Overlapping Community Detection via Constrained PARAFAC: A Divide and Conquer Approach

Fatemeh Sheikholeslami
Dept. of ECE and Digital Tech. Center
University of Minnesota
Minneapolis, MN 55455, USA
Email: sheik081@umn.edu

Georgios B. Giannakis
Dept. of ECE and Digital Tech. Center
University of Minnesota
Minneapolis, MN 55455, USA
Email: georgios@umn.edu

***Abstract*—The task of community detection over complex networks is of paramount importance in a multitude of applications. The present work puts forward a top-to-bottom community identification approach, termed DC-EgoTen, in which an egonet-tensor (EgoTen) based algorithm is developed in a divide-and-conquer (DC) fashion for breaking the network into smaller subgraphs, out of which the underlying communities progressively emerge. In particular, each step of DC-EgoTen forms a multi-dimensional egonet-based representation of the graph, whose induced structure enables casting the task of overlapping community identification as a constrained PARAFAC decomposition. Thanks to the higher representational capacity of tensors, the novel egonet-based representation improves the quality of detected communities by capturing multi-hop connectivity patterns of the network. In addition, the top-to-bottom approach ensures successive refinement of identified communities, so that the desired resolution is achieved. Synthetic as well as real-world tests corroborate the effectiveness of DC-EgoTen.**

***Index Terms*—Community detection, overlapping communities, egonet subgraphs, tensor decomposition, constrained PARAFAC.**

I. Introduction

Real-world networks often exhibit distinct characteristics, such as power-law degree distribution, the small-world phenomena, and the presence of densely connected sub-graphs, also referred to as "communities" or "clusters" [1]. Focusing on the last, strong connectivity of a subset of nodes along with their sparse interactions with the rest of the network is indicative of a "real-world association" among the participating nodes. The task of community detection targets the discovery of such *communities*, whose identification is of great importance in diverse fields ranging from gene-regulatory networks [2], to brain functionality [3], and social-media evolution analysis [4], [5], to name a few.

Past works on community detection include those based on generative and statistical models [6]–[8], modularity and related local-metric optimization [9]–[11], spectral clustering [12], and matrix factorization approaches [3], [13]–[17]; see also [1] and [18] for comprehensive overviews. However, most existing works pursue a bottom-up approach, where small collections of nodes with strong connectivity patterns (e.g., cliques) are selected as "seeds," and larger communities are "grown" around them by merging other (clusters of) nodes [9], [19]. In contrast, another class of algorithms follows a top-to-bottom perspective, where a graph is progressively broken into smaller pieces, out of which communities eventually emerge [20]–[22].

Recent exploratory studies have revealed new challenges over contemporary networks, addressing the presence of overlapping communities [23]–[25], multimodal interaction of nodes over multiview networks [26], [27], exploitation of nodal and edge-related side-information [28], as well as dynamic interactions within a network [29], [30]. In tackling these challenges, *tensors* as multi-modal structures offer increased representational capacity, which translates to improved performance [26], [27], [29], [31]–[34].

In this work, we develop a novel top-to-bottom community detection approach, termed "divide-and-concur EgoTen" (DC-EgoTen), which relies on a successive application of "EgoTen", a tensor-based toolbox for intermediate steps of community identification. Our core algorithm EgoTen builds on a novel multi-dimensional representation of a network, whose ability in capturing multi-hop connectivities is particularly appealing when communities are overlapping as well as highly-mixing. The proposed tensor-based approach views a network as a union of its egonets, where each egonet is the subgraph induced by a node, its immediate neighbors, and their connections [35]. The resultant three-way tensor is thus built by concatenation of egonet adjacency matrices as frontal slabs. The tensor's constrained decomposition lends itself to an algorithm revealing communities though the trilinear decomposition factors. A desirable characteristic of this algorithm is its ability to trade off flexibility for increased redundancy and memory costs. Nevertheless, the resulting tensor is extremely sparse, and off-the-shelf tools for sparse tensor computations can be readily utilized; see e.g., [36]–[38].

The upshot of our novel framework is three-fold: i) the performance of community detection in complex networks improves markedly thanks to the rich structure of tensors; ii) construction of the egonet-tensor via parallel implementation and exploitation of sparsity endow the algorithm with scalability; and, iii) the proposed top-to-bottom approach offers communities with the desired *resolution*. In fact, many of the previously developed algorithms are susceptible to "resolution

Work in this paper was supported by NSF grants 1500713, 1442686, 1514056, and NIH grant no. 1R01GM104975-01.

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.22

limit" [39], where identification of very large communities reveals little information on the underlying graph structure.

The rest of the paper is organized as follows. Section II introduces DC-EgoTen, and Section III presents EgoTen as the core tensor-based community detection approach along with its solver. Extraction of communities and performance metrics are the subjects of Section IV, while Section V presents numerical tests, and Section VI concludes the paper.

Notation. Lower- (upper-) case boldface letters denote column vectors (matrices), and underlined upper-case boldface letters stand for tensor structures. Calligraphic symbols are reserved for sets, while ^{T} stands for transposition. Symbols $\circ$ and $\otimes$ are reserved for outer- and Kronecker-product, respectively, while $\mathrm{Tr}\{\mathbf{X}\}$ denotes the trace of matrix $\mathbf{X}$.

II. Preliminaries and the top-down approach

Given a network of N vertices (or nodes) $v \in \mathcal{V}$ where $|\mathcal{V}| = N$, and their edgeset $\mathcal{E}$, community detection aims at finding subsets of nodes, a.k.a. clusters or communities, for which resident nodes demonstrate dense intra-community connections while distinct communities are sparsely connected. A cover is defined as the set of such communities, with "desirable covers" exhibiting certain characteristics, namely: i) constituent communities should include dense intra-connections and sparse inter-connections; ii) communities of very large sizes are not appealing as they bear little information on the underlying structure of the network; and, iii) the union of the identified communities should cover the entire graph, leaving few or no "homeless" nodes, not assigned to any community.

The proposed method, called "DC-EgoTen," relies on the construction of an egonet-based multi-dimensional representation of the network. It utilizes "EgoTen" to solve a sequence of nonnegative tensor decomposition subproblems, and progressively unveils the identified communities over the graph. Let us treat EgoTen as a black-box module in this section, postponing its detailed explanation to Section III, and further delineate the overall algorithm here.

In particular, DC-EgoTen takes a top-down approach for the overall task of community identification. To this end, "EgoTen" is initially applied over the entire network to provide an assignment of nodes to a few "coarse" communities. Each of the detected communities is in fact a subset of nodes, inducing a subgraph in the overall graph. Thus, the identified "coarse" communities are further amenable to a subsequent application of EgoTen for unraveling a more refined community structure. This procedure can be applied consecutively for a number of times over each of the detected communities, creating a tree of communities, until the desired resolution, i.e., maximum acceptable community size, is achieved for all detected communities (at the leaves of the tree). In Section III, the proposed egonet-based multi-dimensional graph representation is introduced, and "EgoTen" as our core toolbox for community detection is detailed.

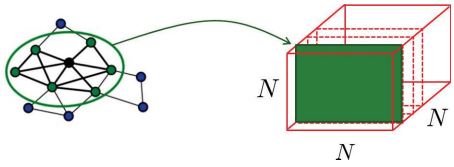


Fig. 1: Construction of the three-way egonet-tensor.

