany.edu

Ekta Gujral
Department of Computer Science and Engineering
University of California Riverside
egujr001@ucr.edu

Evangelos E. Papalexakis
Department of Computer Science and Engineering
University of California Riverside
epapalex@cs.ucr.edu

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

ABSTRACT

Communities are essential building blocks of complex networks enjoying significant research attention in terms of modeling and detection algorithms. Common across models is the premise that node pairs that share communities are likely to interact more strongly. Moreover, in the most general setting a node may be a member of multiple communities, and thus, interact with more than one cohesive group of other nodes. If node interactions are observed over a long period and aggregated into a single static network, the communities may be hard to discern due to their in-network overlap. Alternatively, if interactions are observed over short time periods, the communities may be only partially observable. How can we detect communities at an appropriate temporal resolution that resonates with their natural periods of activity?

We propose LARC, a general framework for joint learning of the overlapping community structure and the periods of activity of communities, directly from temporal interaction data. We formulate the problem as an optimization task coupling community fit and smooth temporal activation over time. To the best of our knowledge, the tensor version of LARC is the first tensor-based community detection method to introduce such smoothness constraints. We propose efficient algorithms for the problem, achieving a $2.6x$ quality improvement over all baselines for high temporal resolution datasets, and consistently detecting better-quality communities for different levels of data aggregation and varying community overlap. In addition, LARC elucidates interpretable temporal patterns of community activity corresponding to botnet attacks, transportation change points and public forum interaction trends, while being computationally practical—few minutes on large real datasets. Finally, LARC provides a comprehensive *unsupervised* parameter estimation methodology yielding high accuracy and rendering it easy-to-use for practitioners.

*Corresponding Author.

KDD 2018, August 19–23, 2018, London, United Kingdom
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5552-0/18/08...$15.00
https://doi.org/10.1145/3219819.3220118

KEYWORDS

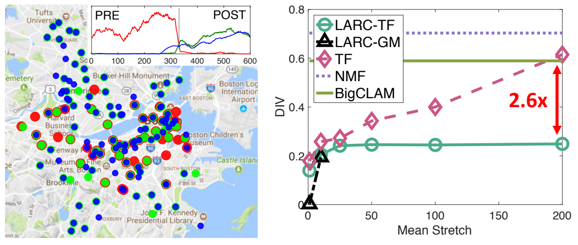
Dynamic graphs; overlapping community detection; tensor factorization; community activation

ACM Reference Format:
Alexander Gorovits, Ekta Gujral, Evangelos E. Papalexakis, and Petko Bogdanov. 2018. LARC: Learning Activity-Regularized Overlapping Communities Across Time. In *KDD 2018: 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3220118

1 INTRODUCTION

Mapping the community structure of a network is essential for understanding its underlying system and processes. Social circles, gene pathways and cliques of scientific collaborators are all examples of functional units in large networks that can be modeled as overlapping communities. Common approaches for community detection rely on a static network [22, 49, 50], however, temporal interaction data is becoming increasingly available and, thus, holds the potential to inform better community detection methods. Consider, for instance, a community among professionals in the workplace, in which communication is likely to occur during work hours and in week days. If we attempt to detect work groups from communications within this network, we would have to discover and take into account this community activation pattern. The problem becomes even more challenging when a node communicates with connections in other (non-professional) circles, e.g. family members and friends, within the same communication medium and with varying intensity and temporal resolution. *How can we exploit the timings of these interactions by enforcing appropriate smoothness on their time course to tease apart overlapping communities and their activity periods?*

An illustrative example of communities and their activity detected by our methods in bike trips data from Boston, MA is presented in Fig. 1(a). Nodes in this network are bike rental stations and trips between stations (check-out to drop-off) are modeled as temporal interactions. We detect the overlapping structure of groups of stations with heavy within-group traffic in contiguous intervals. The moving activation average over almost 2 years (top-right in Fig. 1(a)) reveals an important change point of expanding the bike rental service from downtown Boston (*red*) to the outskirts resulting



(a) Boston Bike Rides Communities (b) Quality with Temporal Spread

Figure 1: (a) A visualization of 3 color-coded groups of bike rental stations in Boston that observe strong within-group traffic in contiguous intervals of time detected by our algorithm LARC. Beyond the overlapping community structure, we identify interpretable community activity profiles (visualized top-right) elucidating a timepoint of the service expansion as well as university break dips. (b) LARC's quality in detecting ground truth (GT) communities is consistently and up to $2.6x$ superior to baselines (lower divergence (DIV) is better) when the temporal interactions are heterogeneously spread in time—simulating data sampled at high temporal resolution (V=200,T=100…20k).

in two previously inactive communities: *green*—rides within downtown and popular new stations (Harvard and MIT); and *blue*— rides including the outskirts. Such an analytic tool can inform activity-aware improvements by city planners as well as resource provisioning for the the bike rental service provider. Other real-world applications abound: resource provisioning in computer networks, botnet detection and activity analysis, and time-aware advertising within social network communities to name a few. Our methods are designed to handle the temporal activity heterogeneity, and thus consistently outperform baselines in detecting ground truth communities Fig. 1(b) with an increasing gap (up to $2.6x$) when community interaction events are spread randomly over increasing intervals (high temporal resolution).

Our goal in this paper differs from evolutionary clustering [7, 30] where the objective is to study the long-term evolution (growth, splits, merges) of network clusters. Instead, we focus on the short-term (repeated) activity within relatively stationary communities, a setting in which the interactions within the communities are much more common than changes in the overall community structure. Considering temporal information, however, is a knife that cuts both ways. If aggregated inadequately, temporal network data can reveal unrealistic patterns [13, 14, 17, 45]. Furthermore, the appropriate window for aggregation may not be uniform for the whole timeline [18]. Hence, in order to capitalize on temporal network information, it is important to jointly learn an appropriate temporal resolution of the data as well as the actual community structure. Community detection is notoriously hard under multiple popular measures including conductance [17], modularity [19] and ratio cut [47]. Allowing for overlap among communities makes the search space even larger [49]. Furthermore, employing the dynamic interaction behavior—our goal in this paper—adds yet another dimension to the already computationally challenging problem.

We propose a general framework for *Learning Activity-Regularized Communities (LARC)* which detects jointly the overlapping community structure and the activity periods of communities, directly from temporal network data. Our framework simultaneously optimizes community fit and activation profiles that are close to piece-wise constant functions, thus ensuring interpretability, resilience to noisy interactions, and temporal oversampling. To quantify community fit we consider both reconstruction and generative models and derive efficient solvers for the resulting optimization problems. In addition, we derive *unsupervised* solutions for automatic selection of the number of communities based on activation-aware core consistency diagnostic as well as for parameter learning based on the minimum description length (MDL) principle. Our solutions scale to large real-world instances and can be employed in a variety of applications.

Our contributions in this work are as follows:
• **Novel Problem Formulation**: We propose the novel problem of jointly learning the overlapping community structure and smooth activation profiles from dynamic interaction data.
• **Flexible Optimization Framework**: We develop a general framework LARC for the problem imposing fused lasso regularization on the activation profiles and demonstrate that it can be coupled with both generative and reconstruction models for community fit.
• **Real-World Utility**: Our extensive evaluation on synthetic and real-world data demonstrates LARC's $2.6x$ quality improvement over baselines in high temporal resolution, and superior quality with temporal aggregation and varying community overlap. LARC also reveals botnet attack activity and transportation change points. It remains practical—completes in minutes—for large datasets.

LARC's implementation, data generator and evaluation datasets are available at http://www.cs.albany.edu/~petko/lab/code.html.

