Community Detection in Social Media by Leveraging Interactions and Intensities

Maria Giatsoglou, Despoina Chatzakou, and Athena Vakali

Informatics Department, Aristotle University of Thessaloniki
{mgiatsog,deppych,avakali}@csd.auth.gr

Abstract. Communities' identification in topic-focused social media users interaction networks can offer improved understanding of different opinions and interest expressed on a topic. In this paper we present a community detection approach for user interaction networks which exploits both their structural properties and intensity patterns. The proposed approach builds on existing graph clustering methods that identify both communities of nodes, as well as outliers. The importance of incorporating interactions' intensity in the community detection algorithm is initially investigated by a benchmarking process on synthetic graphs. By applying the proposed approach on a topic-focused dataset of Twitter users' interactions, we reveal communities with different features which are further analyzed to reveal and summarize the given topic's impact on social media users.

Keywords: community detection, user weighted interaction networks

1 Introduction

Events and topics emerging in the real world and social networks influence one another mutually due to social media users activities which have radically changed information dissemination and people's opinions communication. In social media frameworks, such as in microblogs, event-relevant information (in the form of news broadcasts or opinion snapshots) is directly propagated to the users' *followers* but it can be discovered by other users as well (by public posts). *User interaction networks* capture users' associations derived from their activities in social media such as: commenting on others' posts, replying to comments, referencing other users, etc. Variation in users' interaction frequency or intensity can be captured by the assignment of different strengths to the networks' associations, thus resulting in weighted networks. Earlier research has indicated that several types of user interaction networks (such as coauthorship networks [15], communication networks [12], friendship networks [21]) exhibit a structure of non random nature and organize around *communities*. Communities can be generally defined as *groups of users that are "closely-knit"*, in the sense that a group's interconnections are more dense compared to connections with the rest of the network.

Here we study user communities in social media users interaction networks as formed around a given theme/topic related to a real-world event. Our focus is on revealing the types of communities generated with respect to certain events by analyzing them in the dimensions of size, topic diversity and time span. This work builds on the idea that user interaction strengths are crucial in communities formation and that the detection and qualitative characterization of communities can lead to a better understanding of the impact of real world events on society. To achieve this, both inherent structural measures and emergent features are needed. These structural properties and

measures are here related to the well-known community detection algorithm SCAN [22], due to its scalability and its capability to detect hubs and outliers, adapted for weighted networks.

In summary this work's main contributions are as next:

- *adapts a weighted similarity measure* which encompasses both the structural properties and the weighted connectivity patterns existing in the locality of nodes. This measure extends *structural similarity* and brings closer nodes that not only share common neighbors, but are also connected to them with matching intensities.
- *deals with inherent limitations in local structural / density-based community detection algorithms* which characterize SCAN driven approaches. The proposed method reveals communities in weighted networks based on *weighted structure connected order of traversal* [4] and an approximate peak detection approach inspired by [18].
- *introduces a community meta-analysis approach* that highlights the usefulness of communities' detection on user interaction networks to reveal and summarize real world events' impact. Our Twitter case study validates the proposed methodology.

2 Related Work

Community detection has been applied to user interaction networks such as e.g. to the Enron e-mail exchange dataset [19] and coauthorhips networks [15], however few works have tackled community detection in social media users' interaction networks [9, 12]. In social media's context, community detection has been mainly applied to friendship networks generated by the declared users' affiliations [8], resulting in easily-interpreted groups of users. Thus, interaction networks may be of more complex nature with their derived communities' interpretations being non-obvious since interactions within their members may indicate that they are both aware of each other in the web world, while also interested in common topics. Important aspects of community detection in social media are covered in [17] where the need to detect meaningful communities of nodes as well as identify *hub* and *outlier* nodes is highlighted. This requirement is addressed in SCAN [22], a community detection algorithm which builds on the density-based clustering algorithm DBSCAN [6]. While DBSCAN has been widely used for clustering spatial points based on their density distribution, SCAN operates on graphs based on a *structural similarity* measure. The main limitation of DBSCAN and SCAN is their sensitiveity to the selection of an initial similarity threshold parameter, whose fine-tuning requires repeated algorithm executions for several parameter values. An approach to alleviate this limitation in SCAN was given in [20] with the clustering quality *modularity* criterion [14] being used to find the optimal parameter's value.