Algorithm 1 Egonet-tensor construction

procedure EGONET-TENSOR CONSTRUCTION($\mathcal{V}, \mathbf{W}$)
  for $n \in \mathcal{V}$ **do**
    $\mathcal{N}(n) := \{v \in \mathcal{V} | w_{nv} \neq 0\}$
    $\mathbf{W}^n \leftarrow \text{subgraph}\Big(\{n\} \cup \mathcal{N}(n), \mathbf{W}\Big)$
    $\underline{\mathbf{W}}_{:,:,n} = \mathbf{W}^{(n)}$
  end for
end procedure
return $\underline{\mathbf{W}}$

III. Egonet-Tensor Construction and Constrained Decomposition

Given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the binary adjacency matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is constructed by setting the (i,j)-th entry as $w_{ij} = 1$ if $(i,j) \in \mathcal{E}$, and $w_{ij} = 0$, otherwise. Furthermore, the egonet of node n is defined as the subgraph induced by node n, its one-hop neighbors denoted by $\mathcal{N}(n)$, and all their connections [35]. Thus, the egonet of node n can be conveniently represented by the induced subgraph $\mathcal{G}^{(n)} := (\mathcal{V}, \mathcal{E}^{(n)})$, where $\mathcal{E}^{(n)}$ is the edge set of the links in between nodes $\{n\} \cup \mathcal{N}(n)$. Subsequently, the egonet adjacency matrix $\mathbf{W}^{(n)} \in \mathbb{R}^{N \times N}$ is defined as

$$w_{ij}^{(n)} := \begin{cases} w_{ij} & \text{if } (i,j) \in \mathcal{E}^{(n)} \\ 0 & \text{otherwise.} \end{cases}$$

Typically, the center node n is excluded from $\mathcal{G}^{(n)}$, but it is included here for convenience.

Let us now consider a *three-way egonet-tensor* $\underline{\mathbf{W}} \in \mathbb{R}^{N \times N \times N}$ constructed by contanetating egonet adjacency matrices $\mathbf{W}^{(n)}$ for all nodes $n \in \mathcal{V}$ in the frontal slabs of $\underline{\mathbf{W}}$. In tensor parlance, that is tantamount to setting the n-th frontal slab of $\underline{\mathbf{W}}$ as $\underline{\mathbf{W}}_{:,:,n} := \mathbf{W}^{(n)}$, where : is a free index that spans its range.

The advantage of representing a graph via its egonet-tensor is due to the fact that tensors as multi-way data structures are capable of capturing higher-order connectivities, namely two-hop links among neighboring nodes. Thus, in networks where overlapping as well as highly-mixed communities render the task of community detection very challenging, egonet-tensors provide a rich representation of the graph, which will be leveraged in the upcoming algorithm. The egonet-based representation is also of interest particularly in the absence of extra nodal features, as the enhanced representation is a result of careful exploitation of the adjacency matrix where no other source of information is provided.

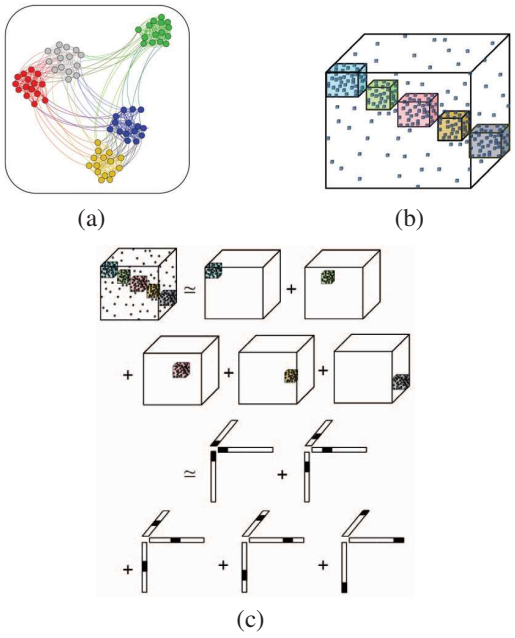


Fig. 2: (a) A toy network with 5 non-overlapping communities; (b) corresponding egonet-tensor; and (c) its community-revealing factorization via PARAFAC decomposition.

Fig. 1 illustrates the egonet-tensor construction procedure, while Algorithm 1 provides its pseudocode. In the ensuing subsection we cast the task of community detection as a constrained tensor decomposition over the egonet-tensor $\underline{\mathbf{W}}$, elaborate on the intuition behind the proposed approach, and introduce *EgoTen* as its efficient solver.

A. EgoTen: A Constrained Tensor Decomposition Approach

In order to gain insights into the properties of the introduced egonet-tensor, consider the toy network whose connectivity is depicted in Figure 2a. The network under consideration comprises five communities with dense intra-community and fewer inter-community connections. Upon constructing the egonet-tensor and after permutation (so that resident nodes are indexed right after one another), it becomes evident that the egonet-tensor demonstrates a block structure; see Fig. 2b. In particular, dense diagonal blocks in the tensor capture the dense intra-community links, while spare off-diagonal entries represent inter-community connections.

Had the communities been complete sub-graphs, each block would have been an all-one three-way tensor (considering non-zero diagonal entries provided by self-loops), which could have been readily decomposed into the outer product of three all-one vectors (each of the size of the community); that is, $\mathbf{1}_{p\times p\times p} = \mathbf{1}_{p\times 1} \circ \mathbf{1}_{p\times 1} \circ \mathbf{1}_{p\times 1}$, where p is the size of the community. Moreover, had the communities been disjoint, that is if no inter-community links were present, the egonet-tensor could have been readily written as the summation of five tensors, each of whom can be effectively approximated by the outer-product of three vectors; see Fig. 2c.

Such decomposition is indeed reminiscent of the well-known canonical polyadic decomposition (CPD) [36] also known as PARAFAC, where the number of terms, i.e., the rank of the decomposition, reveals the number of communities. Prompted by this observation, let us introduce the constrained nonnegative PARAFAC over the egonet adjacency tensor $\underline{\mathbf{W}}$ as

$$\begin{aligned}\{\widehat{\mathbf{A}},\widehat{\mathbf{B}},\widehat{\mathbf{C}}\} = \quad & \underset{\mathbf{A},\mathbf{B},\mathbf{C}}{\arg\min}\Big\{\|\underline{\mathbf{W}} - \textstyle\sum_{k=1}^{K}\mathbf{a}_k\circ\mathbf{b}_k\circ\mathbf{c}_k\|_F^2 \\ & \qquad +\lambda(\|\mathbf{A}\|_F^2+\|\mathbf{B}\|_F^2)\Big\} \qquad (1)\\ & \text{s.t.} \quad \mathbf{A}\geq\mathbf{0},\mathbf{B}\geq\mathbf{0},\mathbf{C}\geq\mathbf{0} \\ & \textstyle\sum_{k=1}^{K} c_{nk}=1 \qquad \forall n=1,2,...,N\end{aligned}$$

The first term in the objective is the original Frobenious term in the well-known PARAFAC, through which minimization of the mismatch between the multi-way data structure $\underline{\mathbf{W}}$ and its approximation is achieved. Furthermore, nonnegativity of the egonet-tensor is effected through additional constraints over the factors $\mathbf{A} := [\mathbf{a}_1,\ldots,\mathbf{a}_K]$, $\mathbf{B} := [\mathbf{b}_1,\ldots,\mathbf{b}_K]$ and $\mathbf{C} := [\mathbf{c}_1,\ldots,\mathbf{c}_K]$. Regarding the simplex constraints on the rows of matrix $\mathbf{C}$, let us now focus on the n-th frontal slab of the egonet-tensor. One can readily show that the tensor approximation gives rise to following decomposition

$$\mathbf{W}^{(n)} \simeq \sum_{k=1}^{K} c_{nk}(\mathbf{a}_k\circ\mathbf{b}_k) \qquad (2)$$

where c_{nk} denotes the (n,k)-th entry of factor $\mathbf{C}$. As stated earlier, parameter K is referred to as the rank of the decomposition, and in this application reveals the number of identified communities. Thus, such decomposition can be interpreted as a weighted sum over K "basis" $\{\mathbf{a}_k\circ\mathbf{b}_k\}_{k=1}^{K}$, where $(\mathbf{a}_k\circ\mathbf{b}_k)$ captures the "connectivity structure" within the k-th community. Consequently, c_{nk} can be viewed as *association level* of node n to community k. Thus, the simplex constraint over the rows of matrix $\mathbf{C}$ readily guarantees a normalized association vector for every node in the graph to the identified K communities. Finally, the Frobenious regularizers over factors $\mathbf{A}$ and $\mathbf{B}$ simply resolve the scaling ambiguity between the two factors, and is different from [40].