2 BACKGROUND AND RELATED WORK

Static communities: Our work is different from static (overlapping) community detection [21, 22, 34, 49, 51] in that we consider dynamic interactions to improve the quality and also provide an interpretable temporal activation profile for each community. Furthermore, our goal is complementary to static community detection approaches, as objective functions from the static setting can be generalized within our framework to the dynamic one. We demonstrate this for the affiliation generative model in [50]. In addition, we compare to static methods by aggregating the temporal interactions and demonstrate that utilizing the temporal information is advantageous for recovering ground truth communities.
Temporal communities: Various temporal subgraph detection methods have also been considered in the literature: communities [17, 25], dense subgraphs [15, 24, 42, 43], heavy-weight subgraphs [8, 37] and persistent subgraphs [3, 36]. Many of these methods detect one subgraph over one "active interval" at a time, as opposed to recurring activation of multiple subgraphs [3, 8, 17, 36, 37]. These methods do not consider overlap, cannot ensure stable community membership if iteratively run to extract multiple communities, and are sensitive to the temporal resolution (aggregation) of the interaction data. Other methods enforce user-defined consistency by introducing parameters such as number of occurrences and time span [43] or some notion of persistence (e.g. time-to-live

interval) for interaction edges [3, 15, 24, 42]. Different from all the above, we detect multiple overlapping communities over time and let the data define the natural periods of activity which may vary with communities, application and across time. Finally, Gauvin et al. [25] detect both overlapping and data-driven dynamics of communities using tensor factorization (TF), however, as we show in our evaluation, employing TF without activity regularization is sensitive to the temporal resolution, and thus, results in sub-par quality.

Evolving communities: The goal in evolutionary clustering [7, 30] and evolving community and heavy subgraph detection [16, 38] is different from ours as they characterize how community membership changes in the long term. Instead we focus on detecting the stationary overlapping community structure in a short time frame in which membership is relatively stable.

Tensor methods: Tensor factorization (TF) has also been employed to detect communities from temporal [4, 25, 35] and multi-view [27, 40] network data.To the best of our knowledge, this is the first work that enforces temporal smoothness constraints in the factorization model, in order to uncover more accurate communities. We demonstrate how tensor-based reconstruction error can be used as a goodness-of-fit metric in our framework and further extend tensor factorization approaches to handle our activity regularization objective. We compare LARC-TF experimentally with popular TF approaches, and demonstrate its superior performance.

3 PROBLEM FORMULATION

We next formalize the problem of joint detection of communities and their activity profiles. We expect that, as time progresses, communities (i) exhibit relatively stationary membership and (ii) alternate between active (multiple internal interactions) and inactive states. Thus, interactions in time can be utilized to improve detection as communities will be more discernible at an appropriate temporal scale as opposed to in a fully-aggregated graph. We represent the observed symmetric interactions among a finite node set V, $|V| = n$ over a finite interval of discrete time steps $[1, T]$ as a 3-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}_{\geq 0}^{n \times n \times T}$, where each element is a value modeling the number of interactions between a pair of nodes at a given time. The tensor face of observed interactions at time t is denoted as $X(t)$ and can be viewed as a weighted undirected network snapshot for that time. Let $C \in \mathbb{R}_{\geq 0}^{n \times k}$ be a *community matrix* specifying the strength of affiliation of nodes to each of k communities. Let also $A \in \mathbb{R}_{\geq 0}^{t \times k}$ be an *activation matrix*, modeling the activation profiles of communities over time. A high value of A_{tk} denotes high level of activity of community k at time t, i.e. many internal interactions.

Let $J(\boldsymbol{\mathcal{X}}, C, A, k)$ be an *error-of-fit* function for k communities for given C and A matrices. Our goal is to minimize J while enforcing contiguous active and inactive periods for each community. To this end, we impose a smoothness regularization on the community activation profiles, i.e. the columns of A. In particular, we incorporate a fused lasso [46] regularizer, which has been shown effective for piece-wise constant signal approximations [28]. In our setting we enforce *shrinkage* by an L_1 penalty on the activation matrix A and *total-variation denoising* by an L_1 penalty on the difference of consecutive values in A's columns. Formally, our regularization has the following matrix form:

$$R(A) = \lambda_c ||A||_1 + \lambda_d ||DA||_1, \tag{1}$$

where D is a row-wise difference matrix with $D_{i,i} = -1, D_{i,i+1} = 1$ and 0 for all other elements, and λ_c and λ_d are regularization parameters controlling the importance of the regularization objectives.

Definition 3.1. **Dynamic Overlapping Communities:** Given a dataset $\boldsymbol{\mathcal{X}}$, number of communities k, λ_c and λ_d, solve:

$$\min_{A,C} J(X, C, A, k) + R(A). \tag{2}$$

The error fit function J can be instantiated based on different models reflecting how "good overlapping comunities" should manifest in terms of inter-node interactions. Next, we demonstrate two possible realizations for J based on (i) tensor factorization (TF) and (ii) a temporal extension of the affiliation generative model (GM).

3.1 Tensor factorization model (TF)

Since the observed interaction data $\boldsymbol{\mathcal{X}}$ in our setting is a tensor, tensor factorization models are a natural fit for learning the communities C and temporal profiles A as corresponding tensor factors. We focus on CANDECOMP/PARAFAC [11], which decomposes a tensor into a sum of rank-one tensors $\boldsymbol{\mathcal{X}} \approx \sum_k c_k \circ c'_k \circ a_k$, which we also denote in matrix form as $\boldsymbol{\mathcal{X}} \approx [\![C, C', A]\!]$, where C' is equivalent to our community matrix C and A holds the temporal dimension factors corresponding to our activation matrix. Note that since temporal snapshots $X(t)$ are symmetric matrices, solutions for community factor matrices will be equivalent $C \approx C'$, and thus. we further simplify the reconstruction notation to $\boldsymbol{\mathcal{X}} \approx [\![C, A]\!]$. PARAFAC minimizes the factorization reconstruction error for a fixed number of factors k, which we adopt as error-fit function:

$$J_{TF}(\boldsymbol{\mathcal{X}}, C, A, k) = ||\boldsymbol{\mathcal{X}} - [\![C, A]\!]||_F^2. \tag{3}$$

3.2 Affiliation generative model in time (GM)

Probabilistic affiliation generative models (GM) [10, 32] provide an alternative to reconstruction (TF) models. In static network GMs, nodes have affiliations to communities C which drive randomly observed connections among affiliated nodes [50]. Here, we extend this model to overlapping communities over time, arriving at an alternative error-of-fit function J_{GM}. Specifically, we define the probability of observing an edge (i, j) at time t due to community k as: $P_k(i, j, t) = 1 - e^{-C_{ik} A_{tk} C_{jk}}$. Note, that if the temporal profile of the community is fixed $A_{:k} = 1$, the likelihood of interaction reduces to that in the static affiliation model [50]. The temporal profile A acts as a selector enabling interactions when the community is active ($A_{tk} > 0$) and inhibiting them during periods of inactivity. The probability of interaction due to any community is then:

$$P(i, j, t) = 1 - \prod_k (1 - P_k(i, j, t)) = 1 - e^{-C_i A(t) C_j^T}, \tag{4}$$

where $A(t) = diag(A_{t:})$ is a diagonal matrix of community activation states at time t. Assuming independence of interactions, the conditional likelihood $L(\boldsymbol{\mathcal{X}}|C, A)$ of observing $\boldsymbol{\mathcal{X}}$ is then:

$$L(\boldsymbol{\mathcal{X}}|C, A) = \prod_t \prod_{\boldsymbol{\mathcal{X}}(i,j,t) \neq 0} P(i, j, t) \prod_{\boldsymbol{\mathcal{X}}(i,j,t)=0} [1 - P(i, j, t)]. \tag{5}$$

Note that the basic probabilistic affiliation model, unlike the reconstruction model (TF), assumes binary data, i.e. interactions are either 1 or 0. To make it applicable to non-binary data, one can consider thresholding schemes. We define the GM error-of-fit function based on the log-likelihood $LL(\boldsymbol{\mathcal{X}}|C,A)$ of observed interactions:

$$J_{GM}(\boldsymbol{\mathcal{X}},C,A,k) = -LL(\boldsymbol{\mathcal{X}}|C,A) =$$
$$-\sum_t \Big[\sum_{\boldsymbol{\mathcal{X}}(i,j,t)=0} \log\big(1-e^{-C_i A(t) C_j^T}\big) - \sum_{\boldsymbol{\mathcal{X}}(i,j,t)\neq 0} C_i A(t) C_j^T \Big].$$

4 ALGORITHMS

In this section, we propose solvers that minimize the two objectives corresponding to the error-of-fit functions J_{TF} and J_{GM} combined with the same fused lasso regularization term $R(A)$. As we demonstrate experimentally, both models show comparable quality in recovering ground truth communities that is superior to baselines due to their shared regularization approach which explicitly models smooth active/inactive community behavior. We further derive estimators for (i) the optimal number of communities k and (ii) the regularization parameters λ_c and λ_d.