Alternative efforts to address the problem of DBSCAN and SCAN parameter produced the so called *reachability plots* [1, 4], which represent the algorithms' multiple clustering outcomes for every possible parameters' combination. A technique proposed in [18] operates on reachability plots produced by DBSCAN to automatically determine significant clusters. Up to now SCAN's applicability to weighted networks has only been addressed in [20] where a structural similarity measure for weighted networks is proposed, but no explicit experimental results are offered for such networks. Thus, all previous efforts were tested on limited (unweighted) synthetic or *closed-world* networks. *Closed-world* networks are limited within the scope of a certain "community" (e.g. the Enron email network with internal company email exchanges), when on the contrary, social media users' interactions are of an open nature since they generate networks "connecting" people of different disciplines and wider scope. Topic- and event-specific networks can be derived from broader social media generated networks

by keeping as edges only interactions relevant to the given topic/event. Users, in principle, interact with different intensities, the level of which can be inferred by the interactions' frequency, duration, etc. Although weighted networks are a natural representation of such interactions' intensities, many community detection approaches for real world datasets, operate on unweighted networks after preserving either all relationships, or only those whose intensity is above a cut-off threshold. Here, we examine whether SCAN approaches, which generally have desirable traits for application to user interaction networks, can successfully uncover the underlying community structure in real world networks, or they need to be adapted to leverage the interactions' intensity. SCAN and the proposed adaptation for weighted networks WSCAN (i.e. WeightedSCAN) are evaluated on a series of synthetic networks. The combination of both approaches' experimental results with the corresponding intrinsic network properties (the global *clustering* and *weighted clustering* coefficients [16]) leads to an empirical criterion for the selection of SCAN or WSCAN for the network at hand. WSCAN's limitation of parameter selection is also addressed by an automatic approach, AutoWSCAN, which detects communities from nodes' weighted structure connected order of traversal, inspired by [18], and is validated in synthetic and real-world event-centric networks.

3 Proposed Methodology

To generate users' interaction networks given an event-related topic T, we first aggregate for a given period of time ΔT user activity data from selected social media applications, then extract the observed interactions Int_t, and connect users who have interacted at least once. An edge connecting nodes u and v is weighted by $w_{u,v} = \sum_{t \in \Delta T} Int_t(u, w)$. To detect communities in user networks embedding interaction strengths, here we adapt existing SCAN-based algorithms, and propose the use of WSCAN and AutoWSCAN.

3.1 Getting from SCAN to WSCAN

SCAN [22] discovers cohesive network subclusters based on parameters μ and ϵ, which control the minimum community's size and the minimum *structural similarity* between two community's nodes, respectively. Generally, a larger μ value leads to fewer and larger communities, while a larger ϵ value to tighter communities and more outliers. Using structural similarity as a clustering criterion, nodes with several common neighbors are placed in the same (μ, ϵ)-core community. To adapt SCAN for weighted interaction networks we propose *weighted structure reachability* for (μ, ϵ)-cores' detection.

Definition 1 *Given a weighted undirected network (G, w), where $G = \{V, E\}$ and $w : E \to \mathbb{R}$, the* ***weighted structural similarity*** *$wSSim$ of two nodes, u and v, is defined as:*

$$wSSim(u, v) = \frac{\sum_{k \in \Gamma(u) \cap \Gamma(v)} w_{u,k} \cdot w_{v,k}}{\sqrt{\sum_{k \in \Gamma(u)} w_{u,k}^2} \sqrt{\sum_{k \in \Gamma(v)} w_{v,k}^2}} \ . \tag{1}$$

where $\Gamma(v)$ is the ***neighborhood*** *of node v: $\Gamma(v) = \{k \in V | (v, k) \in E\} \cup \{v\}$, $w_{u,v} \in [0, 1) | u \neq v; w_{u,v} = 1 | u = v$.*