The overall optimization in (1) is a trilinear block-convex problem [41], whose solver is detailed in the following subsection.

B. Constrained PARAFAC Solver

Exploiting the block-convex structure of the constrained PARAFAC in (1), the optimization can be solved by alternating minimization, where each of $\mathbf{A},\mathbf{B},\mathbf{C}$ is optimized respectively by fixing the other two at their current values. Factors are repeatedly updated until a stopping criterion or a maximum number of iterations is achieved. Considering iteration i, factors are updated as follows.

1) Factor $\mathbf{A}$ *update:* Fixing factors $\mathbf{B}^{(i-1)}$ and $\mathbf{C}^{(i-1)}$ at their current values, the update of factor $\mathbf{A}$ is obtained by the corresponding subproblem, which after algebraic manipulation can be readily rewritten as a regularized nonnegative least-squares (LS) minimization as

$$\mathbf{A}^{(i)} = \arg\min_{\mathbf{A}\geq\mathbf{0}} \|\mathbf{W}_1 - \mathbf{H}_A^{(i)}\mathbf{A}^\top\|_F^2 + \lambda\|\mathbf{A}\|_F^2 \qquad (3)$$

where $\mathbf{W}_1 := [\text{vec}(\underline{\mathbf{W}}_{1,:,:}), \ldots, \text{vec}(\underline{\mathbf{W}}_{N,:,:})] \in \mathbf{R}^{N^2\times N}$ is a matricized reshaping of the tensor $\underline{\mathbf{W}}$, and $\mathbf{H}_A^{(i)} := \left[\mathbf{b}_1^{(i-1)} \otimes \mathbf{c}_1^{(i-1)}, \ldots, \mathbf{b}_K^{(i-1)} \otimes \mathbf{c}_K^{(i-1)}\right]$, with $\mathbf{b}_c^{(i-1)}$ ($\mathbf{c}_c^{(i-1)}$) denoting column c of $\mathbf{B}^{(i-1)}$ (resp. $\mathbf{C}^{(i-1)}$), and $\otimes$ the Kronecker product operator; see also [36]. Solving the subproblem in (4) by the alternating direction method of multipliers (ADMM), the augmented Laugrangian of the cost is

$$\begin{aligned}\mathcal{L}_A^{(i)}(\mathbf{A}, \bar{\mathbf{A}}, Y) = \quad & \|\mathbf{W}_1 - \mathbf{H}_A^{(i)}\bar{\mathbf{A}}^\top\|_F^2 + \lambda\|\bar{\mathbf{A}}\|_F^2 \\ & + r_+(\mathbf{A}) + (\rho/2)\|\mathbf{Y} + \mathbf{A} - \bar{\mathbf{A}}\|_F^2\end{aligned} \qquad (4)$$

where $\bar{\mathbf{A}}, \mathbf{Y} \in \mathbb{R}^{N\times K}$ are the auxiliary and dual variables, respectively, and $r_+(\mathbf{A})$ is the regularizer corresponding to the nonnegativity constraint,

$$r_+(\mathbf{A}) := \begin{cases} 0 & \text{if} \quad \mathbf{A} \geq \mathbf{0} \\ +\infty & \text{o.w.} \end{cases}$$

Simulated tests suggest that selection of the regularization parameter $\rho = \|\mathbf{H}_A^{(i)}\|_F^2/K$ can provide near-optimal performance [41], and that is the choice adopted henceforth.

The ADMM solver then proceeds by iteratively updating blocks of variables $\mathbf{A}, \bar{\mathbf{A}}, \mathbf{Y}$ as

$$\begin{cases} \bar{\mathbf{A}}^{(r)} = & \arg\min_{\bar{\mathbf{A}}} \mathcal{L}_A^{(i)}(\mathbf{A}^{(r-1)}, \bar{\mathbf{A}}, \mathbf{Y}^{(r-1)}) \\ & = \left(\mathbf{H}_A^{(i)\top}\mathbf{H}_A^{(i)} + (\lambda+\rho/2)\mathbf{I}_{K\times K}\right)^{-1} \\ & \quad \times\left(\mathbf{W}_1^\top\mathbf{H}_A^{(i)} + \frac{\rho}{2}(\mathbf{Y}^{(r-1)} + \mathbf{A}^{(r-1)})\right) \\ \mathbf{A}^{(r)} = & \mathcal{P}_+(\mathbf{Y}^{(r-1)} - \bar{\mathbf{A}}^{(r)}) \\ \mathbf{Y}^{(r)} = & \mathbf{Y}^{(r-1)} - \rho(\mathbf{A}^{(r)} - \bar{\mathbf{A}}^{(r)}) \\ r \quad = & r+1 \end{cases} \qquad (5)$$

until $\|\mathbf{A}^{(r)} - \mathbf{A}^{(r-1)}\|/\|\mathbf{A}^{(r-1)}\| \leq \epsilon$, or the maximum number of iterations is exceeded. Upon its termination, factor $\mathbf{A}$ is updated as $\mathbf{A}^{(i)} \leftarrow \mathbf{A}^{(r)}$, and the algorithm proceeds with updating factor $\mathbf{B}$ as in the following.

2) Factor $\mathbf{B}$ *update:* Upon fixing $\mathbf{A} = \mathbf{A}^{(i)}$ and $\mathbf{C} = \mathbf{C}^{(i-1)}$, factor $\mathbf{B}$ is updated by solving the subproblem

$$\mathbf{B}^{(i)} = \arg\min_{\mathbf{B}\geq\mathbf{0}} \|\mathbf{W}_2 - \mathbf{H}_B^{(i)}\mathbf{B}^\top\|_F^2 + \lambda\|\mathbf{B}\|_F^2 \qquad (6)$$

where $\mathbf{W}_2 := [\text{vec}(\underline{\mathbf{W}}_{:,1,:}), \ldots, \text{vec}(\underline{\mathbf{W}}_{:,N,:})] \in \mathbb{R}^{N^2\times N}$, and $\mathbf{H}_B^{(i)} := \left[\mathbf{a}_1^{(i)} \otimes \mathbf{c}_1^{(i-1)}, \ldots, \mathbf{a}_K^{(i)} \otimes \mathbf{c}_K^{(i-1)}\right]$, yielding a similar optimization problem as in (4). Undertaking the same approach as for (5), the ADMM update for solving (6) yields

$$\begin{cases} \bar{\mathbf{B}}^{(r)} = & \left(\mathbf{H}_B^{(i)\top}\mathbf{H}_B^{(i)} + (\lambda+\rho/2)\mathbf{I}_{K\times K}\right)^{-1} \\ & \quad \times\left(\mathbf{W}_2^\top\mathbf{H}_B^{(i)} + \frac{\rho}{2}(\mathbf{Y}^{(r-1)} + \mathbf{B}^{(r-1)})\right) \\ \mathbf{B}^{(r)} = & \mathcal{P}_+\left(\mathbf{Y}^{(r-1)} - \bar{\mathbf{B}}^{(r)}\right) \\ \mathbf{Y}^{(r)} = & \mathbf{Y}^{(r-1)} - \rho(\mathbf{B}^{(r)} - \bar{\mathbf{B}}^{(r)}) \\ r \quad = & r+1 \,. \end{cases} \qquad (7)$$

Upon the termination of (7) due to either attaining the stopping criterion or reaching the maximum number of iterations, factor $\mathbf{B}$ is updated as $\mathbf{B}^{(i)} \leftarrow \mathbf{B}^{(r)}$.