4.1 Regularized Tensor Factorization LARC-TF

The objective for our reconstruction model is as follows:

$$\min_{C,A} ||X - [\![C,A]\!]||_F^2 + \lambda_c||A||_1 + \lambda_d||DA||_1. \quad (6)$$

It can be viewed as a regularized tensor factorization problem in which we have imposed a fused lasso penalty $R(A)$ to enforce piecewise-constant and sparse solution for the time path $A_{:i}$ of each community. Furthermore, in our solutions for Eq. 6 we seek to obtain non-negative factors C and A as they model affiliation and temporal activation respectively, thus ensuring interpretability within our problem. We extend a commonly-used non-negative factor method for solving the PARAFAC problem, namely alternating least squares (ALS) with non-negative factors [11]. An ALS solution for PARAFAC keeps two factors fixed, and takes advantage of the convexity and existence of a closed-form analytical solution for the third. Iterative updates for any factor U_1, assuming the remaining two U_2 and U_3 are fixed, have the following form:

$$U_1 \leftarrow \arg\min_U ||X_{(1)} - (U_3 \odot U_2)U||_F^2, \quad (7)$$

where $X_{(1)}$ is the tensor unfolding on the updated dimension, the factors U_i correspond to our community C or activation A matrices, and $\odot$ is the Khatri-Rao product [29]. We cannot use this framework directly as we need to incorporate the fused lasso regularization $R(A)$. Our formulation, however, retains some of the advantageous properties allowing an ALS-like solution, namely simplicity of updates and convexity.

To solve the updates efficiently, we adopt the Alternating Direction Method of Multipliers (ADMM) which has recently been employed in a number of high-dimensional large-scale problems for efficiently utilizing batch updates typically occurring in ALS [9]. In particular, we devise an Alternating Optimization ADMM (AOADMM) which combines the alternating least squares and the ADMM framework [29]. Intuitively, the main idea is to divide the problem into simpler-to-update blocks before reconciling these partial solutions.

Algorithm 1 LARC-TF

```
Require: Tensor $\boldsymbol{\mathcal{X}}$, number of factors $k$, regularization parameters $\lambda_c$, $\lambda_d$
Ensure: Community $C \approx C'$ and activation $A$ matrices.
 1: Initialize $C$, $C'$, $A$ randomly
 2: Initialize residual matrices $R_C$, $R_{C'}$, $R_A$ to 0
 3: while The factors $C$, $C'$, $A$ have not converged do
 4:     for Each factor $H$ in $\{C, C', A\}$ do
 5:         Let $R_H$ and $dim(H)$ be the residual and tensor dimension of factor $H$
 6:         Let $H_1$ and $H_2$ be the other two fixed factors
 7:         $\breve{H} \leftarrow H1 \odot H2$
 8:         $\rho \leftarrow tr(\breve{H}^T\breve{H})$
 9:         $L \leftarrow$ Lower Cholesky decomposition of $(\breve{H}^T\breve{H} + \rho I)$
10:         $F$ = MTTKRP($\boldsymbol{\mathcal{X}}$, $\breve{H}$, $dim(H)$)
11:         while Not converged do
12:             $\tilde{H} \leftarrow (L^T)^{-1}L^{-1}(F + \rho(H + R_H))$        ▷ Optimized Eq. 9
13:             $H \leftarrow proxOpp(H, \tilde{H}, R_H, \rho, dim(H))$
14:             $R_H \leftarrow R_H + H - \tilde{H}^T$        ▷ Eq. 11
15:         end while
16:     end for
17: end while
18: return $C$, $C'$, $A$
```

We enforce a non-negativity constraint on the updates of both community factors C, C' and design a custom update for the activation factor A that handles the fused lasso penalty and also enforces non-negativity. The objective in our ADMM update for A is to solve the following convex sub-problem (adding $R(A)$ maintains convexity):

$$\min_{A,\tilde{A}} ||X_{(3)} - \breve{C}\tilde{A}||_F^2 + \lambda_c||A||_1 + \lambda_d||DA||_1 \text{ s.t. } A = \tilde{A}^T, A \geq 0, \quad (8)$$

where $X_{(3)}$ is the tensor unfolding on the third temporal dimension, $\breve{C} = C' \odot C$ is the Khatri-Rao product of the community factors and $\tilde{A}$ is an auxiliary ADMM variable used to update A. The minimization can be solved by iterating over the following update sequence:

$$\tilde{A} \leftarrow (\breve{C}^T\breve{C} + \rho I)^{-1}(\breve{C}^T X_{(3)} + \rho(A + R_A)^T) \quad (9)$$

$$A \leftarrow \arg\min_A \lambda_c||A||_1 + \lambda_d||DA||_1 + \rho/2||R_A + A - \tilde{A}^T||_F^2 \quad (10)$$

$$R_A \leftarrow R_A + A - \tilde{A}^T, \quad (11)$$

where $\rho = tr(\breve{C}^T\breve{C})$ is the trace of the Khatri-Rao product of the community factors and its transpose and R_A is a running residual matrix for factor A. The first "fit" update (Eq. 9) and third "residual" update (Eq. 11) are common for all factors (i.e. C and C' as well as A) and there exist fast solutions for them based on Lower Cholesky decomposition and Matricized Tensor Times Khatri-Rao Product (MTTKRP) in non-regularized ADMM methods [29] detailed further in the Alg. 1. The second "regularization update" (Eq. 10) is also referred to as the *proximity operator (proxOpp)* of the trace-scaled regularization function $1/\rho R(A)$. While it involves minimization of a convex function of A, there is no closed-form analytic solution for it, so we employ coordinate descent with a non-negativity constraint for this step.

Algorithm 1 shows the steps of our AOADMM approach LARC-TF for fused lasso tensor factorization. After initialization of the factors and their corresponding residual matrices (Steps 1,2), we iteratively update the factors one at a time while keeping the other two fixed (Steps 3-17) until convergence. In the ADMM update step for each factor H (Steps 4-16) we first pre-compute several matrices and scalars that let us speed up the fit update from Eq. 9. Namely, the Khatri-Rao product of the fixed factors $\breve{H}$ (Step 7); the trace ρ (Step 8); a lower Cholesky decomposition of the first inverted matrix $(\breve{H}^T\breve{H} + \rho I)$ of Eq. 9 (Step 9), and the MTTKRP (Step 10). Note, that all the above are constant during the repeated updates of $H, \tilde{H}, R_H$, and thus precomputing them only once saves time. In the updates of H (Step 13) we employ the appropriate proximity operator for each factor. For factors C, C', since we enforce non-negativity the proximity operator simply replaces negative elements with 0, i.e. element-wise $max(H, 0)$. If the update is for the activity factor A, however, we need to solve the minimization problem in Eq. 10. To this end, we perform a coordinate descent with line search to determine an appropriate learning rate β, where the main update is along the gradients for each timestep:

$$\partial / \partial A_t = \lambda_c \mathbf{1} + \lambda_d (sgn(A_t - A_{t-1}) - sgn(A_{t+1} - A_t)) \\ + \rho(R_A + A_t - \tilde{A}_t),$$

where A_t is a short hand for the t-th row of A, and $sgn()$ is the element-wise sign function setting elements to $\{+1, -1\}$ depending on their sign. Non-negative projection, similar to those for C and C', is also applied at the end of the gradient descent for A.

