Data Min Knowl Disc (2013) 27:225–258
DOI 10.1007/s10618-012-0289-3

Discovery of extreme events-related communities in contrasting groups of physical system networks

Zhengzhang Chen · William Hendrix · Hang Guan · Isaac K. Tetteh · Alok Choudhary · Fredrick Semazzi · Nagiza F. Samatova

Received: 21 October 2011 / Accepted: 6 August 2012 / Published online: 1 September 2012

Abstract The latent behavior of a physical system that can exhibit extreme events such as hurricanes or rainfalls, is complex. Recently, a very promising means for studying complex systems has emerged through the concept of complex networks. Networks representing relationships between individual objects usually exhibit community dynamics. Conventional community detection methods mainly focus on either mining frequent subgraphs in a network or detecting stable communities in time-varying networks. In this paper, we formulate a novel problem—*detection of predictive and phase-biased communities in contrasting groups of networks*, and propose an efficient and effective machine learning solution for finding such anomalous communities. We build different groups of networks corresponding to different system's phases, such as higher or low hurricane activity, discover phase-related system components as seeds to help bound the search space of community generation in each network, and use the proposed contrast-based technique to identify the changing communities across different groups. The detected anomalous communities are hypothesized (1) to play an important role in defining the target system's state(s) and (2) to improve the predictive

Responsible editor: Eamonn Keogh.

Z. Chen · W. Hendrix · I. K. Tetteh · F. Semazzi · N. F. Samatova
North Carolina State University, Raleigh, NC 27695, USA

Z. Chen · N. F. Samatova (✉)
Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
e-mail: samatova@csc.ncsu.edu

H. Guan
Zhejiang University, Hangzhou, 31000 Zhejiang, China

A. Choudhary
Northwestern University, Evanston, IL 60201, USA

skill of the system's states when used collectively in the ensemble of predictive models. When tested on the two important extreme event problems—identification of tropical cyclone-related and of African Sahel rainfall-related climate indices—our algorithm demonstrated the superior performance in terms of various skill and robustness metrics, including 8–16 % accuracy increase, as well as physical interpretability of detected communities. The experimental results also show the efficiency of our algorithm on synthetic datasets.

Keywords Spatio-temporal data mining · Complex network analysis · Community detection · Comparative analysis · Network motif detection · Extreme event prediction

1 Introduction

Recent studies of the structure, dynamics, and function of complex networks have witnessed a growing interest. Such complex networks model a variety of systems including societies, ecosystems, the Internet, and others (Newman 2003). In particular, climate networks have lately emerged as a promising approach for modeling spatio-temporal dynamics of the climate system (Gozolchiani et al. 2008; Steinhaeuser et al. 2011; Tsonis and Swanson 2008; Tsonis et al. 2010). In these climate networks, nodes (or oscillators) represent spatial grid points, and the edges between pairs of nodes exist depending on the degree of statistical interdependence between the corresponding pairs of time series taken from the climate data set (Tsonis and Roebber 2004).

Complex networks have enabled hypothesis-driven insights about the intricate interplay between the topology and dynamics of the physical system at different scales. For example, on the global scale, climate networks exhibit "small-world" properties due to teleconnections (i.e., edges linking geographically distant nodes), such as those in El Niño and La Niña climate networks (Tsonis and Swanson 2008; Gozolchiani et al. 2008), that stabilize the climate system and enhance the energy and information transfer within the system (Tsonis et al. 2006, 2008). Likewise, the collective behavior of interacting subsystems in a network of different climate indices has explained the great climate shifts of the twentieth century as synchronized transitions between different equilibria of oscillators representing the earth system (Tsonis et al. 2007).

To complement these fruitful hypothesis-driven studies, data-driven approaches to discovery of predictive insights from complex networks have emerged (Steinhaeuser et al. 2009; Ganguly et al. 2009). A representative example of such approaches focuses on detecting and characterizing the community structure, in which nodes are grouped into communities with more interactions (i.e., edges) within communities and fewer interactions between communities. A community is a common structure in many real-world networks (Girvan and Newman 2002; Tsonis et al. 2010), including social networks, biological networks, and climate networks. However, the enormous size and the intrinsic complexity of the system data used for network construction challenge existing graph-based approaches and call for a paradigm shift in how the networks are analyzed.

Comparative analysis of multiple networks is a promising strategy. It can be performed at multiple levels for the purpose of (a) understanding climate dynamics

over different time periods, (b) comparing multiple climate simulation models, (c) quantifying the agreement between climate simulation and observation data, or (d) correlating networks derived for different climate variables. Such analyses could translate to different problems on graphs, such as finding conserved network motifs to detect and track climate regions of similar behavior, or communities, over subsequent time windows (Steinhaeuser et al. 2009), or graph-based anomaly detection to identify which communities have grown/contracted, merged/split, or born/vanished (Chen et al. 2011).

It is often the case that such multiple networks could be partitioned into different groups, such as those corresponding to different system phases; it is a known fact that a dynamic physical system often undergoes phase transitions in response to fluctuations induced on system parameters (Hey et al. 2009). For example, in a tropical cyclone (TC) prediction system, one can build three different groups of climate networks, with one corresponding to high TC years, and another corresponding to medium TC years, and the other one for low TC years. Different groups of networks may exhibit different properties of the community structure. The question is how one could discover network motifs that could contribute to our understanding of the system's behavior for a given phase.

In this paper, we hypothesize that anomalous communities, or dense subnetworks that are conserved within one group of networks but undergo statistically significant structural transformation in the other groups of networks, could be candidate structures for explaining physical basis underlying the group-related extreme events. For example, if an anomalous community corresponding to the El Niño/La Niña–Southern Oscillation (ENSO) climate index is identified, then the changes in such a community structure would explain why a particular season would enjoy low tropical cyclone activity or would be affected by the severity of the abnormally high number of hurricanes (Camargo et al. 2010). It is thus important to find effective means for detecting anomalous communities in contrasting (or system phase-related) groups of networks. To the best of our knowledge, such a problem has not been addressed before in literature.

It is worth noticing that performing such analyses for larger-scale, high-resolution physical models and over multiple heterogeneous data sources is a challenging problem not only computationally but also methodologically. For example, current algorithms for identifying conserved network motifs are limited in either the size (Borgelt and Berthold 2002; Peng et al. 2008) or the number (Kalaev et al. 2008; Sharan et al. 2005) of networks they can effectively compare; plus they are not particularly designed for contrasting groups of networks. To detect the differences, one may want to find those communities that are conserved across dynamic networks derived from one data source but not conserved for the other data source. However, most algorithms do not support such contrast-based detection and tend to require that the motif be conserved in every one of the input networks (Kalaev et al. 2008; Sharan et al. 2005). While some comparative techniques have been designed for the biological networks (Gill et al. 2010; Zhang et al. 2009), they only consider the structural or topological differences between pairs of networks. Similarly, previous work has been done on finding dense subgraphs that are present in a majority (Zeng et al. 2007) or every member of a set of graphs (Pei et al. 2005), but neither of these are applicable to contrasting groups of networks, nor

can they identify anomalous communities. Likewise, graph-based anomaly detection has been mainly focused on identifying anomalous nodes (Moonesinghe and Tan 2006; Sun et al. 2005), unusual edges (Chakrabarti 2004), or small abnormal patterns (Eberle and Holder 2007) in a single graph, with few exceptions focusing on graph-based discovery of anomalies in noisy multivariate time series data (Cheng et al. 2008), for multiple data sources (Sun et al. 2006), and across multiple graphs (Chan and Mahoney 2005; Chen et al. 2011; Sun et al. 2007). However, none of these approaches provides a means for detecting anomalous communities in contrasting groups of graphs.

Our approach follows from the need to address the graph classification problem of detecting predictive and phase-biased anomalous communities in contrasting groups of networks. We build groups of networks corresponding to different system phases, detect system phase-related components as seeds to help prune the search space in community generation, and use the proposed contrast-based techniques to discover abnormal communities that are further used to build the ensemble of classifiers for predicting the system states/phases.

2 Definitions and theorems

In this paper, the ultimate goal is to detect and track phase-biased communities in contrasting groups of networks. Thus, in this section, we first provide some formal definitions related to the community structure of a network. Next, we present a number of theoretical results that help bound our search for the communities of a network. A weighted undirected graph is used to represent a complex network in this paper.

Definition 1 (*Community*) A community is a dense subgraph or a group of vertices within which the connections are denser than between different groups (Girvan and Newman 2002).

In other words, a community is a "fuzzy cluster," or a quasi-clique, but not necessarily a "formal clique" with a set of vertices that are all adjacent to one another.

To be more specific, the community structure can be defined:

Definition 2 (γ-*dense Community*) Given a labeled graph G and a real value $\gamma \in [0.5, 1]$, a subgraph S of G is a γ-*dense community*, if and only if every vertex of S is adjacent to at least $\gamma(|S| - 1)$ of the other vertices of S (Pei et al. 2005; Zeng et al. 2006).

The advantage of this community definition is twofold. First, it corresponds nicely with the typical use of the term "density" in that it forces a certain fraction of the possible edges in the subgraph to exist. The second advantage is that our definition must be satisfied by every vertex of the community, ensuring that each vertex "belongs" to the community. One disadvantage of this definition is that it is not monotone; that is, a superset or subset of a γ-dense community does not need to be γ-dense, though basing our definition on the density of the subgraph rather than a maximum number of disconnections (as in a k-plex Balasundaram et al. 2011; Seidman and Foster 1978) gives us more flexibility in finding large subgraphs.

If a γ-dense-community contains a number of vertices in a seed or query set, we call it μ-enriched γ-dense community:

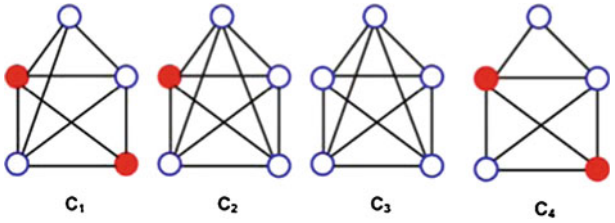

Fig. 1 An example of (μ, γ)-communities. *Filled nodes*: seed nodes, *empty nodes*: normal nodes

Definition 3 ((μ, γ)-*community*) Given a labeled graph G, a "seed" set of vertices Q, a real value $\gamma \in [0.5, 1]$, and a real value $\mu \in [0, 1]$, a γ-dense community S is μ-*enriched* with respect to Q, if and only if at least $\mu|S|$ vertices of S are contained in Q.

The "seed" set Q can be used to incorporate the domain scientists' knowledge. For example, we can take in a biologist's prior knowledge as a set of "seed" proteins and identify all the communities in a biological network that contain some part of the "seed" proteins.

Figure 1 shows an example of (μ, γ)-communities. If we set $\mu = 0.2$ and $\gamma = 0.75$, then only C_1 and C_2 in Fig. 1 can be considered (0.2, 0.75)-communities. Subgraph C_3 is not a (0.2, 0.75)-community, because it does not contain any "seed" node. Although subgraph C_4 has two "seed" nodes, not all of the vertices in C_4 are adjacent to at least three (i.e. $0.75 * (5 - 1)$) of the other nodes. Thus, it is also not a (0.2, 0.75)-community. But if we relax the requirements to be $\mu = 0$ and $\gamma = 0.5$, then all four subgraphs can be considered as communities.

One of the main ways in which (μ, γ)-communities differ from traditional communities, such as those produced by modularity-based clustering algorithms (e.g., Clauset et al. 2004; Wakita and Tsurumi 2007) is that (μ, γ)-communities are allowed to overlap. As climatological factors in a particular region may contribute to multiple system events, this is a very desirable feature for a community detection algorithm to have in the climate domain, as well as other scientific domains like biological networks, where pathways or gene modules work in a cross-talking manner. While such algorithms may have other advantages, such as the parameter-free nature of clustering algorithms that maximizes modularity, and might work better for other domains like social networks, these algorithms are partitional by nature and typically heuristic, giving no guarantees of global optimality or the quality of individual communities.