3) Factor $\mathbf{C}$ *update:* Fixing factors $\mathbf{A} = \mathbf{A}^{(i)}$ and $\mathbf{B} = \mathbf{B}^{(i)}$, update of factor $\mathbf{C}$ is obtained by solving the subproblem

$$\begin{aligned}\mathbf{C}^{(i)} = \quad & \arg\min_{\mathbf{C}} \|\mathbf{W}_3 - \mathbf{H}_C^{(i)}\mathbf{C}^\top\|_F^2 \\ \text{s.t.} \quad & , \mathbf{C} \geq \mathbf{0} \quad \textstyle\sum_{k=1}^K c_{nk} = 1 \quad \forall n = 1, \ldots, N\end{aligned} \qquad (8)$$

where $\mathbf{W}_3 := [\text{vec}(\underline{\mathbf{W}}_{:,:,1}), \ldots, \text{vec}(\underline{\mathbf{W}}_{:,:,N})]$ is the matricized version of $\underline{\mathbf{W}}$ along the 3-rd mode, and $\mathbf{H}_C^{(i)} := \left[\mathbf{a}_1^{(i)} \otimes \mathbf{b}_1^{(i)}, \ldots, \mathbf{a}_K^{(i)} \otimes \mathbf{b}_K^{(i)}\right]$. Augmented Laugrangian of the cost can be readily formed as

$$\begin{aligned}\mathcal{L}_C^{(i)}(\mathbf{C}, \bar{\mathbf{C}}, Y) \quad & = \|\mathbf{W}_3 - \mathbf{H}_C^{(i)}\bar{\mathbf{C}}^\top\|_F^2 + r_{\text{simp}}(\mathbf{C}) \\ & \quad + (\rho/2)\|\mathbf{Y} + \mathbf{C} - \bar{\mathbf{C}}\|_F^2\end{aligned}$$

where $r_{\text{simp}}(\mathbf{C})$ is the regularizer corresponding to the simplex constraint on the rows of matrix $\mathbf{C}$ as

$$r_{\text{simp}}(\mathbf{C}) := \begin{cases} 0 & \text{if} \quad \mathbf{C} \geq \mathbf{0}, \textstyle\sum_{k=1}^K c_{nk} = 1 \; \forall n \\ +\infty & \text{o.w.} \end{cases}$$

The ADMM solver then proceeds by iteratively updating the blocks of variables $\mathbf{C}, \bar{\mathbf{C}}, \mathbf{Y}$ as

$$\begin{cases} \bar{\mathbf{C}}^{(r)} = & \arg\min_{\mathbf{C}} \mathcal{L}_C^{(i)}(\mathbf{C}^{(r-1)}, \bar{\mathbf{C}}, \mathbf{Y}^{(r-1)}) \\ & = (\mathbf{H}_C^{(i)\top}\mathbf{H}_C^{(i)} + \rho/2\,\mathbf{I}_{K\times K})^{-1} \\ & \quad \times\left(\mathbf{W}_3^\top\mathbf{H}_C^{(i)} + \frac{\rho}{2}(\mathbf{Y}^{(r-1)} + \mathbf{C}^{(r-1)})\right) \\ \mathbf{C}^{(r)} = & \mathcal{P}_{\text{simp}}(\mathbf{Y}^{(r-1)} - \bar{\mathbf{C}}^{(r)}) \\ \mathbf{Y}^{(r)} = & \mathbf{Y}^{(r-1)} - \rho(\mathbf{C}^{(r)} - \bar{\mathbf{C}}^{(r)}) \\ r \quad = & r+1 \,. \end{cases} \qquad (9)$$

Projection of the rows of matrix $(\mathbf{Y}^{(r-1)} - \bar{\mathbf{C}}^{(r)})$ onto the simplex set can be achieved via the algorithm in [42]. Upon termination, factor $\mathbf{C}$ is updated as $\mathbf{C}^{(i)} \leftarrow \mathbf{C}^{(r)}$.

Once the overall trilinear optimization in (1) is solved, factor $\mathbf{C}$ unravels soft community association of the nodes. Extraction of hard communities based on the learned PARAFAC model is discussed in the next section. Also, Algorithm 2 lists the pseudocode of the proposed EgoTen followed by hard community assignments.

Algorithm 2 EgoTen Community Detection Core Algorithm

procedure EGOTEN($\underline{\mathbf{W}}, K$)
  Initialize $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{N \times K}$ at random and set $i = 0$
  while $i < I_{\max}$ **do** or not-converged
    $\mathbf{A}^{(i)} \leftarrow$ Solve (4) using (5)
    $\mathbf{B}^{(i)} \leftarrow$ Solve (6) using (7)
    $\mathbf{C}^{(i)} \leftarrow$ Solve (8) using (9)
    $i \leftarrow i + 1$
  end while
  for $k = 1, 2, \cdots, K$ **do**
    $\hat{\mathcal{C}}_k = \{\}$
    for $n = 1, 2, \cdots, N$ **do** $\hat{\mathcal{C}}_k \leftarrow \hat{\mathcal{C}}_k \cup \{n\}$ **if** $c_{nk} \geq \tau_k$
    end for
  end for
end procedure
retrun $\{\hat{\mathcal{C}}_k\}_{k=1}^K$

IV. Community assignment and quality evaluation

As discussed in Section III, the introduced EgoTen community detection algorithm aims at solving a constrained decomposition of the egonet-tensor, thus providing factor $\mathbf{C}$ whose entries unravel soft community associations. In order to transform the "soft" to "hard" memberships, one can simply utilize a threshold approach, according to which if $c_{nk} > \tau_k$, node n is assigned to community k, and it is not assigned otherwise. The main challenge here is on selecting a proper threshold τ_k. To this end, let $\hat{\mathcal{C}}_k$ denote the set of nodes in community k (with hard memberships), and define its conductance as [1]

$$\phi(\hat{\mathcal{C}}_k) := \frac{\sum_{i \in \hat{\mathcal{C}}_k, j \notin \hat{\mathcal{C}}_k} \mathbf{W}_{ij}}{\min\{\text{vol}(\hat{\mathcal{C}}_k), \text{vol}(\mathcal{V} \setminus \hat{\mathcal{C}}_k)\}}$$

where

$$\text{vol}(\hat{\mathcal{C}}_k) := \sum_{i \in \hat{\mathcal{C}}_k, \forall j} \mathbf{W}_{ij}$$

and $(\mathcal{V} \setminus \hat{\mathcal{C}}_k)$ is the complement of $\hat{\mathcal{C}}_k$. According to $\phi(.)$, *high-quality* communities yield small conductance scores as they exhibit dense connections among the nodes within the community and sparse connections with the rest.

Considering conductance as a measure of community quality, we can now set threshold τ_k such that the quality of community k after hard member assignment is maximized. In order to lower complexity, we simply choose τ_k from the discretized range $[1/K, 2/K, \dots]$. Note that having an association level $c_{nk} = 1/K \ \forall k$ for a given node n is tantamount to having an equally favorable association with the K communities, and having threshold $\tau_k = 1/K$ will result in a community assignment if the association is higher than this uniform level. Also, setting $\tau_k = 1/K$ together with the simplex constraints on the rows of factor $\mathbf{C}$ guarantees that every node will be assigned to at least one community, and no node will be left unassigned. However, tuning τ_k to obtain low conductance communities improves quality.