The gradient descent in Step 13 has the highest computational footprint, being nested in two convergence loops and further depending on T to ensure smoothness of A's columns. However, all operations preserve sparsity and, thus, are expected to scale almost linearly with the sizes of the input. Additionally, allowing a fixed update size instead of a full line search results in significant speedup at minimal quality expense. Our experimental evaluation reveals that with increasing T the number of iterations for A's convergence grows slightly, however, the overall running time remains practical for our largest instances.

4.2 Affiliation Model Solver LARC-GM

To minimize the objective $f_{GM} = J_{GM}(X, A, C, k) + R(A)$, we consider block coordinate descent methods. The gradient with respect to a community membership vector C_i is:

$$\frac{\partial f_{GM}}{\partial C_i} = \sum_t \Big(\sum_{\mathcal{X}(i,j,t)=1} C_j A(t) \frac{e^{-C_i A(t) C_j^T}}{1 - e^{C_i A(t) C_j^T}} - \sum_{\mathcal{X}(i,j,t)=0} C_j A(t) \Big).$$

Similarly, differentiating with respect to A_{tk}, we get:

$$\frac{\partial f_{GM}}{\partial A_{tk}} = \sum_{\mathcal{X}(i,j,t)=1} \frac{C_{ik} C_{jk} e^{-C_i A(t) C_j^T}}{1 - e^{-C_i A(t) C_j^T}} - \sum_{\mathcal{X}(i,j,t)=0} C_{ik} C_{jk} \\ -\lambda_s - \lambda_d (sgn(A_{tk} - A_{(t+1)k}) - sgn(A_{(t+1)k} - A_{tk})),$$

which can be combined into a single update for the block $A_{t:}$. Direct coordinate or gradient descent will not scale for large instances $\mathcal{X}$, hence we seek to scale our solutions by avoiding re-computation of the full gradient. Particularly, similar to the static network affiliation model solutions [50], only a small number of elements are updated in the unobserved edges component. Hence, one can re-write the no-edge portion of the update as:

$$\sum_{\mathcal{X}_{i,j,t}=0} C_j A(t) = \sum_j C_j A(t) - C_i A(t) - \sum_{j \in N_i^t} C_j A(t), \qquad (12)$$

where N_i^t is the set of neighbors of i (i.e. nodes with which i interacted) at time t. We can thus, compute and store $\sum_j A(t) C_j$ at each iteration (over all i), leading to faster updates of C_i over the much smaller set of neighbors. A similar approach can be adopted to speed up A's gradient as well. First, we notice that we can update

Algorithm 2 LARC-CCD: Detect k^* with Time-Warped CCD

```
Require: Tensor $\mathcal{X}$, factorization $[C, C', A]$ produced by LARC
Ensure: Activity-aware optimal $k^*$
 1: $[U, \Sigma, V] \leftarrow SVD(A)$
 2: for r=1: k  do
 3:     $\mathbf{\Pi}_r = U(:, 1:r) * U(:, 1:r)^T$
 4:     $\mathcal{X}_r = \mathcal{X} \times_3 \mathbf{\Pi}_r$
 5:     $A_r = \mathbf{\Pi}_r * A$
 6:     c(r) = efficient_corcondia($\mathcal{X}_r$,$C$, $C'$, $A_r$,$\mathbf{1}_k$)
 7: end for
 8: $k^*$ = AutoTen(max(c))
 9: return $k^*$
```

either entire faces (fixed t) or communities (fixed k) at a time. In the first case, the $\sum_{i,j \notin X(t)} C_{ik} C_{jk}$ term can be similarly decomposed by storing $\sum_j C_j k$ in vector form for all k, as $C_\Sigma = \sum_j C_j$. We can then iterate over i (instead of i, j) and compute:

$$\frac{\partial f_{GM}}{\partial A(t)} = -\lambda_s I - \lambda_d * (sgn(A(t) - A(t-1)) - sgn(A(t+1) - A(t))) \\ + diag\Big(\sum_i (\sum_{j \in N_i^t} \frac{C_i \circ C_j e^{-C_i A(t) C_j}}{1 - e^{-c_i A(t) C_j}} - C_i \odot (C_\Sigma - C_i - \sum_{j \in N_i^t} C_j)) \Big),$$

where $\circ$ denotes the Hadamard (element-wise) product of the two matrix rows, and $diag()$ is the diagonal matrix of the argument vector. LARC-GM then iterates between coordinate descent steps on for C and A using also a line search for an appropriate learning rate. The update optimizations in LARC-GM increase its speed compared to direct coordinate descent significantly, however, as we show experimentally LARC-TF scales much better than than LARC-GM, while they both produce better quality communities compared to baselines.

4.3 Learning the number of communities k

An important question when analyzing a new dataset is how to set k. We extend TF consistency approaches to our activity regularized objectives and develop a method LARC-CCD for selecting k which outperforms regularization-oblivious alternatives. Finding the rank of a tensor is a NP-hard problem, however, there exist heuristic models such as the *Core Consistency Diagnostic (CCD)* algorithm [12, 41]. Given a tensor $\mathcal{X}$ and its PARAFAC factorization $[C, C', A]$, CCD provides a number indicative of the factorization quality, thus allowing for selection of maximum number of good-quality communities [39]. In our case, however, using CCD as a black-box may lead to bad estimates of k as our solution's factorization $[C, C', A]$ is a product of an activity smoothness regularization, which as we demonstrate experimentally, is different from no-regularization PARAFAC factorization.

In order to make CCD amenable to our regularization, we need to "compress" the temporal mode of the tensor in a way that respects the temporal smoothness discovered by LARC. If A yields very smooth latent factors, we aim to compress the temporal mode accordingly, so that we adjust the tensor to that smoothness. The key to that compression is the row space of the activity matrix A: if there exists a subspace of that row space which, when we project both the tensor and matrix $\mathbf{A}$, yields a higher core consistency than simply using the computed factors and the uncompressed tensor, we choose that subspace to generate the core consistency that characterizes the quality of our solution.

	Statistics				k % Deviation		LARC-TF			TF [25]			NMF [33]			BigCLAM [50]		
Dataset	**|V|**	**T**	**nnz**	**k**	**LARC-CCD**	**AutoTen[39]**	**DIV**	**NMI**	**time**	**DIV**	**NMI**	**time**	**DIV**	**NMI**	**time**	**DIV**	**NMI**	**time**
Synthetic	200	$20k$	$61k$	5	32%	71%	**0.24**	**1**	62	0.61	0.24	5	0.72	0.13	8	0.58	0.09	2
Football	115	$2k$	$17k$	12	13%	42%	0.08	0.91	38	0.14	0.77	28	0.64	0.1	1	**0**	**1**	1
Reality Min.	94	$8k$	$0.1m$	5	12%	56%	**0.62**	0.03	30	0.66	0.03	6	0.66	0.03	1	0.80	**0.08**	1
Reddit-sports	$120k$	267	$1.1m$	15	27%	57%	**0.53**	**0.27**	57	0.62	0.11	41	0.67	0.19	131	.90	0.02	40
Reddit-news	$140k$	267	$0.7m$	5	21%	40%	**0.35**	**0.26**	13	0.36	0.21	10	0.48	0.22	54	0.77	0.14	16
Bike Rides	145	628	$0.8m$	-	11%*	45%*	-	-	14	-	-	14	-	-	1	-	-	1
Botnet	$20k$	$6k$	$0.5m$	-	20%*	20%*	-	-	10	-	-	1	-	-	1	-	-	3

Table 1: Dataset statistics (cols 1-5) and success in the estimation of k in datasets with ground truth (GT) communities (cols 6-7). Comparison of quality (DIV and NMI) and running time in seconds for all competing methods and datasets (cols 8-19). Note that, due to the lack of GT communities in the *Bike Rides* and *Botnet* datasets, only running time and variance in the estimation of k is reported.