Definition 4 (*Corresponding Community*) Given two communities $C_{i,m}$ and $C_{j,n}$ belong to networks G_m and G_n, $C_{j,n}$ is a corresponding community to $C_{i,m}$ if and only if $\frac{|C_{i,m} \bigcap C_{j,n}|}{|C_{i,m} \bigcup C_{j,n}|} > \alpha$, where $\alpha \in (0, 1]$ and $m \neq n$, and $|C|$ is the number of vertices in the community.

For example, in Fig. 2, community $\{V_1, V_2, V_3, V_4, V_6\}$ of graph G_2 and community $\{V_2, V_3, V_4\}$ of graph G_4 are both corresponding communities to community $\{V_1, V_2, V_3, V_4\}$ in graph G_1, if we set $\alpha = 0.6$, $\mu = 0.1$, and $\gamma = 0.75$. Thus, each

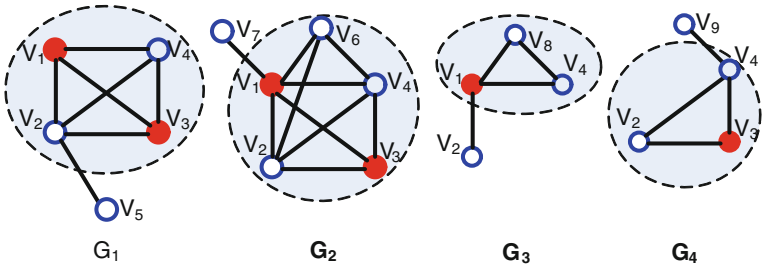

Fig. 2 An example of corresponding communities and conserved communities. *Filled nodes*: seed nodes, *empty nodes*: normal nodes, *dashed circles*: communities

community can have multiple corresponding communities, and each corresponding community can correspond to several communities.

Definition 5 (*Conserved Community*) Given a set of k different networks $\{G_1, G_2, \ldots, G_k\}$, a community $C_{i,m}$ of graph G_m, where $1 \leq m \leq k$, is an (α, β)-conserved community, if and only if $C_{i,m}$ has an α-corresponding community in more than $\beta \times k$ networks, where $\alpha \in (0, 1]$ and $\beta \in [0.5, 1]$. If both α and β are larger than or equal to 0.5, we call this community a stable community in a group of networks.

For example, community $\{V_1, V_2, V_3, V_4\}$ in graph G_1 (Fig. 2) can be considered as a conserved community, if α, β, μ and γ are set to be 0.6, 0.75, 0.1, and 0.75, respectively. Although community $\{V_1, V_2, V_3, V_4\}$ does not have any corresponding community in graph G_3, it still meets the requirement of a conserved community, because it has corresponding communities in the other two graphs.

Definition 6 (*Anomalous Community*) Given τ different groups of networks $\{U_1, U_2, \ldots, U_\tau\}$, a community C is an anomalous community if and only if C is an (α, β)-conserved community in one group of networks U_j, where $1 \leq j \leq \tau$, $\alpha \in [0.5, 1]$ and $\beta \in [0.5, 1]$, but C has no ω-corresponding community among the (α, β)-conserved communities of all the other groups of networks, where $\omega \in (0, \alpha)$.

Figure 3 shows an example of anomalous communities, where ω is set to be 0.4, and we assume that C_{11}, C_{12}, C_{21}, and C_{22} are conserved communities detected from two different groups of networks, U_1 and U_2 with $\alpha = 0.6$ and $\beta = 0.75$. C_{12} and C_{22} are anomalous communities, because they do not have any ω-corresponding community among the conserved communities of the other group.

Problem 1 (*Detecting Predictive and Phase-biased Communities in Contrasting Groups of Complex Networks*) Given a *multi-phase* system that can be characterized by different groups of networks, the problem is to detect all the anomalous communities that are biased toward a target system phase from the training networks, and utilize all the detected phase-biased communities as the features to build an ensemble of classifiers to predict the unknown system phases on the testing data.

According to the statement of Problem 1, the main goal of our technique is to create an ensemble classifier for determining the phase-state of a network based on

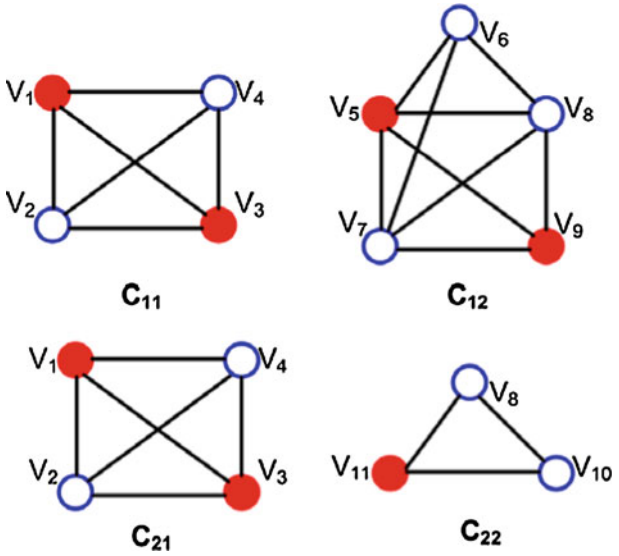

Fig. 3 An example of anomalous communities. C_{11} and C_{12} are conserved communities from the network group U_1, and C_{21} and C_{22} are conserved communities from the network group U_2. *Filled nodes*: seed nodes, *empty nodes*: normal nodes

the phase-biased communities detected in the training set. Given a set of networks, we form this ensemble by: (1) identifying phase-related system components, (2) enumerating the (μ, γ)-communities enriched by these phase-related components, (3) identifying phase-biased communities, and (4) forming a classifier ensemble, where each member predicts the phase-state of a network based on the features in these phase-biased communities.

Next, we present a number of theoretical results that will allow us to efficiently enumerate all of the (μ, γ)-communities of a network. This step of our technique is described further in Sect. 3.5. To enumerate all of these communities, we first observe that so long as $\mu > 0$, every (μ, γ)-community must contain at least one "seed" node, so we can enumerate all (μ, γ)-communities by iteratively finding all of the (μ, γ)-communities that contain each "seed" vertex. In order to enumerate all of the (μ, γ)-communities that contain a specific "seed" vertex, we use the following theorem in order to establish an initial set of *candidate* vertices that can be added to the seed vertex to form a (μ, γ)-community.

Let the *distance* between two vertices be the length of the shortest path between them.

Theorem 1 *Let S be a subgraph of G. For* $\gamma \geq 0.5$*, any* γ*-dense community that is a supergraph of S will consist of vertices at most distance* 2 *from every vertex of S. We denote this set as* $\mathbf{N^2(S)}$.

Proof Let C be a γ-dense community with $\gamma \geq 0.5$, and suppose there are two vertices, u and v in C that are distance 3 apart. As u and v are at distance 3, they must have no neighbors in common. As $\gamma \geq 0.5$, u and v must each be adjacent to two disjoint

sets of at least $(0.5)(|C|-1)$ vertices, not including u and v. Thus, between u, v, and their neighborhoods, there are at least $1+1+2(0.5)(|C|-1) = |C|+1$ vertices in C. This is clearly impossible, so there must not be any vertices in C that are distance 3 apart. Thus, the diameter of any γ-dense community with $\gamma \geq 0.5$ can be at most 2. □

By methodically exploring all subgraphs of the vertices within distance 2 of a seed vertex, we can be assured that we will find all of the (μ, γ)-communities. However, as this set of subgraphs may form a very large search space, we wish to establish some additional results that will allow us to prune the search space without missing any of the (μ, γ)-communities (see Algorithm 2 in Sect. 3).

The remaining theorems will enable us to eliminate candidates from consideration, reducing the search space for the algorithm. The intuition behind Theorems 2–3 is that if we are unable to satisfy the density or enrichment requirements after adding all of the candidate vertices that are query vertices or adjacent to a vertex of our (μ, γ)-community, then we know that there is no way to form a (μ, γ)-community from the current subgraph, and we needn't explore this branch of the search space. Theorems 4–7 deal with placing limits on the number of non-adjacent query candidate vertices or non-query adjacent candidates vertices we may add to our subgraph based on a combination of the density and enrichment requirements.

Theorem 2 *Let S be a subgraph of G, and let V be the set of all possible vertices that may be contained in some supergraph of S that is a γ-dense community. Let v be any vertex of S, and let s_a and c_a be the number of vertices of S and V, respectively, that are adjacent to v. If $s_a + c_a < \gamma(|S| - 1 + c_a)$, no supergraph of S can be a γ-dense community. And if $s_a + c_a \leq \gamma(|S| + c_a)$, neither $S \cup \{v\}$ nor any supergraph of it can be a γ-dense community.*

Proof Suppose the negation: $s_a + c_a < \gamma(|S| - 1 + c_a)$ and there exists a supergraph H of S that is a γ-dense community.

Let h_a be the number of vertices of $H \setminus S$ adjacent to v. By the definition of the set V, $h_a \leq c_a$. Subtracting $\gamma(c_a - h_a)$ from both sides of the previous inequality, we see that $s_a + c_a - \gamma(c_a - h_a) < \gamma(|S| - 1 + h_a)$. As $0 < \gamma \leq 1$ and $c_a - h_a \geq 0$, $c_a - h_a \geq \gamma(c_a - h_a)$, so

$$\begin{aligned} s_a + h_a &= s_a + c_a - (c_a - h_a) \\ &\leq s_a + c_a - \gamma(c_a - h_a) \\ &< \gamma(|S| - 1 + h_a) \end{aligned}$$

Since H must have at least $|S| + h_a$ vertices, $\gamma(|S| - 1 + h_a) \leq \gamma(|H| - 1)$, so $s_a + h_a < \gamma(|H| - 1)$. However, H has only $s_a + h_a$ vertices adjacent to v, implying that H is not a γ-dense community. This is a contradiction; therefore, the claim must be true.

Similarly, we can prove that if $s_a + c_a \leq \gamma(|S| + c_a)$, neither $S \cup \{v\}$ nor any supergraph of it can be a γ-dense community. □

Theorem 3 *Let c_q be the number of vertices in $V \cap Q$. If there are fewer than $\mu|S| - (1-\mu)c_q$ vertices in $S \cap Q$, then neither S nor any supergraph will be μ-enriched.*

Proof Suppose not, and let H be a supergraph of S that is a μ-enriched. Let h_q be the number of vertices of $H \setminus S$ that are in Q. Since V contains c_q vertices in Q, $h_q \leq c_q$. As S contains less than $\mu|S| - (1-\mu)c_q$ vertices in Q, H contains less than $\mu|S| - (1-\mu)c_q + h_q \leq \mu|S| - (1-\mu)h_q + h_q = \mu(|S| + h_q)$ vertices in Q. However, as $|H| \geq |S| + h_q$, this implies that H contains less than $\mu|H|$ vertices in Q, contradicting our assumption that H is μ-enriched. □

Lemma 1 *Let H be a (μ, γ)-community, let S be a subgraph of H, and let v be a vertex of S. Let s_a be the number of vertices of S adjacent to v, $s_{\overline{q}}$ be the number of vertices in S that are not in Q, c_{aq} be the number of vertices in $H \setminus S$ that are in Q and adjacent to v, and $c_{a\overline{q}}$ be the number of vertices in $H \setminus S$ that are adjacent to v but not in Q.*

$$(1-\mu)(s_a + c_{aq} + c_{a\overline{q}} + \gamma) - \gamma(s_{\overline{q}} + c_{a\overline{q}}) \geq 0$$

Proof H must contain at least $\mu|H|$ vertices in Q, so it contains at most $|H| - \mu|H| = (1-\mu)|H|$ vertices not in Q. By our definitions, $s_{\overline{q}}$ and $c_{a\overline{q}}$ both represent vertices in H that are not in Q, so this fact implies that

$$s_{\overline{q}} + c_{a\overline{q}} \leq (1-\mu)|H|. \tag{1}$$

Similarly, H must contain at least $\gamma(|H| - 1)$ vertices adjacent to v, so

$$\gamma(|H| - 1) \leq s_a + c_{aq} + c_{a\overline{q}}, \text{ or} \tag{2}$$
$$\gamma|H| \leq s_a + c_{aq} + c_{a\overline{q}} + \gamma. \tag{3}$$

Multiplying Eq. 1 by γ and Eq. 3 by $1 - \mu$ (both of which must be positive), we see that

$$\gamma(s_{\overline{q}} + c_{a\overline{q}}) \leq \gamma(1-\mu)|H| \text{ and} \tag{4}$$
$$\gamma(1-\mu)|H| \leq (1-\mu)(s_a + c_{aq} + c_{a\overline{q}} + \gamma). \tag{5}$$

Thus, $\gamma(s_{\overline{q}} + c_{a\overline{q}}) \leq (1-\mu)(s_a + c_{aq} + c_{a\overline{q}} + \gamma)$, proving the claim.