Definition 2 *The ϵ-neighborhood of a given node u is the subset of its neighborhood containing only nodes that are at least ϵ-similar with u:*

$$N_\epsilon(u) = \{v \in \Gamma(u) | wSSim(u, v) \geq \epsilon\} \ . \tag{2}$$

Definition 3 *A vertex v is called a* (μ, ϵ)***-core*** *if its* ϵ*-neighborhood contains at least* μ *vertices:* $CORE_{\mu,\epsilon}(v) \Leftrightarrow |N_\epsilon(v)| \geq \mu$.

Additional nodes are attached to (μ, ϵ)-cores based on *structural connectivity*. A node u is *structure-reachable* from a core node v if u can be reached from v through a chain of nodes each belonging to the ϵ-neighborhood of the previous one. Nodes u and v are *structure-connected* if they are *reachable* from the same core node k. A community is then defined as a set of structure-connected nodes that is maximal in terms of structure reachability. Nodes not assigned to any communities, are characterized as either outliers or hubs depending on whether they are linked to a single or multiple communities, respectively. For the calculation of wSSim it is important to ensure that all weights are < 1, since a weight of 1 is used as each node's self-similarity in the definition of wSSim. To achieve this, we scale all interactions' weights before community detection.

3.2 AutoWSCAN

Our experiments with WSCAN showed its high sensitivity to parameter ϵ. Finding an ϵ value that leads to a balanced community structure regarding outliers' number, coherence, and communities' separation is, though, tedious. A heuristic approach is proposed in [22] for selecting the ϵ value based on the "knee-point hypothesis" for the μ-nearest neighbor similarity plot. Thus, our application of this approach to real-world networks with both the "unweighted" and weighted structural similarity, did not reveal clear knee-points at such plots. We rather adopt the structure connected order of traversal which represents all *structure-connected* community sets detected in a network for all possible ϵ values [4]. To this end, nodes are re-ordered by structure-connected order of traversal based on *weighted core reachability* and *weighted reachability similarity*, defined next.

Definition 4 *Given a network* (G, w)*, the* ***weighted core reachability*** *wCS im of node u is defined as:*

$$wCS\,im(u) = \begin{cases} wS\,S\,im(u, \mu NN(u)), if\ |\Gamma(u)| \geq \mu \\ UNDEFINED, else \end{cases}, \tag{3}$$

where $\mu NN(u)$ *is the* μ*-nearest neighbor of node u.*

Definition 5 *Given a network* (G, w)*, the* ***weighted reachability similarity*** *wRS im of node v from node u is defined as:*

$$wRS\,im(v, u) = \begin{cases} max\left(wCS\,im(u), wS\,S\,im(u, v)\right), if\ |\Gamma(u)| \geq \mu \\ UNDEFINED, else \end{cases}. \tag{4}$$

Weighted core reachability is calculated for each node, standing for the minimum ϵ value that would allow this node to become a core (Alg. 1). Then, each possible core node u ($|\Gamma(u)| \geq \mu$) is "visited", a process that involves finding the node's neighbors, calculating their weighted reachability similarity from the current core, and inserting them at a priority queue based on the *wRS im* value (or reordering the queue if they have already been inserted). At each iteration, the node with the highest *wRS im* value from any previously visited node is extracted from the queue to ensure that regions of higher weighted structural similarity are spanned before surrounding areas of lower similarity [4]. The node visiting order represents the weighted structure connected order of traversal. For a connected network, the algorithm will never return to its first loop, thus, since thematic social media users' interaction networks are often disconnected, this is probable. Our approach is to generate partial nodes' sequences based on structure-connected order of traversal for each disconnected component and detect communities in them.