Algorithm 2 summarizes our method for learning k called LARC-CCD. Given $\mathcal{X}$ and a candidate decomposition $[C, C', A]$, we first compute the the Singular Value Decomposition (SVD) of $A = U\Sigma V^T$, where U is a basis for A's row space. Then we quantify the activity-aware CCD $c(r)$ for all $r \le k$ (Steps 2-8) and maintain the r that maximizes $c(r)$. For each r we create a projector matrix Π_r for the subspace defined by the dominant r singular values of A. We use Π_r to compress the tensor $\mathcal{X}$ by taking its *3-mode* product $\times_3$ with Π_r (Step 4). The n-mode product multiplies a tensor and a matrix that match on the n-th mode of the tensor, in the same fashion as matrix-matrix multiplication. We similarly compress A to obtain A_r (Step 5) and compute the core consistency of $\mathcal{X}_r$ using $[C, C', A_r]$, employing *efficient_corcondia* [41] (Step 7). The highest core consistency value $c(r)$ is supplied to AutoTen [39] which estimates the rank k^*, which reveals the number of communities in the data (Step 9). While the optimal k detection is tensor-oriented, one can easily adopt it for GM and other error-of-fit models, relying on a fixed natural number of communities in the data.

4.4 Learning λ_c and λ_d using MDL

The regularization parameters λ_c, λ_d control the relative importance of the fused lasso regularization in our objective functions. Thus, it is important to set them appropriately to balance the contributions of J and $R(A)$. We propose to set λ_c and λ_d based on the Minimum Description Length (MDL), where we aim to minimize the number of bits needed to encode errors due to the fit and the number of "switches" between active and inactive community states in A. Intuitively, the higher the deviation of each element of $\mathcal{X}(i, j, t)$ from its reconstruction, the more bits are needed to encode this error in a lossless compression employing the characterization.

For the case of LARC-TF, we formalize the average bits to encode the the error of reconstruction as:

$$B_{TF}^{\{\lambda_c,\lambda_d\}} = -\log\left(||\mathcal{X} - \mathcal{X}_{\{\lambda_c,\lambda_d\}}||_F^2\right)/|\mathcal{X}|, \qquad (13)$$

where $\mathcal{X}_{\{\lambda_c,\lambda_d\}}$ is the reconstruction obtained by employing LARC-TF with parameters set to λ_c and λ_d, and $|\mathcal{X}|$ is the number of elements of the tensor. Similarly we define the average bits to encode error due to LARC-GM as:

$$B_{GM}^{\{\lambda_c,\lambda_d\}} = -\log\Big(\sum_{i,j,t}[\mathcal{X}(i,j,t) - P_{\{\lambda_c,\lambda_d\}}(i,j,t)]^2\Big)/|\mathcal{X}|, \qquad (14)$$

where $P_{\{\lambda_c,\lambda_d\}}(i, j, t)$ is the probability of observing an edge according to the model learned by LARC-GM using λ_c and λ_d.

The second part of our MDL encoding is the number of "switches" between active and inactive community states which we quantify as $\Delta^{\{\lambda_c,\lambda_d\}} = -\log(||DA_{\{\lambda_c,\lambda_d\}}||_F^2)/|A_{\{\lambda_c,\lambda_d\}}|$, where $A_{\{\lambda_c,\lambda_d\}}$ is the community activation matrix learned using the corresponding regularization parameters by either of the models. To find the parameters, we then minimize the total number of bits:

$$\{\lambda_c, \lambda_d\} = \arg\min B^{\{\lambda_c,\lambda_d\}} + \Delta^{\{\lambda_c,\lambda_d\}}, \qquad (15)$$

where $B^{\{\lambda_c,\lambda_d\}} \in \{B_{GM}^{\{\lambda_c,\lambda_d\}}, B_{TF}^{\{\lambda_c,\lambda_d\}}\}$. Minimizing the objective depends on invoking LARC-TF or LARC-GM, thus, we perform a grid search over possible values and pick the configuration that minimizes MDL. We demonstrate that in a variety of synthetic and real datasets setting the parameters according to the MDL principle results in optimal quality of detecting ground truth communities.

A similar heuristic can be utilized to learn a variety of parameters, including our rank. In that case, however, we would need a more complex encoding to account for the additional information held by an increased rank; we therefore prefer the more established rank method above.

5 EXPERIMENTAL EVALUATION

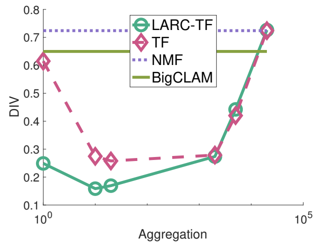
We evaluate the quality, scalability, and real-world utility of LARC on both real and synthetic datasets. All experiments are for single core execution of our methods implemented in Matlab. For tensor manipulations we use the Tensor Toolbox for Matlab [5, 6].

5.1 Data

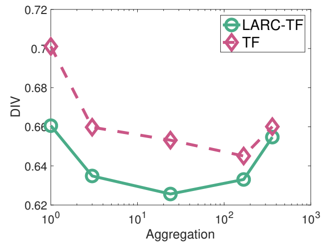
We summarize the datasets used for evaluation in Table 1.

Synthetic: We generate the community structure and a smooth temporal activation. $|V|$ nodes are randomly assigned to k overlapping communities C while fixing the average community overlap (Jaccard similarity). Piece-wise constant activation profiles $A_{:k}$ are sampled from a Markov chain with active 1 and inactive 0 states for length T and state-change probability of $p = 0.2$. We next generate temporal interactions for each snapshot $X(t)$ using A, C and based on the GM specified in Sec 3.2. We also "stretch" the temporal dimension of instances in a controlled fashion to simulate temporal oversampling and varying rates of interactions within communities across time. We map each original time step t to $s \sim Poisson(\lambda_{st})$ time steps, where the the interactions in $X(t)$ are uniformly distributed across the s new snapshots. This makes recovering the original profile and communities increasingly challenging.

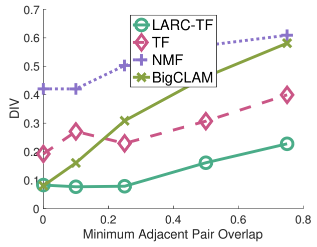
Real-world: We employ several real-world datasets from different domains: *Football* [44]; *Reality Min.* [20] captures the temporal interactions (calls, texts, BT proximity) among students and faculty with self-reported friendship relations which we use to get ground truth communities employing BigCLAM [50]; *Reddit* datasets contain exchanges between users (posts and replies) on reddit.com, where



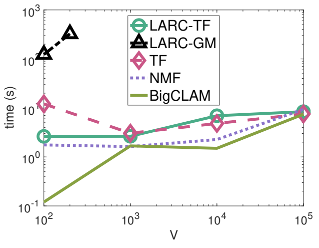
(a) Quality w. aggregation (Synthetic)



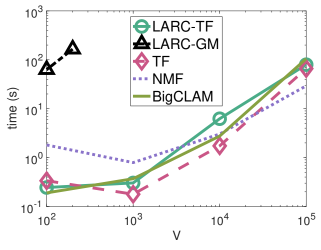
(b) Quality w. aggregation (Reality Min.)