Theorem 4 *Let v be a vertex in S. Let s_a be the number of vertices of S adjacent to v, $s_{\overline{q}}$ be the number of vertices in S that are not in Q, c_{aq} be the number of vertices in V that are in Q and adjacent to v, and $c_{a\overline{q}}$ be the number of vertices in V that are adjacent to v but not in Q.*

If $\gamma < 1 - \mu$ and $(1-\mu)(s_a + c_{aq} + c_{a\overline{q}} + \gamma) - \gamma(s_{\overline{q}} + c_{a\overline{q}}) < 0$, then neither S nor any supergraph of S will be a (μ, γ)-community.

Proof Consider an arbitrary (μ, γ)-community H such that S is a subgraph of H. If we let h_{aq} and $h_{a\overline{q}}$ represent the number of vertices in $H \setminus S$ that are adjacent to v

and are in Q and not in Q, respectively, then h_{aq} and $h_{a\overline{q}}$ must satisfy the inequality $(1-\mu)(s_a + h_{aq} + h_{a\overline{q}} + \gamma) - \gamma(s_{\overline{q}} + h_{a\overline{q}}) \geq 0$ by Lemma 1. As H must be a subgraph of $S \cup V$, h_{aq} and $h_{a\overline{q}}$ must satisfy $0 \leq h_{aq} \leq c_{aq}$ and $0 \leq h_{a\overline{q}} \leq c_{a\overline{q}}$. As $\gamma < 1-\mu$, $(1-\mu)(s_a + h_{aq} + h_{a\overline{q}} + \gamma) - \gamma(s_{\overline{q}} + h_{a\overline{q}})$ is maximized at $h_{aq} = c_{aq}$ and $h_{a\overline{q}} = c_{a\overline{q}}$, so if $(1-\mu)(s_a + c_{aq} + c_{a\overline{q}} + \gamma) - \gamma(s_{\overline{q}} + c_{a\overline{q}}) < 0$, no subgraph of $S \cup V$ containing v may be a (μ, γ)-community.

Theorem 5 *Let v be a vertex in S, and let s_a, $s_{\overline{q}}$, and c_{aq} be as in Theorem 4.*

If $\gamma \geq 1-\mu$ and $(1-\mu)(s_a + c_{aq} + \gamma) - \gamma s_{\overline{q}} < 0$, then neither S nor any supergraph of S will be a (μ, γ)-community.

Proof The proof for Theorem 5 is similar to the proof for Theorem 4, with the exception that $(1-\mu)(s_a + h_{aq} + h_{a\overline{q}} + \gamma) - \gamma(s_{\overline{q}} + h_{a\overline{q}})$ is maximized at $h_{aq} = c_{aq}$ and $h_{a\overline{q}} = 0$.

Lemma 2 *Let H be a (μ, γ)-community, let S be a subgraph of H, and let v be a vertex of S. Let s_q be the number of vertices of S in Q, $s_{\overline{a}}$ be the number of vertices in $S \setminus \{v\}$ not adjacent to v, c_{aq} be the number of vertices in $H \setminus S$ that are adjacent to v and in Q, and $c_{\overline{a}q}$ be the number of vertices in $H \setminus S$ that are in Q but not adjacent to v.*

$$(1-\gamma)(s_q + c_{aq} + c_{\overline{a}q}) - \mu(s_{\overline{a}} + c_{\overline{a}q} + 1 - \gamma) \geq 0$$

Proof This lemma follows from the inequalities $(1-\gamma)(|H|-1) \leq s_{\overline{a}} + c_{\overline{a}q}$ and $\mu|H| \leq s_q + c_{aq} + c_{\overline{a}q}$, similar to Lemma 1. □

Theorem 6 *If $\gamma < 1-\mu$ and $(1-\gamma)(s_q + c_{aq} + c_{\overline{a}q}) - \mu(s_{\overline{a}} + c_{\overline{a}q} + 1 - \gamma) < 0$, then neither S nor any supergraph of S will be a (μ, γ)-community.*

Theorem 7 *If $\gamma \geq 1-\mu$ and $(1-\gamma)(s_q + c_{aq}) - \mu(s_{\overline{a}} + 1 - \gamma) < 0$, then neither S nor any supergraph of S will be a (μ, γ)-community.*

Proof Similar to Theorems 4 and 5, Theorems 6 and 7 follow from maximizing the expression $(1-\gamma)(s_q + h_{aq} + h_{\overline{a}q}) - \mu(s_{\overline{a}} + c_{\overline{a}q} + 1 - \gamma)$ at $h_{aq} = c_{aq}$ and $h_{\overline{a}q} = c_{\overline{a}q}$ or 0, respectively. (Note that $\mu < 1-\gamma$ iff $\gamma < 1-\mu$.)

3 Method

Given the definitions and theorems, in this section we address the aforementioned technical challenges through some key innovative steps underlying the methodology. The methodology is summarized in Fig. 4.

3.1 Step 1: Abstracting the dynamic system

We first define the mathematical form for the dynamic system using climate spatio-temporal data as an example. Formally, let F be a set of variables (or factors) that characterize the system over spatial locations L over time period T. For example,

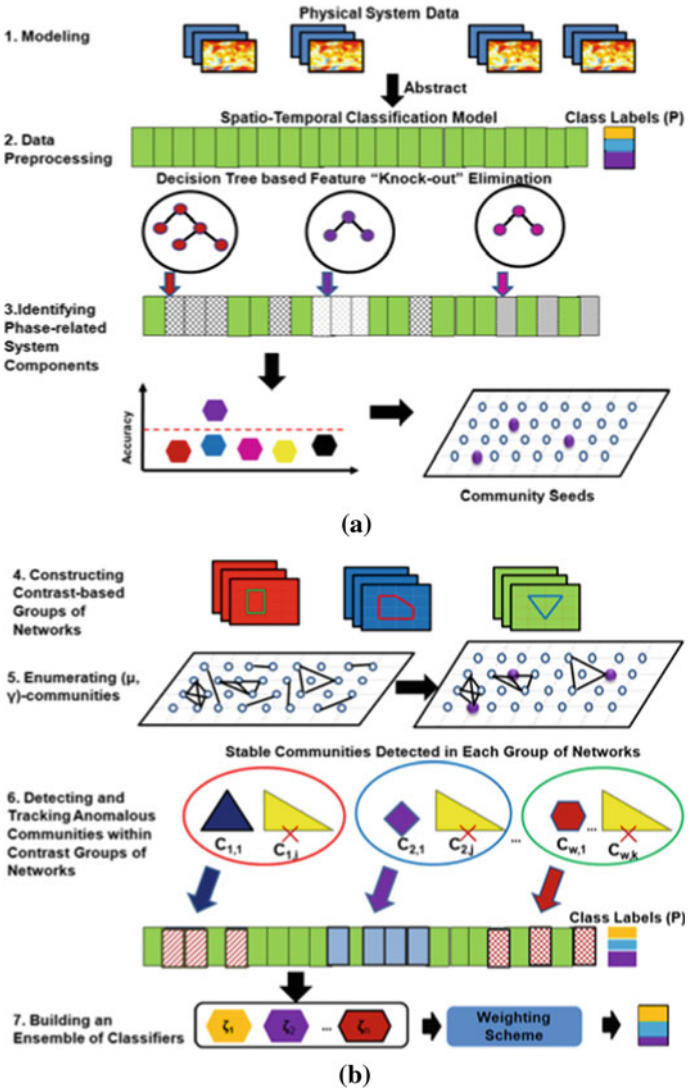

Fig. 4 The overview of our methodology. **a** Step 1–Step 3. **b** Step 4–Step 7

the climate system could be characterized by its climatological factors, such as Sea Surface Temperature (SST), Sea Level Pressure (SLP), and Vertical Wind Shear (VWS) defined over spatial (latitude, longitude, altitude) grid points over a time period of 1950–2010 with monthly mean values.

We divide T into disjoint segments $T_1, T_2, \ldots, T_m$ (say, calendar years), where each T_j can be further split into an *observable* time period $T_{j,o}$ and a *forecasting* time period $T_{j,f}$, according to time frame of the extreme event.

In the context of hurricane extreme events, for example, each time interval T_j may correspond to a calendar year that is further divided into a hurricane season $T_{j,f}$ = {July–November}, for which hurricane activity, say in the North Atlantic

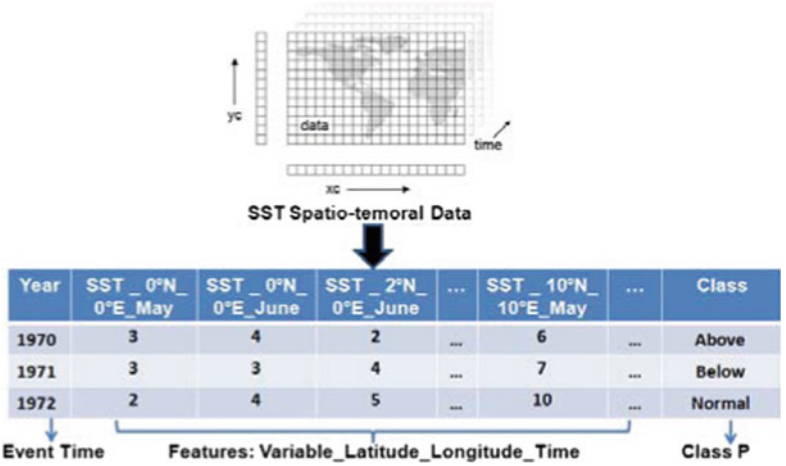

Fig. 5 Our proposed mathematical form for classification of spatio-temporal data

region, is being forecasted based on the observed or simulated monthly means for climatological factors defined over the entire globe L during the hurricane pre-season, $T_{j,o} = \{\text{November–June}\}$.

We consider the problem of classifying the climate system's state P over these time intervals according to some event-specific taxonomy. For example, according to paper (Chu et al. 2007), seasonal hurricane activity of Taiwan region could be broadly categorized as "above normal" (say, more than four hurricanes during the hurricane season), "normal," or "below normal" (say, less than three hurricanes in a season).