Algorithm 3 DC-EgoTen

procedure DC-EGOTEN($\mathcal{V}, \mathbf{W}$)
  Set parameters $K, C_{\max}$
  Define *global* cover set $\mathcal{S} = \{\}$
  $\underline{\mathbf{W}} \leftarrow$ Egonet-tensor construction($\mathcal{V}, \mathbf{W}$)
  $\{\mathcal{C}_i\}_{i=1,2,\dots,K} \leftarrow$ EgoTen($\underline{\mathbf{W}}, K$)
  for $\mathcal{C} \in \{\mathcal{C}_i\}_{i=1,2,\dots,K}$ **do**
    # If community $\mathcal{C}$ is refined enough, add it to the cover set $\mathcal{S}$, otherwise refine it using EgoTen
    if $|\mathcal{C}| < C_{\max}$ **then**
      $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{C}$
    else
      # Extract the subgraph of nodes in $\mathcal{C}$
      $\mathbf{W}_{\text{sub}} \leftarrow$ subgraph($\mathcal{C}, \mathbf{W}$)
      DC-EgoTen ($\mathcal{C}, \mathbf{W}_{\text{sub}}$)
    end if
  end for
end procedure
return $\mathcal{S}$

A. DC-EgoTen

Having delineated different modules of DC-EgoTen, we are ready to present the overall algorithm. Given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, DC-EgoTen initially constructs the egonet-tensor $\underline{\mathbf{W}}$ using Alg. 1, applies EgoTen in Alg. 2 over $\underline{\mathbf{W}}$, and obtains detected communities $\{\hat{\mathcal{C}}_k\}_{k=1}^K$. Next, the resolution of $\hat{\mathcal{C}}_k$ for $k = 1, 2, ...$ will determine whether further refining is necessary for each of the identified communities. That is, if $|\hat{\mathcal{C}}_k| < C_{\max}$, the resolution of detected community $\hat{\mathcal{C}}_k$ is satisfactory, and no further processing is required. On the other hand, if $|\hat{\mathcal{C}}_k| > C_{\max}$, the subgraph induced by the set of nodes in $\hat{\mathcal{C}}_k$ will be extracted, over which the entire process will be repeated. Algorithm 3 lists the pseudocode for the overall DC-EgoTen.

Figure 3 provides a schematic over our toy network with five communities, each of size $|\mathcal{C}_k| = 15$ for $k = 1, 2, \dots, 5$. In this example, in every EgoTen the rank parameter is $K = 2$, which gives rise to a binary tree of detected communities. As in this example, in the first application of EgoTen, the green community is detected by the constrained PARAFAC, while the rest of the network is 'lumped' together in the second community. Thus, the green community needs no further processing as its size is below $C_{\max} = 20$, while application of EgoTen on the second term gives rise to two relatively more refined communities. Proceeding with another set of EgoTen application on the detected communities will reveal the remaining clusters, creating overall five leaves in the tree, corresponding to the detected fine-resolution communities.

If an oracle had provided the number of underlying communities, the algorithm would have identified all clusters in its first application of EgoTen by setting $K = 5$. However, successive application of EgoTen with smaller target rank K can compensate for the lack of such information, which is almost-always encountered in practice. Furthermore, DC-

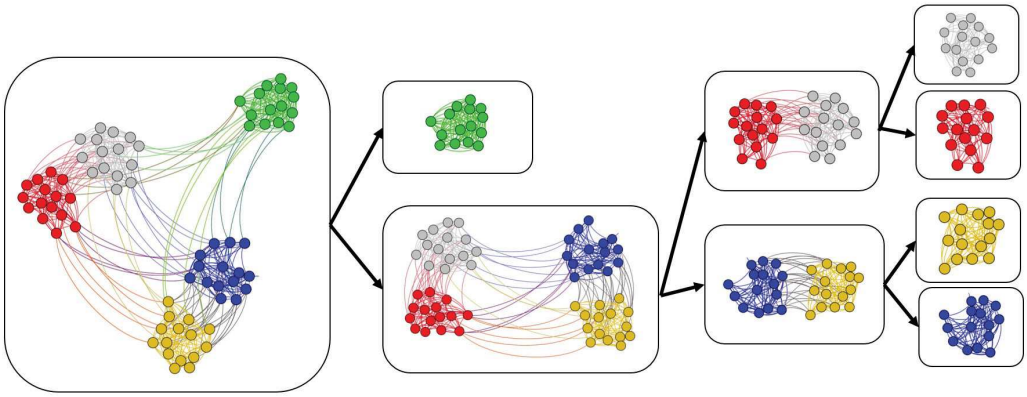


Fig. 3: The proposed DC-EgoTen community detection algorithm on a toy example.

EgoTen nicely proceeds with the desiderata of community identification discussed in Section II, because: i) the multi-dimensional egonet-based representation captures multi-hop connectivities, leading to an improved quality in the detected communities; ii) consecutive division of large communities enhances resolution; and, iii) setting threshold parameter $\tau_k = 1/K$ in EgoTen can guarantee a full coverage of the network, while its tuning can further control the trade-off between coverage and quality.

B. Performance Evaluation

In addition to conductance, normalized mutual information and F1-score are measures for assessing the performance of community identification when ground-truth communities are provided.

Normalized mutual information (NMI) [18]: Given $\mathcal{S}^* = \{\mathcal{C}_1^*, \ldots, \mathcal{C}_{|\mathcal{S}|}^*\}$ and $\hat{\mathcal{S}} = \{\hat{\mathcal{C}}_1, \ldots, \hat{\mathcal{C}}_{|\hat{\mathcal{S}}|}\}$ as ground-truth and detected covers, respectively, the information theoretic measure NMI is defined as (cf. [18])

$$\text{NMI}(\mathcal{S}^*, \hat{\mathcal{S}}) := \frac{2\text{I}(\mathcal{S}^*, \hat{\mathcal{S}})}{\text{H}(\mathcal{S}^*) + \text{H}(\hat{\mathcal{S}})}$$

where $\text{H}(\hat{\mathcal{S}})$ denotes the entropy of set $\hat{\mathcal{S}}$ defined as

$$\text{H}(\hat{\mathcal{S}}) := -\sum_{i=1}^{|\hat{\mathcal{S}}|} p(\hat{\mathcal{C}}_i) \log p(\hat{\mathcal{C}}_i) = -\sum_{i=1}^{|\hat{\mathcal{S}}|} \frac{|\hat{\mathcal{C}}_i|}{N} \log \frac{|\hat{\mathcal{C}}_i|}{N}$$

and similarly for $\text{H}(\mathcal{S}^*)$. Furthermore, $\text{I}(\mathcal{S}^*, \hat{\mathcal{S}})$ denotes the mutual information between $\mathcal{S}^*$ and $\hat{\mathcal{S}}$, defined as

$$\text{I}(\mathcal{S}^*, \hat{\mathcal{S}}) := \sum_{i=1}^{|\mathcal{S}^*|} \sum_{j=1}^{|\hat{\mathcal{S}}|} \frac{|\mathcal{C}_i^* \cap \hat{\mathcal{C}}_j|}{N} \log \frac{N|\mathcal{C}_i^* \cap \hat{\mathcal{C}}_j|}{|\mathcal{C}_i^*||\hat{\mathcal{C}}_j|}. \tag{10}$$

Intuitively, the mutual information $\text{I}(\mathcal{S}^*, \hat{\mathcal{S}})$ reflects a measure of similarity between the two covers. Thus, high values of NMI, namely its maximum at 1, reflect high accuracy in community identification, whereas low values of NMI, namely its minimum at 0, represent poor discovery of the true underlying communities. This measure has been generalized for overlapping communities in [43], and will be utilized for performance assessment in such networks.