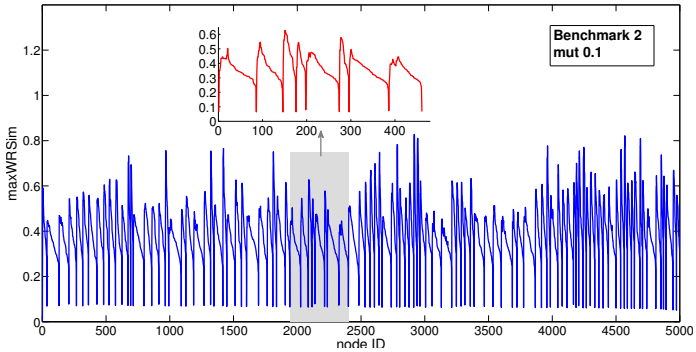


Fig. 1: Weighted reachability plot for Benchmark 2 with $\mu_w = 0.1$

The weighted structure-connected order of traversal can be depicted via a *reachability plot*, which illustrates, in the corresponding order, the maximum weighted reachability value of each node from its previously visited nodes (referred to as *maxWRS im*). Such a reachability plot is depicted in Fig. 1, where we can observe areas in which the *maxWRS im* values steadily rise and then fall at a local minimum to rise again after a while. Such "hills" represent different communities, whereas areas of low *maxWRS im* values are outliers. Such communities can be revealed by 'slicing' the plot horizontally at a selected global similarity threshold, and isolating the regions that lay above it.

Definition 6 *Given a sequence of nodes* $\{n_1, n_2..., n_{|V|}\}$ *ordered based on weighted structure-connected order of traversal, a community is defined with respect to* ϵ_{thres} *as a subsequence of nodes* $\{n_{a-1}, n_a, ..., n_b\}$ *where* $1 < a-1 < b \leq |V|$, *iff* $\forall i \in [a, b], maxWRS\,im(n_i) \geq \epsilon_{thres}$ *and [a,b] is maximal.*

Since in real world networks communities are usually of different cohesion and strength, a global ϵ_{thres} will fail to identify communities of different *similarity-range* scales. Thus, to detect communities at different (local) ϵ values, we apply AutoWSCAN, an algorithm inspired by [18]. AutoWSCAN (Alg. 2) detects communities as contiguous areas between two local minima, satisfying some desired properties that reflect the way a person would identify communities by observing a reachability plot. AutoWSCAN gets a weighted reachability plot and first identifies local minima points, ensuring that they have the lowest value in a subregion centered on them and spanning $2 \cdot \mu$ points. Then, it puts them in a priority queue by increasing value, and iteratively removes the first point from the queue and uses it to split the input sequence of nodes in two subregions. A split point is considered valid only if the generated subregions differ noticeably in their *maxWRS im* values compared to the split's value. Thus, we check that the maximum value in each region is "significantly" larger than the split point's *maxWRS im* (with use of a *minRatio* $\simeq 0.7$). AutoWSCAN is recursively called for each subregion whose size is larger than μ (active), and the same process is applied for the subregion based on the minima points within its span. If there are no more (valid) minima points or both subregions are inactive, then the current region is a community.

3.3 Benchmarking Framework

Our initial hypothesis that WSCAN and AutoWSCAN are more suited for real world user interaction networks compared to SCAN needs to be experimentally validated. Since, to our knowledge, there exist no real world weighted networks with ground truth

Algorithm 1: WeightedSCOT

```
Input: G = (V, E, w), μ
Output: A sequence of nodes in structure-connected order of traversal.
  foreach node v ∈ V do
    if v not visited then
      Cl=AutoSCAN (orderedList)
      Communities.append (cl)
      orderedList =null
    enqueueNeighbors (v)
    if v is core then
      while visitQueue in not empty do
        currNode = visitQueue.getNode ()
        visitNodes.add (curNode)
        enqueueNeighbors (curNode)
Function enqueueNeighbors (v)
  v.visited = true
  orderedList.append (v)
  cs = computeWCoreSim (v)
  if v is core then
    foreach vN in v.neighbors do
      if visitedNodes not contains vN then
        ss = getWStructuralSim (v, vN)
        newWRSim = min (cs, ss)
        if vN.wReachSim is null then
          vN.wReachSim = newWRSim
          visitQueue.update (vN, newWRSim)
        elif newWRSim > vN.wReachSim then
          vN.wReachSim = newWRSim
          visitQueue.setPriority (vN, newWRSim)
  EndFunction
```