Based on the aforementioned notations, the mathematical form can then be defined as follows (Step **1**, Fig. 4). Let each row of the matrix correspond to each time interval T_j, $j \in \{1, 2, \ldots, m\}$, and let each column of the matrix correspond to a 3-tuple defined over $\mathcal{F} = F \times L \times T_{*,o}$, where $T_{*,o}$ is replaced with $T_{j,o}$ for the corresponding row T_j. Thus, each (row, col) cell of the matrix is filled in with the value of the corresponding variable in F for column col defined at the corresponding spatial point in L and the corresponding time $T_{row,o}$.

Furthermore, let us assume that a set of known extreme events E is defined over some spatial region L_e, and the class label from P is assigned to each time interval T_j based on the accumulative statistics of the observed events over $T_{j,f}$ time period in region L_e.

Figure 5 illustrates this mathematical abstraction using SST as variable, or predictand, defined over T = (1970–1972) during the months of $T_{*,o} = \{\text{May, June}\}$ over (latitude, longitude) spatial grid points for the sea-level altitude. The class label is inferred based on the historical record of observed hurricanes in North America during $T_{*,f} =$ (July–November) hurricane season.

3.2 Step 2: Data preprocessing

Given the aforementioned mathematical form of the original system data, the next step of our algorithm is data preprocessing designed to help us identify phase-related

community seeds in Step 3 (see Sect. 3.3). While the choice of which data preprocessing techniques to employ may be dependent on the type of data under consideration, for preprocessing spatio-temporal data, we use two techniques including *spatio-temporal deseasoning* and *discretization-based denoising*.

Spatio-temporal deseasoning If temporal data can exhibit seasonality, such as winter, spring, summer, and fall, each variable's time series at each spatial location is first transformed into the time series with zero mean and unit variance per season. This technique avoids learning a strong seasonality signal and also enables multiple variables with different scales of measurement to be combined into different columns of the same matrix.

Discretization-based denoising Dynamic system data like the climate data contains a lot of "noise" or irrelevant signals, so another important preprocessing step is to perform data cleaning or data denoising. We use a discretization method by Fayyad and Irani (1993) to filter out noise or irrelevant features in the data. This technique has been found to be effective in some domains like microarray analysis (Tan and Gilbert 2003), where non-discriminatory genes are filtered out before performing actual learning process on the gene expression data.

3.3 Step 3: Identifying phase-related system components

In this section, we aim to detect the phase-related system components or features, which can be used as seeds to generate (μ, γ)-communities in a network (see Step 5).

Given the mathematical classification form (Step 1) and the preprocessed spatio-temporal data (Step 2), we deploy decision tree based procedure for identifying the candidate system phase-related components or features.

There are multiple reasons for why we use a methodology based on decision trees for our feature space partitioning, including (a) their efficiency in processing many features (unlike Bayesian Belief Networks (BBNs), which have exponential complexity relative to the number of features), (b) support for multi-class data sets (unlike Support Vector Machines (SVMs), which are inherently binary classifiers), (c) the ability to handle continuous and multi-variate types of features (unlike Neural Networks (NNs), for which distance metrics are poorly defined for mixed data types), among others. We use the Classification and Regression Trees (CART)-decision tree algorithm (Breiman et al. 1984) to select a set of discriminatory features from the available feature space. Basically, CART builds a decision tree by choosing the locally best discriminatory feature at each split step based on the Gini Index Impurity Function. To avoid overfitting, CART employs backward pruning to build smaller, more general decision trees. CART chooses features in a multivariate fashion, which allows the feature selection process to find a set of discriminatory features instead of considering one feature at a time.

More importantly, especially in the context of underdetermined or unconstrained problems, CART's inherent feature pruning capability often leads to a smaller number of features. Also, decision boundaries themselves could result in rules that are more interpretable and could provide additional insights to domain scientists on how much the identified features affect the system's state. Not only is it important to know what

group of features contributes to the system's state, but also to what extent the feature values influence the system's state.

Algorithm 1: Phase-related component enumerator

Input:
$\mathcal{F}$: a set of features
D: a set of training data over $\mathcal{F}$
P: a set of system states over D
A: basic classification algorithms
: (e.g., decision tree, SVM, Naïve Bayes, etc.)
Output:
CIG: identified community seeds

```
1  CIG ← ∅;
2  while stopping criterion is not met do
       /*Run CART-decision tree to get a set of candidate features */
3      Run decision tree algorithm on D with feature set F to get a pruned decision tree M;
4      Let F_M be a set of all features that belong to the internal nodes of M;
5      D_{F_M} ← Extract the data from D only with the features in F_M;
6      Predictive skill score ε_M ← applying A to train D_{F_M};
7      if ε_M meets the training accuracy criterion then
8          Add F_M to CIG;
9      Remove features in F_M from F;
10 return CIG;
```

Specifically, we identify a candidate set of discriminatory features by building a decision tree model M using CART and extract the features that correspond to the internal nodes of M (Lines3–5 in Algorithm 1). The candidate system's features are then assessed in terms of their ability to contribute to the system's states. Basically, the goal is to define a scoring function that measures how well each group of features discriminates between system states. We define a scoring function in terms of classification accuracy (training accuracy in our experiments) provided by multivariate discriminant methods, such as SVMs, BBNs, neural networks, or decision trees. Specifically, we ask a question: if we used only the given set of candidate features to determine the system's state, how much predictive ability would this set have? Since individual features within the candidate group could be related to each other in a complex manner, we first let a proper classifier (e.g., kernel SVM or BBN) learn these complex relationships from the candidate features and predict the state of the system by using the candidate features only (see Lines 5–6 in Algorithm 1). If the training accuracy of the candidate feature set is above the threshold we set, the features are added to the community seed set.

The combinatorial nature of this task necessitates heuristic approaches. Our strategy is inspired by the way biologists often conduct their mutagenesis studies. Namely, they *knock-out* a group of genes (e.g., via gene deletion) and observe the *mutant* system's response. By analogy, our methodology *knocks-out* the selected candidate feature sets and proceeds in an *iterative* fashion until some *stopping criterion* is met (see Line 2 in Algorithm 1). Under this approach, each iteration produces a subset of features out of the current feature set (see Line 4 in Algorithm 1), then removes these features from

Year	Month	Day	(0°N, 0°E)	(0°N, 2°E)	...	(90° N, 180° E)
1970	1	1	3	6	...	7
1970	1	2	3	7	...	6
...	...	...	...	...	...	...
1970	12	31	12	10	...	22

Time Series

Fig. 6 A table-view of spatio-temporal data

the set so that they can't be selected again (see Line 9 in Algorithm 1). The maximum number of iterations is set as our stopping criterion. A set of phase-related features or components is output, when the stopping criterion is met.

3.4 Step 4: Constructing contrast-based groups of networks

There are several steps to construct climate networks, including constructing nodes of a network, calculating anomaly value, building edges of a network, and partitioning the networks into different groups.

The nodes (or oscillators Tsonis and Roebber 2004) of a climate network are identified with the physical locations or spatial grid points, which correspond to the time series of gridded climate data (see Fig. 6).

At each grid point, we calculate for each month $m = 1, \ldots, 12$ (i.e., separately for all Januaries, Februaries, etc.) the mean $\theta_m = \frac{1}{Y}\sum z_{m,y}$ and standard deviation $\sigma_m = \sqrt{\frac{1}{Y-1}\sum(z_{m,y} - \theta_m)^2}$, where y is the year, Y is the total number of years in the dataset, and $z_{m,y}$ is the value of series Z at month m and year y. Each data point is then transformed by using z-score transformation, that is each data point is ($\overline{z_{m,y}} = \frac{z_{m,y}-\theta_m}{\sigma_m}$) subtracted the mean and divided by the standard deviation of the corresponding month.

The edges between pairs of nodes exist depending on the degree of statistical interdependence between the corresponding pairs of time series taken from the climate data set. The Pearson correlation coefficient is chosen as a measure of link strength (Tsonis and Roebber 2004). For two series Z and X the correlation r is computed as $r(Z, X) = \frac{\sum(z_i-\bar{z})(x_i-\bar{x})}{\sqrt{\sum(z_i-\bar{z})^2\sum(x_i-\bar{x})^2}}$, where z_i is the ith value in Z and $\bar{z}$ is the mean of all values in the series. Note that the correlation coefficient has a range of $[-1, 1]$, where 1 denotes perfect agreement and -1 perfect disagreement, with values near 0 indicating no correlation. Since an inverse relationship is equally relevant in the present application, we set the correlation score to $|r|$, the absolute value of the correlation coefficient. Although nonlinear relationships are known to exist in climatological systems, the observed similarity of Pearson correlation still can be considered statistically significant, as concluded by Dongesv et al. (2009). Thus, we use Pearson correlation to measure the similarity between a pair of nodes in this work.

A correlation-based pruning is applied to the networks to prune the edges, that is only the pairs of nodes with the correlation scores above some threshold would be considered connected. To avoid the multiple comparison problem, the Monte Carlo

method is used. Specifically, for each network, we randomly sample N sets (say, $N = 1,000$) from the entire edge set of the tenth size as the original network, and compute the corresponding correlation threshold with p value $= 0.05$ from each sample set. The selected threshold for the target network is the one that meets 95 % confidence level within the threshold distribution for N samples.

Because the networks change over time, we build a network according to a calendar year. For example, for a time period over 1950–2009 with two climate variables (e.g., SLP and SST), up to 120 different networks can be built, with one network per year for each variable.

The complex networks of a dynamic system can be partitioned into different groups corresponding to different system's states (i.e., class P in Fig. 5). For example, in a tropical cyclone (TC) prediction system, we can build three different groups of climate networks, with one corresponding to strong TC years, one with normal TC years, and another with low TC years, based on the distribution of historical data. Different groups of networks may exhibit different properties of the community structure.

3.5 Step 5: Enumerating (μ, γ)-communities

We hypothesize that if the system feature or component is key to defining the system's state then its value distributions will be separable between the observations from different states. If the separation is strong, then such a feature, alone, is likely able to discriminate system states. And almost any method, like entropy-based, would likely succeed in detecting those features. However, with real data sets such a strong separation is less likely. There are different reasons for such an assumption. For example, the evolution of system behavior may induce non-functional changes to the system features. Thus, the effective analysis should not only include an individual feature with a strong discriminatory signal, but also extend to a group(s) of interplaying features out of a set of thousands of features. This creates a multiplicity of possible combinatorial interplays to search for and excludes a possibility for a brute-force enumeration.

In some cases, the domain knowledge may assist with constraining the search space of possible interplays. For example, climate index El Niño/La Niña–Southern Oscillation (ENSO) has been found to be highly correlated with hurricane activities (Camargo et al. 2010). For a more general and domain-independent solution, however, the issue of properly constraining the search space still remains.

Standard algorithms would attempt to find all dense subgraphs throughout the networks. However, in real-world dynamic system data, there are a lot of irrelevant features or "noises." Including all features including the "noises" to generate the dense subgraphs would retrieve a huge number of results irrelevant to the system phases or states. We hope to reduce the problems of high algorithmic complexity and the number of irrelevant results by integrating the system phase-related components or features into the search in the form of a "seed set" of vertices. For example, given a phenotype-expressing organism, a biologist might have known a set of proteins that are related to the target phenotype. By using those proteins as the "seed" set, we can identify all the dense functional modules in a biological network that contain some part of the "seed" vertices.

Specifically, given the set of phase-related system components as seeds (Step 3) and a constructed network (Step 4), the basic premise of our method is that we will build the (μ, γ)-communities one vertex at a time, starting with a single query vertex v_0 and backtracking as we find maximal (μ, γ)-communities or subgraphs that cannot be contained in a (μ, γ)-community. For this section, we continue the convention that S represents the current subgraph under consideration, and C represents the set of vertices that could extend S to produce a (μ, γ)-community. A pseudocode outline of the algorithm appears in Algorithms 2 and 3.