Average F1-score [8]: F1-score is a measure of binary classification accuracy. Specifically, the harmonic mean of *precision* and *recall* takes its highest value at 1 and lowest value at 0. Average F1-score for detected cover $\hat{\mathcal{S}}$ is

$$\bar{F1} := \frac{1}{2|\mathcal{S}^*|} \sum_{i=1}^{|\mathcal{S}^*|} F1(\mathcal{C}_i^*, \hat{\mathcal{C}}_{I(i)}) + \frac{1}{2|\hat{\mathcal{S}}|} \sum_{i=1}^{|\hat{\mathcal{S}}|} F1(\mathcal{C}_{I'(i)}^*, \hat{\mathcal{C}}_i)$$

where

$$I(i) = \arg\max_j F1(\mathcal{C}_i^*, \hat{\mathcal{C}}_j), I'(i) = \arg\max_j F1(\mathcal{C}_j^*, \hat{\mathcal{C}}_i)$$

in which $F1(\mathcal{C}_i, \mathcal{C}_j) := \dfrac{2\,|\mathcal{C}_i \cap \mathcal{C}_j|}{|\mathcal{C}_i| + |\mathcal{C}_j|}$.

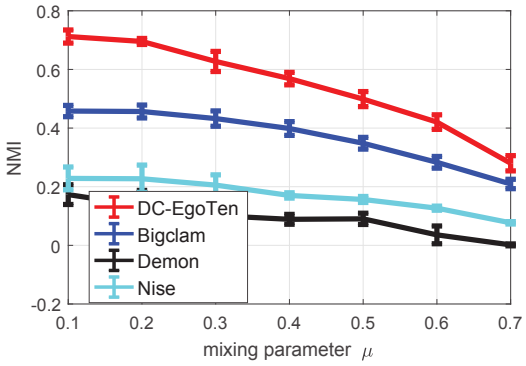
V. Numerical Tests

In this section, the proposed DC-EgoTen is applied to synthetic as well as real datasets. Synthetic Lancicchinetti-Fortunatoand-Radicci (LFR) networks [44] are utilized as a benchmark to study the resilience and performance of different community identification algorithms in the presence of overlapping as well as mixing communities.

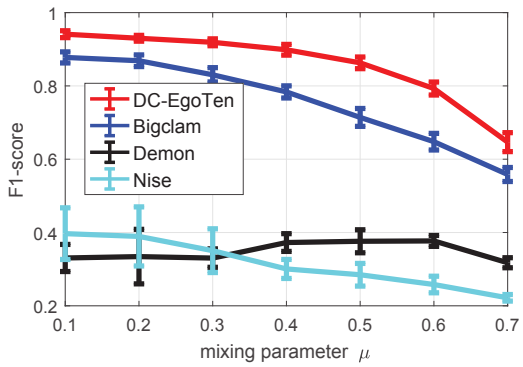
A. LFR Benchmark Networks

LFR graphs serve as benchmark networks in which certain real-world properties, namely power-law distribution for nodal degree and community sizes, as well as the presence of overlapping and mixing communities are preserved. Such networks are configured by a total number of N nodes, $\bar{d}$ average degree, and power-law distribution exponents γ_1 and γ_2 for degree and community sizes, respectively. Furthermore, parameter μ controls the community mixing, where higher values result in more out-of-community edges in between non-resident nodes. Moreover, parameters o_n, o_m respectively set the number of overlapping nodes and communities (with which these nodes are associated).

In order to assess the resilience of the proposed DC-EgoTen to variations of μ and o_n, we have generated networks with

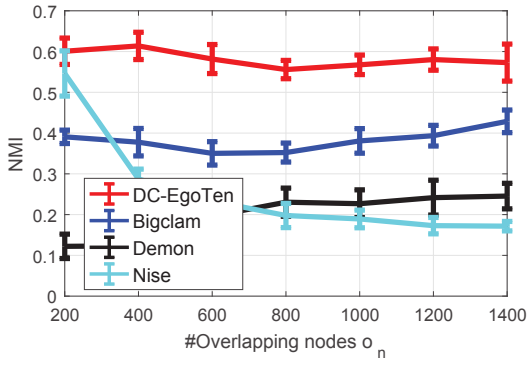


(a)

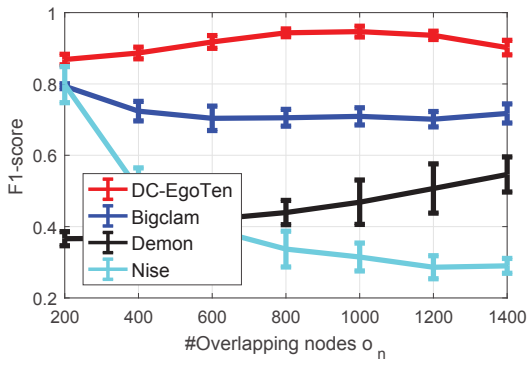


(b)

Fig. 4: Performance of different algorithms versus different community mixing values μ for $o_n = 600$, and $o_m = 3$.



(a)



(b)

Fig. 5: Performance of different algorithms versus different number of overlapping nodes o_n for $\mu = 0.2$, and $o_m = 3$.

$N = 2,000$, $\bar{d} = 100$, $\gamma_1 = 2$, $\gamma_2 = 1$, and varied $\mu \in [0.1, 0.7]$ as well as o_n in $10\% - 70\%$ of the total networks size N, respectively. DC-EgoTen is run by setting the rank K in the initial application as $K = 100$, while following applications are set as $K = 2$, essentially leading to a bisection of the network in the subsequent steps, and sparse tensor decompositions are handled via the SPLATT toolbox [45]. Thresholding parameter τ_k is selected as explained in Section IV for the top EgoTen (allowing for overlapping community detection), and set as $\tau = 1/2$ for next steps. Maximum community size is set as $C_{\max} = 200$. The performance is compared with state-of-the-art algorithms BigClam [8], Demon [46], and Nise [19] with 'spread-hub' seeding strategy, where $|\hat{\mathcal{S}}| = 200$ is provided as an estimate on the number of communities in Nise and BigClam. Due to the availability of underlying communities, the performance is assessed via NMI and F1-scores and averaged over 10 realizations of the network for each setting.

As the results in Figures 4 and 5 corroborate, DC-EgoTen provides higher performance in terms of NMI and F1-score, thanks to the rich egonet-based representation as well as the progressive identification of refined communities.

B. Real-world Networks

In this subsection, the performance of DC-EgoTen is compared with state-of-the-art overlapping community detection algorithms on various real-world networks, listed in Table I, available in [47]. In DC-EgoTen, constructing the egonet-tensors as well as solving the constrained PARAFAC utilize parallel implementation, while Bigclam and Nise also allow for parallel threading. Thus, for networks with $N < 1$ million, these algorithms are run using 8 threads and 32GB of RAM, while for the Youtube dataset, 24 threads with 256 GB of RAM are utilized. As with synthetic datasets, we apply DC-EgoTen with $K = 100$ for the first application of EgoTen, and set $K = 2$ for subsequent steps. Threshold parameter τ_k is selected as explained in Section IV for the top EgoTen (allowing for overlapping community detection), and set as $\tau = 1/2$ for next steps. Also, maximum community size $C_{\max}$ is set to 1% of the network size for each dataset.

Figure 6 plots the run time of different algorithms while Table II lists the coverage and number of detected communities. Due to unavailability of ground-truth communities, NMI and F1-score could not be evaluated, thus performance is assessed using the conductance-coverage curve. To this end, for a given algorithm, the conductance of the identified communities is computed and the communities are sorted accordingly in an

TABLE I: Real-world networks.