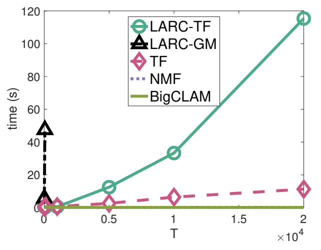
(c) Quality w. Overlap. (Synth.)



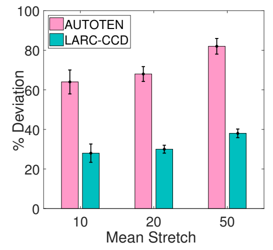
(d) Scalability w. V, fixed nnz



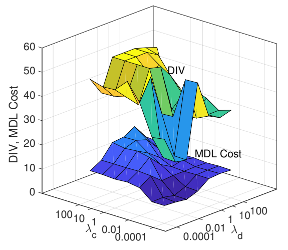
(e) Scalability w. V, nnz=1%



(f) Scalability w. T, V=100



(g) Learning the rank k



(h) Learning λ_c and λ_d

Figure 2: Comparison of LARC-TF's quality with that of competing techniques for varying levels of temporal aggregation on synthetic (V=200,T=20000)(a), and Reality Min. (b) datasets. Quality comparison for increasing community overlap (measured in Jacccard Similarity) on synthetic data (V=200,T=2500) (c). Comparison of scalability with increasing V and fixed nnz=const (T=50) (d), increasing V and nnz=1% (T=50) (e), and increasing T (V=100) (f). Quality of estimating the GT number of communities k for increasing average stretch on synthetic (T=1k,V=100) (g). DIV and MDLfrom Eq. 15 as a function of the regularization parameters λ_c and λ_d (h).

the ground truth communities are based on subset of sports and news *subreddits* and their participants [2]; *Bike Rides* consists of rides between bike rental stations in Boston, MA over 2 years [1]; and *Botnet* contains inter-IP flows of university normal and botnet machines and includes several time-annotated DDoS attacks involving traffic of 10 bot and 1 victim IP [23].

5.2 Experimental Setup

Baselines: We evaluate our methods' quality in retrieving ground truth (GT) overlapping communities and running time in comparison to three baselines. *TF* is a tensor factorization method proposed for overlapping temporal community detection by Gauvin et al. [25] and can also be viewed as a special case of our LARC-TF, where regularization in time is turned off. *BigCLAM* is the state-of-the-art method for overlapping community detection based on the affiliation generative model, which we extend to time to obtain LARC-GM [50]. *Non-negative matrix factorization (NMF)* is another popular approach for overlapping communities [33]. Since they operate on static graphs, we employ both BigCLAM and NMF on an aggregated temporal interactions static graph.

Metrics: For datasets with GT, we compare the level of agreement between GT and learned communities by all competing techniques. The *Kullback—Leibler divergence (KL-Div)* has been previously used to evaluate overlapping cluster solutions [26], where both GT and learned communities are treated as distributions over the nodes and the measure quantifies the differences between them. KL-Div is, however, not symmetric and also not defined when the distributions have regions of 0 density. Hence, we apply a metric alternative *Jansen-Shannon divergence (DIV)* [48] that handles 0 probabilities and varies between 0 (no divergence) and 1. We also adopt the normalized mutual information (NMI) [31] to compare learned and GT communities, where higher NMI corresponds to better detection of GT. This measure requires 0/1 membership, hence, we threshold community vectors using values ranging in $[10^{-3}, 1]$ in order to obtain the best NMI score for each method.

5.3 Quality, scalability, and parameter selection

Quality. The main advantage of LARC is in its treatment of time: enforcing solutions in which communities alternate between active/inactive in contiguous periods. To test this experimentally, we simulate temporal oversampling by stretching a smooth instance with expected on/off behavior by "stretching" it in time and spreading a given time slice's temporal interactions to several new time slices controlled by an average stretch parameter. Fig. 1(b) (Sec. 1) shows a quality comparison in terms of Divergence (DIV) from GT communities of all competing techniques for increasing average stretch of a Synthetic dataset. For small stretch, i.e. a well-behaved smooth instance, LARC-TF and LARC-GM outperform TF by a small margin and static methods (NMF and BigCLAM) by a factor of 3 in DIV. For increasing stretch, LARC-TF maintains a near constant quality, while TF deteriorates to the quality level of BigCLAM (2.6x deterioration). This behaviour is due to regularization forcing

LARC-TF to continue considering a temporal segmentation of time at which communities are most discernible, while TF is affected by only observing partially the interactions of a community in individual timeslices and not employing any smoothness in time. At the same time, static methods suffer from an opposite extreme - over-aggregation in which the communities become also hard to discern. While LARC-GM performs very well in terms of quality, it does not scale to long timelines due to its reliance on coordinate descent methods for both A and C.

Since communities are elusive to the baseline methods, at both high and low (full aggregation) temporal resolution, we next investigate the feasibility of varying regular aggregations to get better communities in a synthetic Fig. 2(a) and the Reality Mining Fig. 2(b) datasets. The trends in both figures reveal the aforementioned challenges at both ends of aggregation and a slightly better performance for TF for medium aggregation. LARC, however, maintains a consistently better quality than the baselines at high resolutions (on synthetic) and as the aggregation coarsens deteriorates to the performance of TF as at these levels the useful temporal information is lost. Note, that in Reality mining (Fig. 2(b)), at the highest resolution even LARC suffers from fragmented communities in time and regularization cannot really attain the best quality at some reasonable small aggregation level. Note also that the GT in Reality mining relies on BigCLAM-extracted overlapping communities from the user-reported friendship graph and thus may not be ideal, resulting in relatively high DIV values. Comparison to LARC-GM is again omitted due to limited scalability for high temporal resolutions and similar to LARC-TF's performance for low-resolutions.

Naturally, the quality of community detection deteriorates with the amount of overlap among communities as evident in Fig. 2(c). However, thanks to the timing of interactions coupled with smoothness regularization LARC-TF consistently outperforms all baselines. Interestingly, the temporal information loses its utility when there is very small overlap and BigCLAM performs on par with LARC-TF although using fully aggregated as opposed to temporal data. As demonstrated in previous studies, NMF performs consistently worse than BigCLAM and thus all other competing methods.

Scalability. While exhibiting good quality on small instances LARC-GM does not scale well with both V: Figs. 2(d), 2(e) and T: Fig. 2(f) due to its reliance on coordinate methods for both A and C. LARC-TF, on the other hand, scales similar to TF and both of them slightly slower than the static baselines for increasing V while keeping the number of non-zeroes (nnz) in the input constant Fig. 2(d). NMF and BigCLAM's time increases on par with the TF methods when the nnz is kept at 1% of the tensor size for increasing V Fig. 2(e), since the resulting aggregate graphs densify with the nnz. As expected, since LARC-TF performs smoothing via a descent method on the activation profiles A, its running time increases faster than that of TF—computational time invested to enable its superior quality performance. It, nevertheless, completes within a few minutes on our largest real-world and synthetic datasets Tbl 1 and Fig. 2(f). It is worth to note that since it is an AOADDM method, it is amenable to parallel and distributed implementations, which can enable its feasible adoption for analysis of even larger timelines.

Parameter selection. We evaluate the ability of our estimation approach LARC-CCD to recover the GT number of communities k_{GT} and compare it to a regularization-oblivious approach from the literature AutoTen [39]. Since both approaches require a maximum k_{max} to probe, we set this value to $2k_{GT}$ for both methods in synthetic data and $k_{max} = 20$ for all real-world datasets. Fig. 2(g) shows the comparison of the two competing methods, where the quality measure on the vertical axis is the percent deviation of the estimated k_{EST} from the GT one: $\frac{|k_{EST}-k_{GT}|}{k_{GT}}$. We report the average of 10 increasing average stretch of a synthetic dataset. LARC-CCD outperforms AutoTen consistently by at least a factor of 2 in terms deviation. The reason for this performance is the regularization-enabled compression we perform on the input tensor detailed in Alg. 4.3. This superior performance in estimating the number of communities is also evident in real-world datasets as reported in columns 6 and 7 of Tbl. 1.