Algorithm 2: (μ, γ)-community generation algorithm, continued in Algorithm 3

Input:
G: a given graph
μ : an enrichment parameter
γ : a density parameter
Q: a "seed" vertex set
Output:
C: Communities or dense enriched subgraphs in graph G

1 **foreach** $v_0 \in Q$ **do**
2 $S \leftarrow \{v_0\}$;
3 $V \leftarrow N^2(v_0)$;
/*$d_v = s_a + c_a - \gamma(|S| + c_a)$: condition of Theorem 2 */
4 Calculate d_v for all $v \in S \cup V$;
/*$\varepsilon = \mu|S| - (1-\mu)c_q$: condition of Theorem 3 */
5 Calculate ε ;
/*$g_v = (1-\mu)(s_a + c_{aq} + c_{a\overline{q}} + \gamma) - \gamma(s_{\overline{q}} + c_{a\overline{q}})$: condition of Theorem 6 and 7 */
6 ;
7 Calculate g_v for all $v \in S \cup V$;
/*$m_v = (1-\gamma)(s_q + c_{aq} + c_{\overline{a}q}) - \mu(s_{\overline{a}} + c_{\overline{a}q} + 1 - \gamma)$: condition of Theorem 6 and 7 */
8 ;
9 Calculate m_v for all $v \in S \cup V$;
10 Remove all unpromising vertices of V ;
11 **if** $S \cup V$ *is maximal* **then**
12 **Call** `Enumerate()`;
13

From Theorem 1, we can see that the vertices at most distance 2 from every vertex of S can serve as an appropriate starting point for our set V. However, rather than recalculating this intersection of sets every time a vertex is added to the set S, we first define V as the set of all vertices within distance 2 of the initial vertex, $N^2(v_0)$, (in Line 3 of Algorithm 2) and intersect V with $N^2(v)$ for each vertex v we add to S as part of line 12 of Algorithm 3. As these $N^2(v_0)$ sets can be precomputed and stored in a matrix, this technique should make for a much more efficient way to apply Theorem 1.

By Theorem 2, we know that for any vertex $v \in V$, if s_a represents the number of vertices of S adjacent to v, c_a represents the number of vertices of V adjacent to v, and $s_a + c_a \leq \gamma(|S| + c_a)$, then neither $S \cup \{v\}$ nor any supergraph can be a (μ, γ)-community. Rather than recomputing this inequality every time we add or remove a vertex from S, we calculate and maintain the value of $s_a + c_a - \gamma(|S| + c_a)$ as d_v (lines 4

Algorithm 3: `Enumerate` function

Input:
G: a given network or graph
μ: an enrichment parameter
γ: a density parameter
V: the set of all vertices within distance 2 of the initial vertex
S: the current subgraph under consideration

Output:
Communities that contain initial vetex in graph G

```
T ← V;
while some vertices of V are marked do
    Remove all marked vertices from V;
    if S violates one of the theoretical constraints then
        Restore all vertices of T \ V to V;
        return;
    Update ε and all d_v, g_v, and m_v values as appropriate;
    if S ∪ V is nonmaximal then
        Backtrack until some vertex of V is restored;

while V ≠ ∅ do
    Choose v in V according to some heuristic and move v to S ;
    Update ε and all d_v, g_v, and m_v values as appropriate;
    if g_v < 0 or m_v < 0 for some v ∈ S then
        Restore vertices of T \ V to V;
        Update ε, d_v, g_v, and m_v values appropriately;
        return ;
    Mark all vertices of V to be removed;
    if S does not violate any of the theoretical constraints then
        Call Enumerate();

    Remove v from S;
    Update ε and all d_v, g_v, and m_v values as appropriate ;
    if S violates one of the theoretical constraints then
        Restore vertices of T \ V to V;
        Update ε and all d_v, g_v, and m_v values ;
        return ;
    Iteratively remove unpromising vertices of V;
    Update ε and all d_v, g_v, and m_v values as appropriate ;
    if S ∪ V is nonmaximal then
        Backtrack until some vertex of V is restored;

if no recursive call of Enumerate() found a (μ, γ)-community then
    Output S;
    Update the maximality index for each vertex in S;
Restore vertices of T \ V to V;
Update ε, d_v, g_v, and m_v values appropriately ;
return ;
```

of Algorithm 2 and lines 7, 13, 16, 22, 25, 28, and 36 of Algorithm 3), reducing the value when we remove an adjacent vertex from the candidate set or when we add a nonadjacent candidate to the current subgraph. When this d_v value becomes zero or negative, v may be removed from V. We also maintain a d_v value for each $v \in S$, as we know from Theorem 2 that $s_a + c_a \leq \gamma(|S| - 1 + c_a)$, where s_a and c_a are defined as before. When the value of d_u becomes negative for a vertex $u \in V$, we can remove u from V by the result of Theorem 2. Additionally, when d_v decreases below γ for a vertex $v \in S$, we can remove all vertices of V that are nonadjacent to v, as adding such vertices to S would violate Theorem 2.

In a similar fashion, we calculate the initial values for ε and each g_v and m_v value (see Lines 4 to 7 in Algorithm 2 for definitions) and update these values as the algorithm progresses. We can then remove vertices from V whose addition to V would violate Theorems 3, 4, 5, 6, or 7.

In order to decide when a (μ, γ)-community is maximal, we propose maintaining a bitmap index of the (μ, γ)-communities that contain each vertex. As the algorithm identifies (μ, γ)-communities, it assigns numbers to them sequentially and adds these values to the indices for the vertices contained in the (μ, γ)-communities. Then, as we add and remove vertices from set V, we check to see if there is an already-discovered (μ, γ)-community that contains all vertices of $S \cup V$ by performing a bitwise *and* of the associated indices. If there is an already-discovered (μ, γ)-community that is a superset of $S \cup V$, we may safely backtrack, as no further extensions of S will be maximal.

We use a hierarchical bitmap index rather than a prefix tree because we need to be able to detect subsets of a set, as well as equality. While prefix trees are very good at detecting equality or initial substrings, being able to recognize an arbitrary subset, such as the subset {b, c, e} of the set {a, b, c, d, e}, would require a very dense prefix tree. In terms of bitmaps, to recognize {b, c, e} as a subset of {a, b, c, d, e}, we apply a binary "and" to all of the bitmaps associated with vertices b, c, and e to see if a previous clique contains all three of these vertices simultaneously, and if {a, b, c, d, e} were the $n^t h$ clique discovered, the $n^t h$ bit of all five bitmaps would be set.

3.6 Step 6: Detecting and tracking anomalous communities in contrasting groups of networks

The anomalous communities in the contrasting groups of networks are more "biased" towards the target system phases than the communities in a single network, or conserved (or stable) communities in a group of time-varying networks. Thus, in this section, our goal is to extract only the anomalous communities from all communities generated from different groups of networks.

Based on the Definition 6, in order to identify anomalous communities, we first need to detect all (α, β)-conserved communities in each group of networks, where $1 \leq j \leq i, \alpha \in [0.5, 1]$ and $\beta \in [0.5, 1]$. A stable community should have at least one α-corresponding community in majority of the networks of the same group. That is the size of overlapping parts between the stable community and its "strict" corresponding community should be larger than half (at a minimum) the size of any of them.

Algorithm 4: Anomalous community detection algorithm, continued in Algorithm 5

```
   Input: C: All communities generated from all graphs
   in contrasting groups {U_1, U_2, ..., U_τ}
   β, ω, α: Parameters
   Output: χ: A set of anomalous communities
   /*Detecting stable communities in each group of networks                    */
1  for i = 1 : τ do
2      anomaly_indicator = 0;
3      SC_i = Call Detecting();
   /*Using the τ sets of SC as inputs for detecting anomalous communities      */
4  anomaly_indicator = 1;
5  α = ω ;
6  χ = Call Detecting();
```

Algorithm 5 summarizes the aforementioned stable community detection procedure. After detecting stable communities from all groups of networks, each stable community is examined to see if it has any "looser" corresponding community (with minimum intersection factor ω, where $\omega \in (0, \alpha]$) in the set of stable communities of all the other groups. Only those communities that do not have any "looser" corresponding community will be considered as anomalous communities.

The anomalous community detection between the different sets of stable communities (with each set generated from each group of networks) only requires a little change with regard to the input variables (see Lines 4 to 6 in Algorithm 4) and the output process (see Lines 16 to 17 in Algorithm 5).

3.7 Step 7: Building an ensemble of classifiers from anomalous communities

While the enumerated set of anomalous communities is important in its own right (as illustrated in Sect. 4), here we combine them altogether by building an ensemble of classifier models.

For each of the anomalous communities χ identified, we specifically distinguish between treating it as a *binary* feature (i.e., the community is present or absent in a graph) or *continuous* features, that is we form a new data set D_χ by restricting the original data to include only the features (or spatial grid points) F_χ in χ. We then train a separate base classification algorithm A (e.g., decision tree, SVM, Naïve Bayes, etc.) on the binary data set or the restricted data set to construct a candidate classifier model ζ. The candidate classifier model ζ will only be included into the ensemble of classifiers if it meets the *model selection criterion*. The resulting class prediction for the event with the unknown class label is based on the majority voting of the selected classifiers ζ's.

Some of the key characteristics for building a robust classifier ensemble include (a) the diversity among the classifier models in the ensemble and (b) the reasonably high accuracy of the individual members in the ensemble. In our case, the former is ensured due to our feature set knock-out strategy (Step 4) and the latter is guaranteed by a combination of the scoring function (Step 2) and the statistical significance assessment (Step 3) that, in combination, also reduce possible redundancy among the

Algorithm 5: `Detecting` function

Input: α, β: Parameters for conserved community
C: All communities in k graphs,
or stable communities from k groups
anomaly_indicator: An indicator for anomalous community detection
Output:
η: A set of detected communities

```
Initialize count;
for snapshot s = 1 : (k − 1) do
    for snapshot n = s + 1 : k do
        indicator =0;
        for each community C_{s,i} in G_s do
            for each community C_{n,j} in G_n do
                overlap_part = |C_{s,i} ⋂ C_{n,j}|;
                if overlap_part/|C_{s,i} ⋃ C_{n,j}| > α then
                    count_{s,i} = count_{s,i} + 1;
                    if indicator=0 then
                        count_{n,j} = count_{n,j} + 1;
                        indicator =1;

                    break;
if anomaly_indicator = 0 then

[I J]=find(count > β ∗ k);
else
    [I J]=find(count < β ∗ k);
Add C_{I,J} to η for each pair of I and J at the same row;
Delete duplicate communities in η;
Output η;
```

models and thus reduce the possible bias (e.g., due to a significantly large portion of highly similar models).

Finally, in the last step (Step 7 in Fig. 4), we need to combine the predictions of all the classifiers that pass statistical significance criterion (Step 3) to come up with the final prediction value. In order for the ensemble to make a prediction, each classifier is given a weighted vote, and the class with the most votes is the prediction of the ensemble. We tested three possible weighting schemes (Tao et al. 2006): a simple majority voting scheme, in which every classifier is given equal weight; a training error-based method, in which every classifier is weighted based on its training error; and a confidence-based method, in which each classifier is weighted by that model's associated confidence value. Due to space limitations, we present results for a simple case, majority voting.

4 Experimental results

The nature of the proposed methodology suggests that detected anomalous communities from contrasting groups of networks (Steps 1–7) (1) could play an important role

in defining the system's state(s) and (2) collectively, could improve the predictive skill of the system's states (Step 7). We also demonstrate the efficiency of our algorithm by applying it to the synthetic datasets.