Dataset	No. of vertices N	No. of edges $\|\mathcal{E}\|$	Edge type
Facebook	4,039	88,234	Undirected
Enron	36,692	183,831	Undirected
Epinion	75,879	508,837	Directed
Slashdot	82,168	948,464	Directed
Email	265,214	420,045	Directed
Stanford	281,903	2,312,497	Directed
Notredame	325,729	1,497,134	Directed
Youtube	1,134,890	2,987,624	Undirected

increasing order. Conductance-coverage curve is then plotted by increasing the maximum conductance, and progressively adding the sorted communities to the set of covered nodes. Figure 7 depicts the aforementioned curve for various datasets. As low values of conductance correspond to more cohesive communities, a smaller area under curve (AUC) generally implies better performance. However, the resolution of the communities is another important metric which must be considered in drawing conclusions. Interestingly, the separation of different scattered points for a given algorithm in the conductance-coverage curve reveals the granularity of the detected communities. That is, if a detected community is very large, its inclusion creates a jump in the coverage, which is noticeable by the two consecutive points in the plot being placed far apart. Thus, examining Figure 7 reveals that the identified communities via DC-EgoTen and Bigclam are usually of more refined sizes as those plots are always smooth, while the performance of Nise and Demon is often limited to detecting very large communities (upto 40% of the whole network). Furthermore, although one may not particularly be interested in 100% coverage, it is desirable that a relatively high number of nodes to be covered whithin the detected communities, and thus low coverage where more than 50% of the nodes are left uncovered is considered undesirable.

VI. Conclusion

This work dealt with identification of overlapping communities via DC-EgoTen, a top-to-bottom tensor-based framework. Specifically, a novel egonet-based tensor representation of a network was introduced and utilized in a constrained PARAFAC decomposition, whose factors subsequently reveal the underlying communities. To provide the detected communities with desirable resolution, this algorithm was applied progressively in a top-to-bottom fashion, where the network is decomposed into K communities per step. Parallel implementation as well as exploitation of the sparsity in the egonet-tensor endow the algorithm with scalability, while the structured redundancy and the rich representational capacity of the egonet-tensor enhance the performance of the toolbox. Sparse sampling of egonets along the third mode is among our future directions, through which memory as well as computational requirements of the algorithm can be reduced, while the structured redundancy in the egonet-tensor is expected to preserve performance.

References

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

[2] P. Du, J. Gong, E. S. Wurtele, and J. A. Dickerson, "Modeling gene expression networks using fuzzy logic," *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 6, pp. 1351–1359, Dec 2005.

[3] J. D. Power, A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar, and S. E. Petersen, "Functional network organization of the human brain," *Neuron*, vol. 72, no. 4, pp. 665–678, 2011.

[4] Y. R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Analyzing communities and their evolutions in dynamic social networks," *ACM Trans. on Knowledge Discovery from Data*, vol. 3, no. 2, p. 8, Paris, France, April 2009.

[5] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, Beijing, China, 2012.

[6] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *Proc. of Advances in Neural Inf. Proc. Systems*, pp. 33–40, Vancouver, Canada, Dec. 2009.

[7] A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade, "A tensor approach to learning mixed membership community models," *J. of Machine Learning Research*, vol. 15, no. 1, pp. 2239–2312, Jan. 2014.

[8] J. Yang and J. Leskovec, "Overlapping community detection at scale: A nonnegative matrix factorization approach," *Proc. of ACM Intl. Conf. on Web Search and Data Mining*, pp. 587–596, Rome, Italy, Feb. 2013.

[9] I. Derényi, G. Palla, and T. Vicsek, "Clique percolation in random networks," *Physical Review Letters*, vol. 94, no. 16, pp. 160 202–1:4, 2005.

[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, pp. P10 008:1–12, 2008.

[11] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical Review E*, vol. 72, no. 2, pp. 027 104:1–4, 2005.

[12] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Aug. 2007.

[13] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, "Community discovery using nonnegative matrix factorization," *ACM Trans. on Data Mining and Knowledge Discovery*, vol. 22, no. 3, pp. 493–521, Paris France, July 2011.

[14] X. Cao, X. Wang, D. Jin, Y. Cao, and D. He, "Identifying overlapping communities as well as hubs and outliers via nonnegative matrix factorization," *Scientific Reports*, vol. 3, p. 2993, 2013.

[15] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, "Overlapping community detection using Bayesian non-negative matrix factorization," *Physical Review E*, vol. 83, no. 6, p. 066114, 2011.

[16] Y. Zhang and D. Y. Yeung, "Overlapping community detection via bounded nonnegative matrix tri-factorization," *Proc. of ACM Intl. Conf. on Knowledge Discovery and Data Mining*, pp. 606–614, Beijing, China, 2012.

[17] Z. Y. Zhang, Y. Wang, and Y. Y. Ahn, "Overlapping community detection in complex networks using symmetric binary matrix factorization," *Physical Review E*, vol. 87, no. 6, p. 062803, 2013.

[18] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, Feb. 2010.

[19] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using neighborhood-inflated seed expansion," *IEEE Trans. on Knowledge and Data Engineering*, vol. 28, no. 5, pp. 1272–1284, May 2016.

[20] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, 2004.

[21] S. Gregory, "An algorithm to find overlapping community structure in networks," *European Conf. on Principles of Data Mining and Knowledge Discovery*, pp. 91–102, Warsaw, Poland, 2007.

[22] M. J. Rattigan, M. Maier, and D. Jensen, "Graph clustering with network structure indices," in *Proc. Intl. Conf. on Machine Learning*, Corvallis, OR, USA, 2007, pp. 783–790.

[23] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using seed set expansion," *Proc. of ACM Intl. Conf. on Information and Knowledge Management*, pp. 2099–2108, San Diego, CA, USA, 2013.

TABLE II: Coverage and number of detected communities of different methods over real-world networks.

Dataset		DC-EgoTen	Bigclam	Demon	Nise
Facebook	Coverage	100%	95%	99%	89%
	No. of comm.	523	500	8	16
Enron	Coverage	100%	90%	65%	100%
	No. of comm.	553	500	343	520
Slashdot	Coverage	100%	100%	95%	100%
	No. of comm.	1163	500	51	485
Epinion	Coverage	100%	100%	35%	100%
	No. of comm.	1274	2000	136	2041
Email	Coverage	100 %	83%	11%	100%
	No. of comm.	965	2000	24	2404
Notredame	Coverage	100%	100%	39%	100%
	No. of comm.	1169	2000	1497	1454
Stanford	Coverage	100%	90%	85%	100%
	No. of comm.	807	2000	2596	1411
Youtube	Coverage	100%	100%	22%	100%
	No. of comm.	813	5000	3835	5162

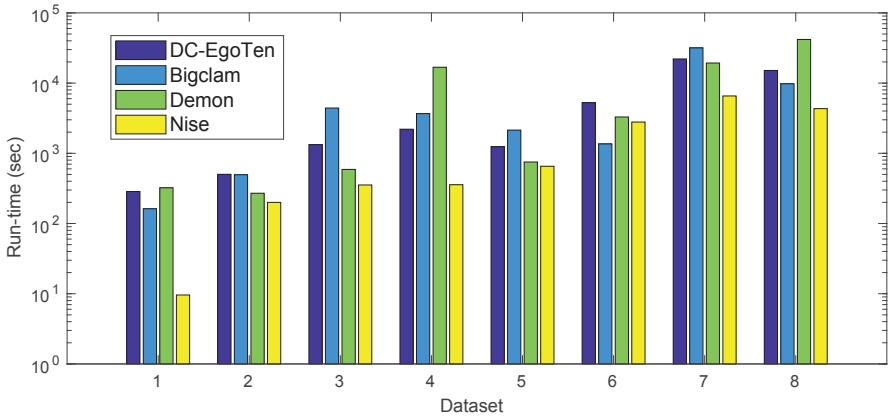


Fig. 6: Runtime of different algorithms on various datasets denoted on the x-axis as: (D1) Facebook, (D2) Enron, (D3) Epinion, (D4) Email, (D5) Slashdot, (D6) Notredame, (D7) Stanford, and (D8) Youtube.