We also evaluate the utility of our MDL approach for automatically selecting the regularization parameters λ_c and λ_d Fig. 2(h). In this experiment, we vary the two parameters in exponential steps and compare the shape of the DIV surface and that for MDL cost, employing the TF bit cost $B_{TF}^{\{\lambda_c,\lambda_d\}}$. Since the global minima of both functions are attained for the same region of values of the parameters, MDL can successfully be employed as a proxy for parameter estimation in conjunction with calls to LARC. We observe similar behavior on other datasets as well. It is worth noting that while DIV has multiple local minima, MDL is much more smooth, hence line-search approaches can be adopted to speed-up the estimation without covering the full grid of parameters.

Overall evaluation and discussion. A comparison of the quality and running time for all datasets is presented in Tbl. 1. Here we show two measures of quality: DIV and NMI, and report all running times in seconds. In terms of LS divergence (DIV), LARC-TF outperforms baselines on all datasets with ground truth except for the Football dataset on which BigCLAM and NMC perform slightly better. The reason for this behavior is the very low overlap between communities in this dataset rendering temporal interactions disadvantageous compared to a full aggregation. Similar behavior is observed in our quality with increasing overlap experiment Fig. 2(c) where for the minimum overlap BigCLAM similarly performs on par with LARC-TF. The pattern is similar for NMI with the exception of the Reality mining dataset, where BigCLAM has a slightly higher, though very close to 0 NMI. As we discussed earlier, the ground truth for this dataset is based on static overlapping friendship communities detected by BigCLAM, which may not align well with the multi-mode user interactions observed in the data, thus resulting in relatively low quality on all datasets.

In terms of running time, while LARC-TF is slower than alternatives, its running time exceeds 1 min only on the long ($T = 20k$) synthetic instance, making it practical to employ on large real-world datasets. An important observation here is that while the aggregate methods are typically much faster than TF and LARC-TF, their running time increases even beyond that of the temporal methods on the Reddit datasets due to the aggregate matrix they operate on being significantly denser than the individual time snapshots on which TF and LARC-TF operate.

LARC-GM's running time increases faster with the input size (Figs. 2(d),2(e),2(f)) due to the more expensive gradient solver. However, its quality on small instances is promising (see first two points in Fig. 1(b)) due to the same temporal regularization we employ in

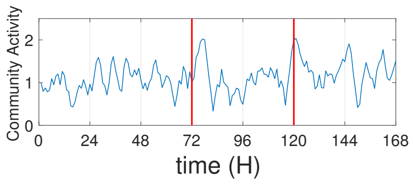


Figure 3: An activation profile of a Reality Mining community

LARC-TF. Better solvers (e.g. stochastic alternatives) for J_{GM} may render it a better fit for some datasets in practice. In addition, alternative *error-of-fit* functions for communities (i.e., beyond J_{TF} and J_{GM}) may also be advantageous within our general regularization framework. We plan to investigate the above questions in future research.

5.4 LARC at work

Beyond our analysis of the temporal changes in the Bike Rides dataset in Fig. 1(a), we also consider a sample community activity profile from the Reality mining dataset in Fig. 3. Weekday midnights are marked by grey lines, while the weekend is enclosed between red vertical lines. Outside of a Saturday morning spike, variance in activity seems lower overall during the weekend, though interpretation of this pattern may require more thorough community information. The diversity of potential options for ground truth and interaction types in the Reality Mining data (friendships, work, and external contacts; proximity and calls) makes assigning definitive communities and behaviors difficult, which may explain relatively higher DIV values. While these patterns are more or less expected, the activity profiles may be employed to detect abnormal activity movements, thus enabling anomaly detection at the community level. We detected such a big change in the bike service data in Fig. 1(a) which coincided with the geographical expansion of the service. The activity profiles can also be used to inform appropriate temporal aggregations of network data which can then be employed for other tasks: e.g. temporal link prediction, partitioning and others.

While we do not have an exact GT community structure for the Botnet data, the meta-data specifies the set of bot IPs and that of the victim which is flooded by packets several times during the trace in coordinated DDoS attacks. We employ LARC-TF on this dataset, setting $k = 3$, (based on the recommendation from our k estimation approach LARC-CCD) and examine the resulting communities. The entire botnet and the victim are consistently included in one of the reported communities. While in terms of total number of network flows, the botnet traffic does not stand out in this trace, the coordinated timings of the attack allow LARC-TF to group participants in the attack, demonstrating its potential as a network traffic analysis tool for security professionals.

6 CONCLUSION

We proposed LARC, a novel dynamic overlapping community detection framework to learn jointly the community structure and the temporal community activity profile. It enforces interpretable piece-wise constant activity profiles via a temporal smoothness regularization. We demonstrated that our framework can successfully accommodate different measures of community fit. Our proposed algorithms, by virtue of effectively leveraging the temporal aspect of the data, demonstrate 2.6× quality improvement over state-of-the-art baselines on data with ground truth communities. We demonstrate the importance of regularizing time when dealing with dynamic networks, and suggest that similar or alternative regularizations can be implemented on top of other community detection methods. Furthermore, LARC produced interpretable and intuitive results when applied “in the wild”, to a variety of real-world scenarios (botnet attacks, change points in urban transportation patterns, and public forum interaction trends), demonstrating its wide applicability and practicality as a data mining tool. In addition to our optimization techniques, we provided a comprehensive set of tools for choosing estimating all method parameters in an *unsupervised* manner, rendering LARC useful for researchers and practitioners alike.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their helpful comments. Work at University at Albany was supported by DARPA under Contract No FA8650-18-C-7824. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of AFRL, DARPA, or the U.S. Government. Work at UCR was partially supported by the Department of the Navy, Naval Engineering Education Consortium under Award no. N00174-17- 1-0005, by the National Science Foundation Grant no. EAGER 1746031, and an Adobe Data Science Research Faculty Award.

REFERENCES

[1] [n. d.]. Hubway Data Visualization Challenge: http://hubwaydatachallenge.org.
[2] [n. d.]. Reddit Comments Crawl https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment/.
[3] Rezwan Ahmed and George Karypis. 2011. Algorithms for Mining the Evolution of Conserved Relational States in Dynamic Networks. In *ICDM*.
[4] Miguel Araujo, Spiros Papadimitriou, Stephan Günnemann, Christos Faloutsos, Prithwish Basu, Ananthram Swami, Evangelos E Papalexakis, and Danai Koutra. 2014. Com2: fast automatic discovery of temporal (‘comet’) communities. In *PAKDD*. Springer, 271–283.
[5] Brett W. Bader and Tamara G. Kolda. 2007. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing* 30, 1 (Dec. 2007), 205–231. https://doi.org/10.1137/060676489
[6] Brett W. Bader, Tamara G. Kolda, et al. 2015. MATLAB Tensor Toolbox Version 2.6. Available online. http://www.sandia.gov/~tgkolda/TensorToolbox/
[7] Tanya Berger-Wolf, Chayant Tantipathananandh, and David Kempe. 2010. Dynamic community identification. In *Link Mining: Models, Algorithms, and Applications*. Springer New York, 307–336. https://doi.org/10.1007/978-1-4419-6515-8_12
[8] Petko Bogdanov, Misael Mongiovi, and Ambuj K. Singh. 2011. Mining Heavy Subgraphs in Time-Evolving Networks. In *ICDM*.
[9] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3, 1 (2011), 1–122.
[10] Ronald L. Breiger. 1974. The Duality of Persons and Groups. *Social Forces* 53, 2 (dec 1974), 181. https://doi.org/10.2307/2576011
[11] Rasmus Bro. 1997. PARAFAC. Tutorial and applications. *Chemometrics and intelligent laboratory systems* 38, 2 (1997), 149–171.
[12] Rasmus Bro and Henk AL Kiers. 2003. A new efficient method for determining the number of components in PARAFAC models. *Journal of chemometrics* 17, 5 (2003), 274–286.
[13] Marcin Budka, Katarzyna Musial, and Krzysztof Juszczyszyn. 2012. Predicting the evolution of social networks: optimal time window size for increased accuracy. In *PASSAT, 2012 Int. Conf. on and 2012 Int. Conf. on Social Computing*. IEEE, 21–30.
[14] Rajmonda Sulo Caceres, Tanya Berger-Wolf, and Robert Grossman. 2011. Temporal scale of processes in dynamic networks. In *Proc. of ICDMW*. IEEE, 925–932.