4.1 Data and tasks

Two real-world extreme event prediction tasks are considered in this paper:

1. Seasonal tropical cyclone prediction: The first task is to predict the seasonal tropical cyclone (TC) count in some spatial region (Goldenberg and Shapiro 1996; Kim and Webster 2010). TCs, especially hurricanes, have become a serious issue of our era because they result in enormous loss of life and property.
2. African Sahel rainfall prediction: The second task is to predict the seasonal rainfall in North Africa, especially, in the Sahel area (Yeshanew and Jury 2007). Rainfall in this area is highly related to meningitis epidemics that affects more than 200,000 people throughout the region annually.

We use the North Atlantic tropical cyclone (TC) count series from 1950 to 2009 from the seasonal (July through November) Atlantic hurricane database (HURDAT) at the National Climatic Data Center to form the class labels. We also utilize the North Pacific seasonal (June through October) TC count series from 1970 to 2006 provided by the Central Weather Bureau (Chu et al. 2007). Monthly rainfall data is obtained from the Climate Research Unit at a $0.5° \times 0.5°$ latitude and longitude resolution for the period of 1950–1998. East Sahel rainfall indices are obtained by averaging seasonal (July through September) mean precipitation data over (10–20°N, 15–30°E).

The monthly mean sea level pressure (SLP), precipitable water (PW), sea surface temperature (SST), and tropospheric vertical wind shear (VWS) data are used for the North Atlantic TC, North Pacific TC and Sahel rainfall class prediction. SLP and PW are NCEP/NCAR reanalysis datasets. They are available at a $2.5° \times 2.5°$ latitude and longitude resolution. SST is from the NOAA Climate Diagnostic Center in Boulder, Colorado, at a resolution of $2° \times 2°$ latitude and longitude. VWS is calculated by computing the square root of the sum of the square of the difference in zonal wind component between 850 and 200 hPa levels and the square of the difference in meridional wind component between 850 and 200 hPa levels (Clark and Chu 2002) from NCEP/NCAR reanalysis data.

The observed extreme event count series of the target system are classified into three classes: below normal, normal, and above normal, with a distribution of 40 % as normal and 30 % each as below normal and above normal. For instance, in the case of Taiwan region TC prediction (Chu et al. 2007), years with fewer than three seasonal TCs are classified as below normal, and years with at least five TCs are classified as above normal.

We use parameters $\gamma = 0.75$ and $\mu = 0.001$, which correspond to searching for dense but not necessarily complete subgraphs as communities that contain at least one of system phase-related components. We use parameters $\alpha = 0.6$, $\omega = 0.4$, and $\beta = 0.6$ for defining the anomalous communities.

Table 1 Identified climate indices related to hurricane activities

Community ID	Variable	Spatial location	Climate indices
1	SST	(4°N, 114°W)	Niño 3
		(2°S, 168°W)	ENSO
		(42°N, 30°W)	
		(32°S,16°W)	
2	VWS	(27.5°N, 65°W)	MDR
		(52.5°N, 37.5°W)	NAO
		(7.5°N, 122.5°W)	Niño 3
		(10°S, 60°W)	
		(27.5°N, 55°E)	
3	PW	(52.5°N,135°E)	PDO
		(82.5°N, 15°W)	AO
		(37.5°N, 40°E)	
4	SLP	(57.5°N, 22.5°W)	NAO
		(60°N, 155°E)	PDO
		(37.5°N, 162.5°W)	
		(12.5°N, 122.5°E)	

4.2 State determining communities

Climate indices associated with hurricane activities:

Table 1 shows four different anomalous communities, representing functionally associated or synchronized groups of oscillators (or spatial grid points), detected by our algorithm for North Atlantic tropical cyclone prediction. In each community, our algorithm is able to identify at least one oscillator corresponding to a known climate index related to tropical cyclone activity. For example, for the first anomalous community detected from the SST networks, we can see that one oscillator is located in the Niño 3 region. Niño 3 SST has a strong correlation with Atlantic hurricane activity (Goldenberg and Shapiro 1996; Kim and Webster 2010). Another oscillator belongs to the El Niño/La Niña-Southern Oscillation (ENSO) region, which has been found to modulate the tropical systems and strongly influences North Atlantic tropical cyclones (Camargo et al. 2010).

The second anomalous community identified oscillators in the hurricane main development region (MDR) and North Atlantic Oscillation (NAO). The MDR index has been shown to contribute to the hurricanes generated in the MDR region (Saunders and Harris 1997; Xie et al. 2005). And the NAO index, especially the June NAO, has been found to be correlated with North Atlantic hurricane tracks of the incoming hurricane season (Elsner 2001; Xie et al. 2005). The Pacific Decadal Oscillation (PDO) index was identified in our third community. Shifts in the PDO phase can have significant implications for Atlantic hurricane activity, and significant differences are shown in hurricane intensity between El Niño and La Niño years when the PDO is in the warm phase (Magill et al. 2008). The PDO index is also identified in the fourth anomalous community. Our algorithm also finds some other anomalous communities, which

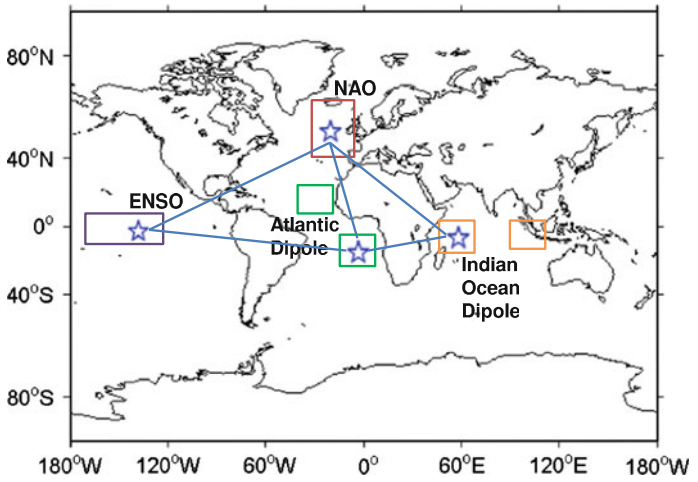

Fig. 7 One anomalous community detected for African Sahel rainfall prediction

correspond to other climate indices like Atlantic multidecadal Oscillation (AMO) and Artic Oscillation (AO) that might affect the North Atlantic tropical cyclone activities too, though this has not been reported in the literature. There are other 342 anomalous communities detected by our algorithm.

African Sahel rainfall-related teleconnection patterns For the African Sahel region rainfall prediction case, our algorithm also detected some anomalous communities with one shown in Fig. 7.

Climate variability in the tropical Atlantic involves complex but interacting processes that actively or passively exert their influences on rainfall and relative humidity variability over West Africa (Sutton et al. 2000). Moisture supply over West Africa primarily emanates from the eastern equatorial and South Atlantic, determined from the strength of the meridional and the zonal modes. However, other teleconnection patterns such as ENSO, NAO, and Indian Ocean dipole are competitively engaged to dictate the rainfall and relative humidity variability at different scales. The equatorward extension of the extratropical NAO pattern influences the West African climate by weakening the northeasterly trades, whose presence is a prerequisite to the formation of large-scale convergence over the continent to reinforce convective development. NAO also influences the region's climate through a modification of the northern lobe of the meridional mode. Thus, the detected anomaly community shown in Fig. 7 appears to support the hypothesis proposed by our climate scientists (our co-authors), which is being further investigated, that the NAO modulates meridional moisture transport over the tropical Atlantic, mediated mainly through the zonal equatorial trades.

4.3 Predictive skill of system's states

Performance evaluation method Because of the small sample size of the spatio-temporal data, leave-one-out cross validation (LOOCV) is employed to evaluate the

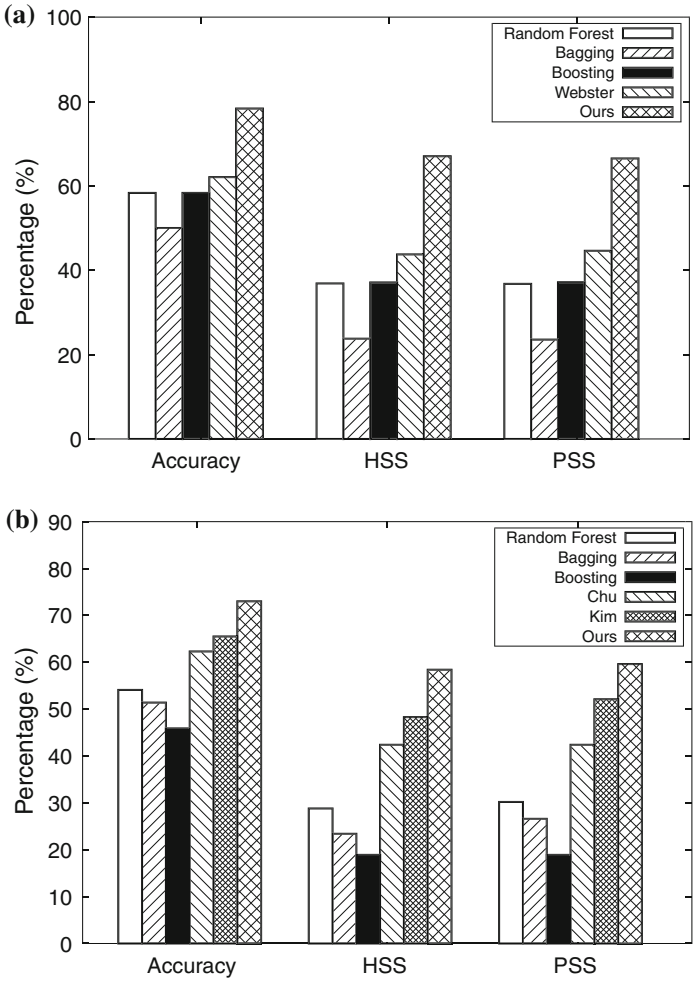

Fig. 8 LOOCV performance for seasonal TC prediction. **a** North Atlantic TC prediction. **b** North Pacific TC prediction

robustness of our methodology. We utilize several metrics to evaluate the performances: accuracy, Heidke Skill Score (HSS) (Jolliffe and Stephenson 2003), and Peirce Skill Score (PSS) (Jolliffe and Stephenson 2003). Accuracy is defined as the ratio of the number of correctly classified data points to the total number of data points in the test set. The HSS measures how well a forecast performs compared to a randomly selected forecast (Jolliffe and Stephenson 2003). And PSS, also called "true skill statistic," is another popular skill score computed by the difference between the hit rate and the false alarm rate (Jolliffe and Stephenson 2003).

Performance comparison: Figure 8 compares our algorithm performance to seasonal tropical cyclone predictions by Chu et al. (2007), Kim et al. (2010), and Kim and Webster (2010), and three benchmark ensemble classification methods: random

Table 2 Different modules' contributions on performance

Metric	OC	OB	BF	AC	RFC
Accuracy	0.82	0.8	0.72	0.75	0.73
HSS	0.72	0.69	0.58	0.60	0.58
PSS	0.72	0.68	0.59	0.62	0.60
GSS	0.71	0.68	0.55	0.63	0.60

forest, bagging, and boosting. The same basic classifier—CART decision tree, and the same data including four variables (SST, SLP, VWS, and PW) with all features are used for all methods. For the North Pacific region, there is a roughly 8 % increase over the 65.5 % reported by Kim et al. (2010). For the North Atlantic region, our method achieves an increase of at least 16 % in accuracy and 20 % in HSS and PSS over the four benchmark methods.