[24] K. He, Y. Sun, D. Bindel, J. Hopcroft, and Y. Li, "Detecting overlapping communities from local spectral subspaces," *IEEE Intl. Conf. on Data Mining*, pp. 769–774, 2015.

[25] J. J. Whang, I. S. Dhillon, and D. F. Gleich, "Non-exhaustive, overlapping k-means," *Proc. of SIAM Intl. Conf. on Data Mining*, pp. 936–944, 2015.

[26] E. E. Papalexakis, L. Akoglu, and D. Ience, "Do more views of a graph help? Community detection and clustering in multi-graphs," *IEEE Intl. Conf. on Information Fusion*, pp. 899–905, Istanbul, Turkey, 2013.

[27] E. E. Papalexakis, N. D. Sidiropoulos, and R. Bro, "From k -means to higher-way co-clustering: Multilinear decomposition with sparse latent factors," *IEEE Trans. on Signal Proc.*, vol. 61, no. 2, pp. 493–506, Jan. 2013.

[28] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," *IEEE Intl. Conf. on Data Mining*, pp. 1151–1156, Dallas, TX, USA, 2013.

[29] 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," *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pp. 271–283, Tairan Taiwan, Aug. 2014.

[30] B. Baingana and G. B. Giannakis, "Joint community and anomaly tracking in dynamic networks," *IEEE Trans. on Signal Proc.*, vol. 64, no. 8, pp. 2013–2025, 2016.

[31] F. Huang, U. Niranjan, M. U. Hakeem, and A. Anandkumar, "Online tensor methods for learning latent variable models," *J. of Machine Learning Research*, vol. 16, pp. 2797–2835, 2015.

[32] A. R. Benson, D. F. Gleich, and J. Leskovec, "Tensor spectral clustering for partitioning higher-order network structures," *Proc. of SIAM Intl. Conf. on Data Mining,*, pp. 118–126, Vancouver, Canada, Feb. 2015.

[33] T. G. Kolda, B. W. Bader, and J. P. Kenny, "Higher-order web link analysis using multilinear algebra," *IEEE Intl. Conf. on Data Mining*, pp. 8–pp, Houston, Texas, 2005.

[34] F. Sheikholeslami, B. Baingana, G. B. Giannakis, and N. D. Sidiropolous, "Egonet tensor decomposition for community identification," *Proc. of Globalsip*, DC, Dec. 2016.

[35] L. Akoglu, M. McGlohon, and C. Faloutsos, "Oddball: Spotting anomalies in weighted graphs," *Advances in Knowledge Discovery and Data Mining*, pp. 410–421, Washington, DC, USA, 2010.

[36] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[37] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Parcube: Sparse parallelizable tensor decompositions," *Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*, pp. 521–536, 2012.

[38] S. Smith, A. Beri, and G. Karypis, "Constrained tensor factorization with accelerated AO-ADMM," in *IEEE Intl. Conf. on Parallel Processing*, Bristol, UK, Aug. 2017.

[39] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proc. of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[40] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Rank regularization and Bayesian inference for tensor completion and extrapolation," *IEEE Trans. on Signal Proc.*, vol. 61, no. 22, pp. 5689–5703, Nov. 2013.

[41] K. Huang, N. D. Sidiropoulos, and A. P. Liavas, "A flexible and efficient algorithmic framework for constrained matrix and tensor factorization,"

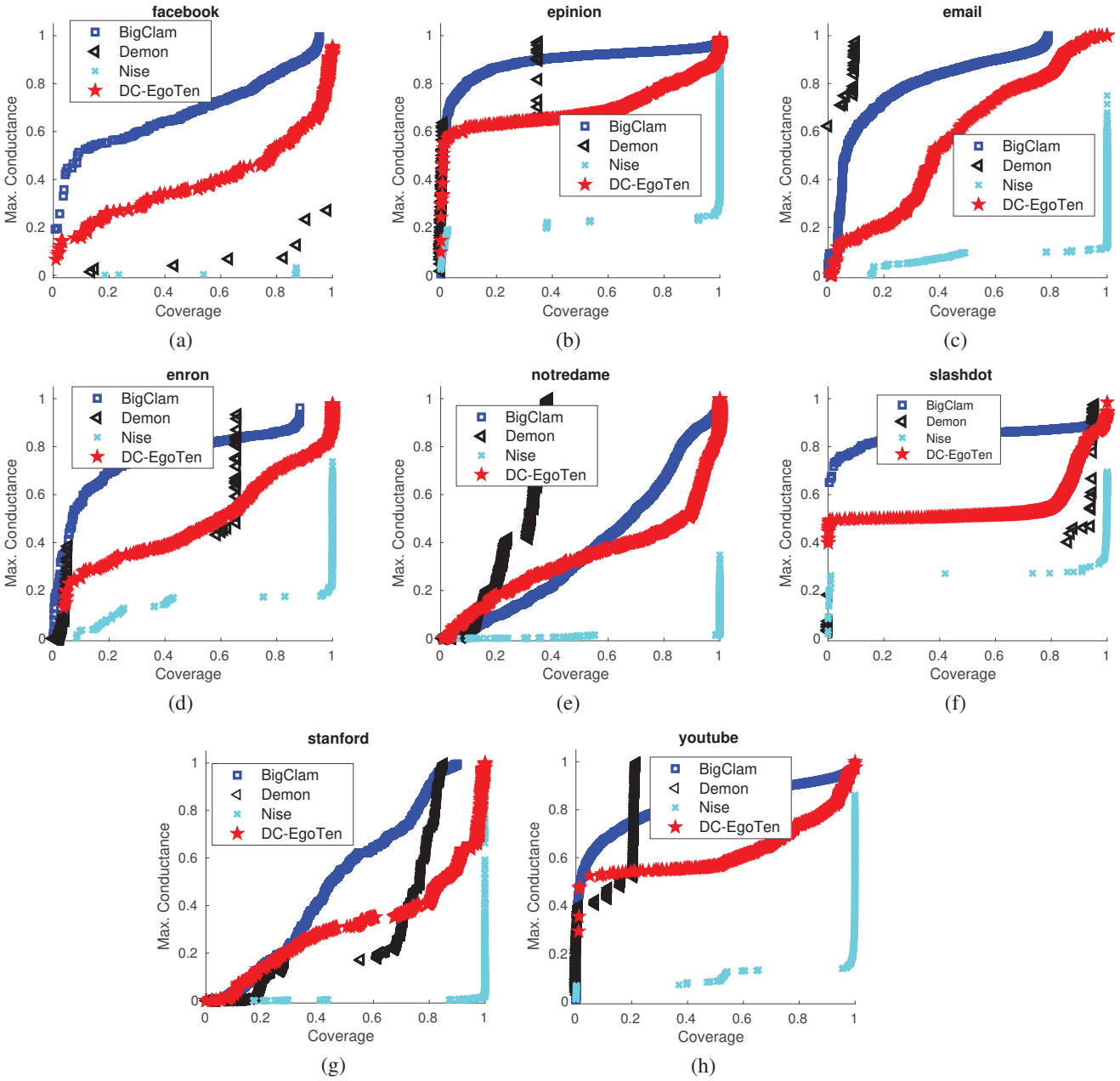


Fig. 7: Conductance-coverage curve for various datasets using different community detection algorithms.

IEEE Trans. on Signal Processing, vol. 64, no. 19, pp. 5052–5065, 2016.

[42] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," *Proc. Intl. Conf. on Machine Learning*, pp. 272–279, 2008.

[43] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, p. 033015, 2009.

[44] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, p. 046110, 2008.

[45] S. Smith and G. Karypis, "SPLATT: The Surprisingly ParalleL spArse Tensor Toolkit," http://cs.umn.edu/~splatt/.

[46] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi, "Demon: A local-first discovery method for overlapping communities," pp. 615–623, 2012.

[47] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.