[15] Jose Cadena, Anil Kumar Vullikanti, and Charu C Aggarwal. 2016. On dense subgraphs in signed network streams. In *ICDM, 2016 IEEE 16th Int. Conf. on.* IEEE, 51–60.

[16] Yudong Chen, Vikas Kawadia, and Rahul Urgaonkar. 2013. Detecting overlapping temporal community structure in time-evolving networks. *preprint arXiv:1303.7226* (2013).

[17] Daniel J DiTursi, Gaurav Ghosh, and Petko Bogdanov. 2017. Local Community Detection in Dynamic Networks. In *Proceedings of the IEEE International Conference on Data Mining (ICDM), New Orleans, USA.*

[18] Daniel J DiTursi, Gregorios A Katsios, and Petko Bogdanov. 2017. Network Clocks: Detecting the Temporal Scale of Information Diffusion. In *Proceedings of the IEEE International Conference on Data Mining (ICDM), New Orleans, USA.*

[19] Haifeng Du, Marcus W Feldman, Shuzhuo Li, and Xiaoyi Jin. 2007. An algorithm for detecting community structure of social networks based on prior knowledge and modularity. *Complexity* 12, 3 (2007), 53–60.

[20] Nathan Eagle and Alex Sandy Pentland. 2006. Reality mining: sensing complex social systems. *Personal and ubiquitous computing* 10, 4 (2006), 255–268.

[21] Santo Fortunato. 2010. Community detection in graphs. *Physics Reports* 486, 3âĂŞ5 (2010), 75 – 174. https://doi.org/10.1016/j.physrep.2009.11.002

[22] Santo Fortunato and Darko Hric. 2016. Community detection in networks: A user guide. *Physics Reports* 659 (2016), 1–44.

[23] Sebastian Garcia, Martin Grill, Jan Stiborek, and Alejandro Zunino. 2014. An empirical comparison of botnet detection methods. *computers & security* 45 (2014), 100–123.

[24] Noé Gaumont, Clémence Magnien, and Matthieu Latapy. 2016. Finding remarkably dense sequences of contacts in link streams. *Social Network Analysis and Mining* 6, 1 (2016), 87.

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

[26] Mark K Goldberg, Mykola Hayvanovych, and Malik Magdon-Ismail. 2010. Measuring similarity between sets of overlapping clusters. In *SocialCom, 2010 IEEE 2nd Int. Conf. on.* IEEE, 303–308.

[27] Ekta Gujral and Evangelos E Papalexakis. 2018. SMACD: Semi-supervised Multi-Aspect Community Detection. In *"In Proc. of SDM.* SIAM.

[28] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. 2015. Statistical Learning with Sparsity: The Lasso and Generalizations. *Crc* (2015), 362. https://doi.org/10.1201/b18401-1

[29] Kejun Huang, Nicholas D Sidiropoulos, and Athanasios P Liavas. 2016. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing* 64, 19 (2016), 5052–5065.

[30] Min-Soo Kim and Jiawei Han. 2009. A particle-and-density based evolutionary clustering method for dynamic networks. *VLDB Endow.* 2 (2009). Issue 1.

[31] Andrea Lancichinetti, Santo Fortunato, and János Kertész. 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* 11, 3 (2009), 033015.

[32] Silvio Lattanzi and D. Sivakumar. 2009. Affiliation networks. In *Proc. of STOC '09.* ACM Press, New York, New York, USA, 427. https://doi.org/10.1145/1536414.1536474

[33] Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788.

[34] Jure Leskovec, Kevin J Lang, and Michael Mahoney. 2010. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web.* ACM, 631–640.

[35] Yu-Ru Lin, Jimeng Sun, Paul Castro, Ravi Konuru, Hari Sundaram, and Aisling Kelliher. 2009. Metafac: community discovery via relational hypergraph factorization. In *Proc. of the 15th ACM SIGKDD ICDM.* ACM, 527–536.

[36] Siyuan Liu, Shuhui Wang, and Ramayya Krishnan. 2014. Persistent community detection in dynamic social networks. In *PAKDD.* Springer, 78–89.

[37] Misael Mongiovi, Petko Bogdanov, Razvan Ranca, Ambuj K. Singh, Evangelos Papalexakis, and Christos Faloutsos. 2013. NetSpot: Spotting Significant Anomalous Regions on Dynamic Networks. In *SDM.*

[38] M. Mongiovì, P. Bogdanov, and A.K. Singh. 2013. Mining evolving network processes. In *Proc. of IEEE ICDM.* 537–546. https://doi.org/10.1109/ICDM.2013.106

[39] Evangelos E Papalexakis. 2016. Automatic unsupervised tensor mining with quality assessment. In *Proc. of SDM.* SIAM, 711–719.

[40] Evangelos E Papalexakis, Leman Akoglu, and Dino Ienco. 2013. Do more views of a graph help? community detection and clustering in multi-graphs. In *FUSION, 2013 16th int. conf. on.* IEEE, 899–905.

[41] Evangelos E Papalexakis and Christos Faloutsos. 2015. Fast efficient and scalable core consistency diagnostic for the parafac decomposition for big sparse tensors. In *ICASSP, 2015 IEEE Int. Conf. on.* IEEE, 5441–5445.

[42] Giulio Rossetti, Luca Pappalardo, Dino Pedreschi, and Fosca Giannotti. 2016. Tiles: an online algorithm for community discovery in dynamic social networks. *Machine Learning* (2016), 1–29.

[43] Polina Rozenshtein, Nikolaj Tatti, and Aristides Gionis. 2014. Discovering dynamic communities in interaction networks. In *Proceedings of ECML/PKDD.*

[44] Fatemeh Sheikholeslami and Georgios B Giannakis. 2017. Identification of Overlapping Communities via Constrained Egonet Tensor Decomposition. *arXiv:1707.04607* (2017).

[45] Rajmonda Sulo, Tanya Berger-Wolf, and Robert Grossman. 2010. Meaningful selection of temporal resolution for dynamic networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs.* ACM, 127–136.

[46] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67, 1 (2005), 91–108.

[47] Yen-Chuen Wei and Chung-Kuan Cheng. 1989. Towards efficient hierarchical designs by ratio cut partitioning. In *Computer-Aided Design, 1989. ICCAD-89. Digest of Technical Papers., 1989 IEEE International Conference on.* IEEE, 298–301.

[48] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. Twitterrank: finding topic-sensitive influential twitterers. In *Proc. of the 3rd ACM WSDM.* ACM, 261–270.

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

[50] Jaewon Yang and Jure Leskovec. 2013. Overlapping community detection at scale. In *Proceedings of the sixth ACM international conference on Web search and data mining - WSDM '13.* ACM Press, New York, New York, USA, 587. https://doi.org/10.1145/2433396.2433471

[51] Jaewon Yang and Jure Leskovec. 2015. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems* 42, 1 (2015), 181–213.