To estimate the contributions of each module in our algorithm to the performance improvement, we implemented different versions of our algorithm: the *original-continuous* version (OC) includes all the algorithm modules by using the *continuous* community features (see Sect. 3.7); the *original-binary* version (OB) also includes all the algorithm modules but uses the *binary* community features; the *brute-force* (BF) version uses all original features without detecting the anomalous communities, but it builds the classifiers by using our ensemble method (see Step 7 in Fig. 4); the *all-community* (AC) version enumerates all γ-dense communities without using the phase-related components as the query set (see Step 3 in Fig. 4), while keeping the other steps in the *original-continuous* algorithm unchanged; and the *random forest with anomalous community detection* (RFC) version changes only one step in the *original-continuous* algorithm by using the random forest instead of our ensemble method to build the ensemble of classifiers. Among those, *AC* is the most time-consuming version because it generates all possible γ-dense communities without using any query vertex. Irrelevant communities containing all "noises" can be generated as well, which would affect the prediction performance.

Table 2 compares the performances of different versions on seasonal North Atlantic tropical cyclone prediction. The *original-continuous* version outperforms the *original-binary* version by 2 % using the *binary* community features. The accuracy decreases by 10 % if we did not use the anomalous communities as the features, and decreases by 7 % if we used γ-dense communities instead of (γ, μ)-communities. And our ensemble method outperforms the random forest method by 9 % using the same selected anomalous community features.

4.4 Efficiency test on synthetic data

In our methodology, the most time-consuming step is (μ, γ)-community generation (see Sect. 3.5). Thus, in this section, we present the experimental results to demonstrate the efficiency of our community generation algorithm in large, scale-free graphs like climate networks.

Table 3 Graph size and number of communities generated using R-MAT

$\|V(G)\|$	$\|E(G)\|$	Communities		
		Clique	Enriched	Dense
27	889	569	23	14
255	1785	1199	64	21
510	3570	2593	104	72
1022	7154	5563	270	257
2039	14273	11831	485	432
4079	28553	24930	943	659
8132	56924	52025	1915	1774
16285	113995	106973	3991	4031

Table 4 Parameter settings for synthetic experiments

Description	γ	μ	$\|Q\|$
Clique	0.999	0.001	$\|V(G)\|$
Enriched	0.5	0.90	$\|V(G)\|/10$
Dense	0.85	0.85	$\|V(G)\|/6$

We used the R-MAT random graph generator (Chakrabarti et al. 2004) to generate those graphs of increasing size. The graphs were generated to have vertices equal to a power of two, with an average vertex degree of 14 ($|E(G)| = 7|V(G)|$). The graphs were then processed to remove isolated vertices, which do not contribute to our search for dense, enriched communities. All graphs were generated using the default R-MAT parameters of $a = 0.45, b = 0.15, c = 0.15$, and $d = 0.25$. More details on the generated graphs can be found in Table 3.

For the synthetic experiments, we ran our algorithm three times in order to detect three different types of (μ, γ)-communities: high density, low enrichment ("clique") communities where Q contains every vertex of the graph; high enrichment, low density ("enriched") communities with a small query set (every 10th vertex of $V(G)$); and moderate enrichment and density ("dense") communities with a medium-sized query set (every 6th vertex of $V(G)$). These settings were chosen to test the algorithm (and various candidate vertex constraints) under a wide variety of conditions. The parameter settings for these three types of communities appear Table 4.

For our implementation, we select the candidate vertex to add to the community using a naïve heuristic: the candidate that appears first in the array is chosen. We tested our algorithm on the R-MAT graphs described in Table 3 using all three of the parameter settings in Table 4, and we calculated the rate at which the (μ, γ)-communities were produced. The results appear in Fig. 9.

From Fig. 9, we can see that the "clique" communities were generated much more quickly than the "dense" or "enriched" communities, likely due to the extremity of the density requirement for the "clique" communities, which ensures that the resulting communities are fully connected. Also notable is that the time required per community

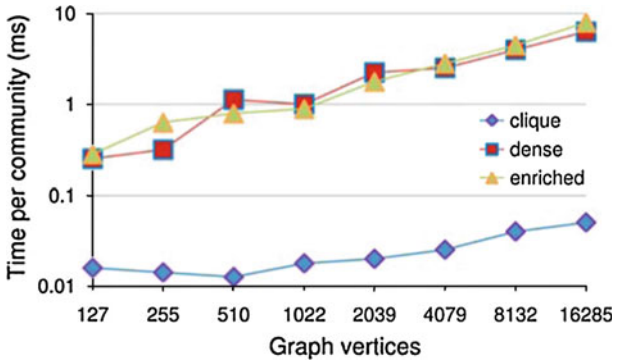

Fig. 9 Timing results for (μ, γ)-community enumeration algorithm

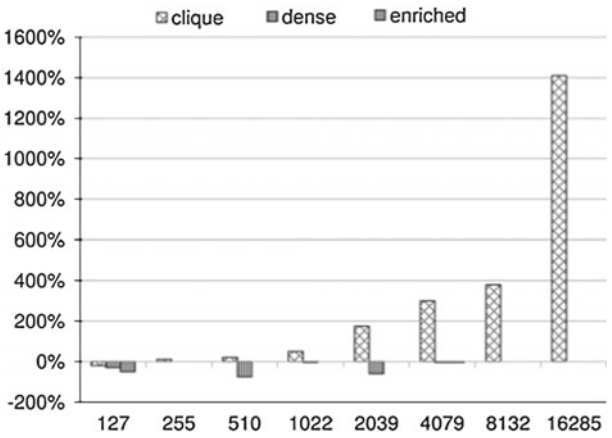

Fig. 10 Speedup results for using hierarchical bitmap index in (μ, γ)-community enumeration algorithm

appears to increase linearly on the log–log plot, implying that the time per community increases polynomially with the size of the graph.

Moreover, the time required for these problems increased sublinearly, rising by a factor of less than 30 in all cases, despite the problem size increasing by a factor of 128. While this scaling is obviously dependent on the graphs being analyzed, this result does suggest that our algorithm would be able to efficiently calculate dense and enriched communities on large, sparse graphs with a power-law degree distribution.

As a second experiment, we wished to evaluate the effectiveness of using the hierarchical bitmap index described in Sect. 3.5. For the purposes of this test, we implemented a second version of the algorithm that used only a flat (non-hierarchical) bitmap index, and we compared the time per community for both implementations.

From Fig. 10, we can see that as the size of the graph increases, the hierarchical bitmap index provides a significant speedup in the rate of identifying "clique" communities. When calculating "dense" and "enriched" communities, the flat index offers a moderate improvement over the hierarchical index (as much as 53 %), though this advantage disappears on graphs larger than 2,048 vertices. These results are likely due

to the fact that the graphs in question have significantly more "clique" communities than "dense" or "enriched" communities—as the size of the index grows, so does the potential advantage in using a hierarchical index. As such, we conclude that the hierarchical index is successful at improving the algorithmic runtime as the size of the index grows.

5 Discussion

5.1 Parameter selection

Our algorithm requires five parameters: the enrichment (μ) and the density (γ) for defining the communities, and parameters ω, α, and β for defining the anomalous communities. The description of these parameters (in Sect. 2) suggests that higher values of γ will produce more connected (clique-like) subgraphs. Similarly, higher values of the enrichment ($\mu \geq 0.5$) will produce subgraphs that are primarily composed of the "query" vertices, whereas a very low value ($\mu \leq 0.001$) will result in enumeration of all the subgraphs that satisfy the γ threshold and contain at least one query vertex. And higher values of α and β will produce fewer conserved communities in each group of networks, whereas higher ω will result in more anomalous communities.

Parameter thresholds depend on the application. In this paper, we are interested in identifying phase-biased communities in contrasting groups of climate networks, given a set of extreme event-related climatological oscillators as a "seed" set. Setting μ value to 0.001 will result in finding all the communities containing at least one "seed" vertex that could potentially be related to the spatio-temporal extreme events. Since climate networks are prone to missing information (edges), the value of $\gamma = 1$ could be too stringent, and the algorithm may miss some of the extreme event-related communities. Hence, we chose a γ value of 0.75 (midpoint of 0.5 and 1) to identify highly connected (but not fully connected) subgraphs as most probable communities that are teleconnected (i.e., edges linking geographically distant nodes) with extreme event-related "seed" oscillators. And due to the dynamics of climatological systems, we set the value of $\alpha = 0.6$ and $\beta = 0.6$ to find all possible but highly phase-related conserved communities in each group of networks. Finally, a relatively small value of $\omega = 0.4$ (smaller than α) is chosen to make sure that the anomalous communities are only conserved within one group of networks, not in the other groups of networks.

Figure 11 shows the sensitivity analysis results of the five parameters on North Atlantic TC prediction. The default values for the five parameters are: $\mu \leq 0.001$, $\gamma = 0.75, \alpha = \beta = 0.6$, and $\omega = 0.4$. We only change the value of one parameter at a time to test the sensitivity. The results shown in Fig. 11 agree with the aforementioned parameter analysis.

5.2 Generalization: detecting biologically relevant functional modules through biological networks

Thus far, we have presented how to detect phase-biased communities from climate networks. But our algorithm can be applied to other domains as well. Here, we provide

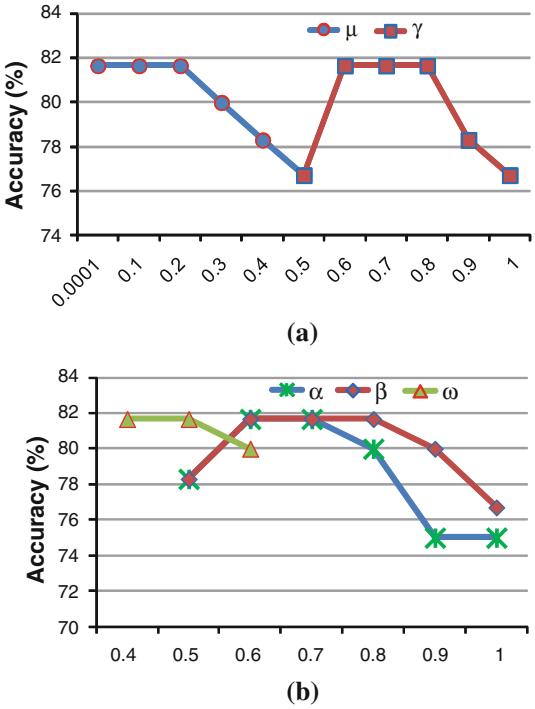

Fig. 11 Sensitivity analysis for seasonal North Atlantic TC prediction

a general idea on how our algorithm can be used to detect functional modules through biological networks.

The biological networks like gene functional association networks can be obtained from the STRING database (Jensen et al. 2009). The nodes in the networks are genes. And a pair of nodes is connected with an edge if the corresponding genes are considered to be functionally associated by some evidence. The edge weights are assigned by the STRING database based on the evidence that support the functional association (Jensen et al. 2009).

For a set of networks corresponding to phenotype-expressing organisms, we hypothesize that the conserved α-corresponding communities across the group of networks are the phenotype-associated functional modules. After generating all communities from each biological network, we first detect the α-corresponding communities across two networks, and then check if the α-corresponding communities detected in the previous two networks are conserved in the third network. This procedure is continued until all networks in the group are examined.

We can take it one step further and use a group of contrast biological networks (i.e., networks of organisms that do not express the phenotype) to filter and obtain communities that are not only identified as conserved in the previous step but are also "biased" towards the target phenotype. Here, by biased, we mean occurring in phenotype expressing organisms but not occurring in the phenotype non-expressing organisms. To achieve this goal, first, the networks are partitioned into different groups according to the phenotype(s), and then the conserved community detection algorithm

Table 5 Dipole detection results

Dipole	Modularity	Our method
North Atlantic Oscillation (NAO)		Found
Southern Oscillation Index (SOI)		Found
Pacific/North American Index (PNA)		Found
Arctic Oscillation (AO)	Found	Found
Western Pacific (WP)		Found

is applied to each group of networks. After getting all the conserved communities from all groups, we remove all the common conserved communities appearing in at least two groups of networks. The remaining anomalous communities are the phenotype-associate functional modules, which can be used to improve the predictive skill of the system's phenotypes.

5.3 Comparison to the modularity-based community detection

Since there is no existing algorithm that is specifically designed for solving our problem (see Problem 1), here we only compare the community detection module in our algorithm (see Algorithm 2) with the modularity-based approach (Clauset et al. 2004). Both algorithms are applied on the SLP network of the year 1950. The known pressure dipoles shown in paper (Kawale et al. 2011) were used as a validation set. Dipoles are one class of teleconnection phenomena that are characterized by recurring patterns of climate anomalies related to each other at long distances. Such teleconnections are important for understanding and interpreting climate variabilities.

Table 5 shows the dipole detection results by the modularity-based method and our (μ, γ)-community generation algorithm. Only if the opposite polarities of a dipole appearing at two different locations were both detected in a single community, the dipole was marked as "found" by the algorithm. Among the five known dipoles, only AO dipole was found by the modularity-based method, while all five dipoles were found by our algorithm. Thus, although modularity-based method might work better for some application domains like social networks, we may lose important teleconnection information by using the modularity-based community definition. Also, our algorithm detected many overlapping communities, which are not shown in the table, while the modularity-based method could only generate the non-overlapping communities. As mentioned earlier, climate communities (or biological functional modules) often work in a cross–talking manner. Ignoring the correlation and interaction between communities is not a good modeling for some complex systems like climatological ocean-atmosphere system.

Another advantage of our (μ, γ)-community generation algorithm is that a set of query nodes can be directly incorporated into the community search to improve the complexity and the quality of the results. For example, a climatologist might wish to search an El Niño or La Niña climate network for those communities associated with El Niño or La Niña events using some of his/her known climate indices as "prior knowledge."

6 Conclusions

In this paper, we introduced the important and challenging problem of detecting predictive and phase-biased communities in contrasting groups of networks. We presented an efficient and effective method that partitions physical system networks into different groups according to the system's phases, discovers phase-related system components, and uses these components as seeds to identify the phase-biased communities across different groups. Our method successfully identified climate indices associated with hurricane activities and found teleconnection patterns related to rainfall in the Africa Sahel region. Our method also improved the predictive skill of the system's state by 8–16 % relative to state-of-the-art approaches and other ensemble methods, such as bagging, boosting, and random forest.

Acknowledgments The authors would like to thank the editor and the anonymous reviewers for their valuable comments and suggestions to improve the paper. This work was supported in part by the U.S. Department of Energy, Office of Science, the Office of Advanced Scientific Computing Research (ASCR) and the Office of Biological and Environmental Research (BER) and the U.S. National Science Foundation (Expeditions in Computing). Oak Ridge National Laboratory is managed by UT-Battelle for the LLC U.S. D.O.E. under contract no. DEAC05-00OR22725.

References

Balasundaram B, Butenko S, Hicks IV (2011) Clique relaxations in social network analysis: the maximum *k*-plex problem. Oper Res 59(1):133–142

Borgelt C, Berthold MR(2002) Mining molecular fragments: finding relevant substructures of molecules. In: ICDM, p 51

Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth and Brooks, Monterey

Camargo S, Kossin J, Sitkowski M (2010) Climate modulation of North Atlantic hurricane tracks. J Clim 23:3057–3076

Chakrabarti D.(2004) AutoPart: parameter-free graph partitioning and outlier detection. In: PKDD, pp 112–124

Chakrabarti D, Zhan Y, Faloutsos C (2004) R-mat: a recursive model for graph mining. In: SIAM international conference on data mining

Chan PK, Mahoney MV (2005) Modeling multiple time series for anomaly detection. In: ICDM, pp 90–97

Chen Z, Hendrix W, Samatova, N (2011) Community-based anomaly detection in evolutionary networks. J Intell Inf Syst 1–27. doi:10.1007/s10844-011-0183-2

Cheng H, Tan P, Potter C, Klooster S (2008) A robust graph-based algorithm for detection and characterization of anomalies in noisy multivariate time series. In: ICDM workshops, pp 349–358

Chu P, Zhao X, Lee C, Lu M (2007) Climate prediction of tropical cyclone activity in the vicinity of taiwan using the multivariate least absolute deviation regression method. Terr Atmos Ocean Sci 18(4):805–825

Clark JD, Chu PS (2002) Interannual variation of tropical cyclone activity over the Central North Pacific. JMSJ 80(3):403–418

Clauset A, Newman MEJ, Moore C (2004) Finding community structure in very large networks. Phys Rev E 1–6. www.ece.unm.edu/ifis/papers/community-moore.pdf

Donges JF, Zou Y, Marwan N, Kurths J (2009) Complex networks in climate dynamics. Eur Phys J Special Top 174(1):157–179. doi:10.1140/epjst/e2009-01098-2

Eberle W, Holder, L (2007) Discovering structural anomalies in graph-based data. In: ICDM workshops, pp 393–398
Elsner J (2001) Tracking hurricanes. AMS 84:353–356
Fayyad UM, Irani K (1993) Multi-interval discretization of continuous-valued attributes for classification learning. In: IJCAI, pp 1022–1027
Ganguly AR, Steinhaeuser K, Erickson DJ, Branstetter M, Parish ES, Singh N, Drake JB, Buja L (2009) Higher trends but larger uncertainty and geographic variability in 21st century temperature and heat waves. Proc Natl Acad Sci 106(37):15555–15559
Gill R, Datta S, Datta S (2010) A statistical framework for differential network analysis from microarray data. BMC Bioinf 11(1):95+. doi:10.1186/1471-2105-11-95
Girvan M, Newman ME (2002) Community structure in social and biological networks. Natl Acad Sci USA 99:7821–7826
Goldenberg S, Shapiro L (1996) Physical mechanisms for the association of El Niño and West African rainfall with Atlantic major hurricane activity. J Clim 9(6):1169–1187
Gozolchiani A, Yamasaki K, Gazit O, Havlin S (2008) Pattern of climate network blinking links follows el niño events. EPL 83:28005
Hey T, Tansley S, Tolle K (2009) The fourth paradigm: data-intensive scientific discovery. Microsoft Research, Redmond
Jensen LJ, Kuhn M, Stark M, Chaffron S, Creevey C, Muller J, Doerks T, Julien P, Roth A, Simonovic M, Bork P, von Mering C (2009) STRING 8—a global view on proteins and their functional interactions in 630 organisms. Nucleic Acids Res 37(Database):D412–D416
Jolliffe IT, Stephenson DB (2003) Forecast verification: a practitioner's guide in atmospheric science. Wiley, New York
Kalaev M, Bafna V, Sharan R (2008) Fast and accurate alignment of multiple protein networks. In: RECOMB, pp 246–256
Kawale J, Chatterjee S, Kumar A, Liess S, Steinbach M, Kumar V (2011) Anomaly construction in climate data: issues and challenges. In: CIDU, pp 189–203
Kim HM, Webster PJ (2010) Extended-range seasonal hurricane forecasts for the North At lantic with a hybrid dynamical-statistical model. Geophys Res Lett 37(21):L21705
Kim HS, Ho CH, Chu PS, Kim JH (2010) Seasonal prediction of summertime tropical cyclone activity over the East China Sea using the least absolute deviation regression and the Poisson regression. Int J Climatol 30(2):210–219
Magill T, Christopher J, Magill TH, Clark JV, Melick CJ, Market PS (2008) The interannual variability of hurricane activity in the art. NWD 32:1–15
Moonesinghe H, Tan PN (2006) Outlier detection using random walks. In: ICTAI, pp 532 – 539
Newman MEJ (2003) The structure and function of complex networks. SIAM Rev 45(2):167–256
Pei J, Jiang D, Zhang A (2005) Mining cross-graph quasi-cliques in gene expression and protein interaction data. In: Proceedings of the 21st international conference on data engineering (ICDE 2005), pp 353–356
Pei J, Jiang D, Zhang A (2005) On mining cross-graph quasi-cliques. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, KDD '05. ACM, New York, pp 228–238. doi:10.1145/1081870.1081898
Peng J, Yang L, Wang J, Liu Z, Li M (2008) An efficient algorithm for detecting closed frequent subgraphs in biological networks. In: BMEI, pp 677–681
Saunders M, Harris A (1997) Statistical evidence links exceptional 1995 Atlantic hurricane season to record sea warming. JGRL 24:1255–1258
Seidman SB, Foster BL (1978) A graph-theoretic generalization of the clique concept. J Math Sociol 6:139–154
Sharan R, Ideker T, Kelley B, Shamir R, 2004, RMK, Karp RM (2005) Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. J Comput Biol 12(6):835–846
Steinhaeuser K, Chawla NV, Ganguly AR (2009) An exploration of climate data using complex networks. In: SensorKDD, pp 23–31
Steinhaeuser K, Chawla NV, Ganguly AR (2011) Complex networks as a unified framework for descriptive analysis and predictive modeling in climate science. Stat Anal Data Mining 4(5):497–511
Sun J, Faloutsos C, Papadimitriou S, Yu PS (2007) Graph scope: parameter-free mining of large time-evolving graphs. In: KDD, pp 687–696

Sun J, Qu H, Chakrabarti D, Faloutsos C(2005) Neighborhood formation and anomaly detection in bipartite graphs. In: The fifth IEEE ICDM, pp 418–425
Sun J, Tao D, Faloutsos C (2006) Beyond streams and graphs: dynamic tensor analysis. In: KDD '06, pp 374–383
Sutton RT, Jewson S, Rowell DP (2000) The elements of climate variability in the tropical atlantic region. J Clim 13:3261–3284
Tan A, Gilbert D (2003) Ensemble machine learning on gene expression data for cancer classification. Appl Bioinf 2:S75–S83
Tao D, Tang X, Li X, Wu X (2006) Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Trans Pattern Anal 28:1088–1099
Tsonis A, Roebber P (2004) The architecture of the climate network. Physica A 333:497–504
Tsonis A, Swanson K(2008) Topology and predictability of el niño and la niña networks. Phys Rev Lett 100(22):228502
Tsonis A, Swanson K, Kravtsov S (2007) A new dynamical mechanism for major climate shifts. GRL 34:L13705+
Tsonis A, Swanson K, Roebber P (2006) What do networks have to do with climate?. BAMS 87(5):585–595
Tsonis A, Swanson K, Wang G (2008) On the role of atmospheric teleconnections in climate. J Clim 21:2990–3001
Tsonis A, Wang G, Swanson K, Rodrigues F, Costa L (2010) Community structure and dynamics in climate networks. Clim Dyn 1–8. doi:10.1007/s00382-010-0874-3
Wakita K, Tsurumi T (2007) Finding community structure in mega-scale social networks. CoRR abs/cs/0702048
Xie L, Yan T, Pietrafesa L (2005) The effect of Atlantic sea surface temperature dipole mode on hurricanes: implications for the 2004 Atlantic hurricane season. JGRL 32:3701+
Yeshanew A, Jury MR (2007) North african climate variability. Part 3: resource prediction. Theor Appl Climatol 89(1–2):51–62
Zeng Z, Wang J, Zhou, L, Karypis G (2006) Coherent closed quasi-clique discovery from large dense graph databases. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06. ACM, New York, pp 797–802. doi:10.1145/1150402.1150506
Zeng Z, Wang J, Zhou L, Karypis G (2007) Out-of-core coherent closed quasi-clique mining from large dense graph databases. ACM Trans Database Syst 32(2):13
Zhang B, Li H, Riggins RB, Zhan M, Xuan J, Zhang Z, Hoffman EP, Clarke R, Wang Y (2009) Differential dependency network analysis to identify condition-specific topological changes in biological networks. Bioinformatics 25(4):526–532. doi:10.1093/bioinformatics/btn660