communities, we utilize synthetic networks with planted partitioning of nodes in communities for the algorithm's evaluation. In specific, we use the well-known LFR benchmark graphs [10] since they support weights and possess some important real world networks' features (node degree and community size heterogeneous distributions). Our benchmarking involves the application of WSCAN and SCAN on a series of LFR graphs generated with different parameters for several linearly increasing values of the parameter ϵ, while maintaining the same value for parameter μ. The accuracy of each run is evaluated by the well known Normalized Mutual Information (NMI) score [5], which quantifies the closeness between the identified communities and the ground-truth communities in a scale of 0 to 1 (1 denotes identical assignment of nodes to communities). For each graph we record the best NMI score achieved and the corresponding ϵ value. To assess the performance of AutoWSCAN, we apply it on the same graphs, and also compare it with a modified implementation for unweighted graphs, AutoSCAN. The latter follows exactly the same process as AutoWCAN with the exception that is uses the classic (unweighted) measures of core reachability and reachability similarity.

SCAN-based approaches might characterize some nodes as outliers or hubs and not assign them to a community, as opposed to the LFR graphs which consider that each node belongs to at least one community. Since we are not aware of any weighted benchmark network with known community structure embedding also outliers and hubs, we adopt the LFR benchmark graphs and follow a workaround to extract NMI scores. Thus, upon the algorithms' execution, we assign i) outliers to the community with which they

Algorithm 2: AutoWSCAN

```
Input: partialWReachabilityPlot: maxWRSim
Output: Clusters
  find localMinima
  order localMinima from min to max
  pNode.setRange(reachVal(start), reachVal(end))
  return findClusters(pNode, localMinima)
Function findClusters(treeNode,localMinima)
  if localMinima is empty then
    if sizeOf(treeNode) > μ then
      treeNode is a cluster
      return
  lMin = localMinima.pop
  [leftNode, rightNode] = split(treeNode, lMin)
  remove all points before/after lMin that have
  the same wReach value
  if sizeOf(leftNode) > μ then leftNode: active
  if sizeOf(rightNode) > μ then rightNode: active
  if leftNode & rightNode inactive
    then treeNode is a cluster
  foreach activeNode do
    find its maximum wReach_max value
    if (lMin/wReach_max) > minRatio then
      ignore split point
      findClusters(treeNode, localMinima)
    actMinima = localMinima in activeNode's range
    findClusters(activeNode, actMinima)
EndFunction
```

have at least one connection, and ii) hubs to the community towards which they are most strongly connected based on the (weighted) structural similarity or (weighted) reachability score for (W)SCAN and Auto(W)SCAN, respectively. This evaluation approach, although not optimal, reflects as accurately as possible how closely the given algorithm approximates ground-truth communities.

After obtaining the NMI scores for all approaches, we seek to reason their comparative performance by examining the benchmark graphs' structural properties. To this end, we employ two metrics: the global clustering coefficient and weighted clustering coefficient. The global clustering coefficient, *CC*, expresses the density of triplets of nodes in a network, where a *triplet* comprises three nodes connected by two (*open triplet*) or three edges (*closed triplet*). It is defined as 3 times the number of closed triplets (for each pair of the triangle's edges) over the total number of triplets at the network, and its value ranges from 1 for a fully connected network to 0 for random networks with sufficiently large size. A similar idea is followed by the global weighted clustering coefficient, *wCC*, in weighted networks [16]. By assigning a value to each triplet, *wCC* is defined as the sum of all closed triplets' values over the sum of all triplets' values. Four methods have been proposed for the calculation of a triplet's value: the arithmetic mean, geometric mean, maximum, and minimum of the weights of the corresponding two edges. Of all four approaches, we select to use the geometric mean since it is considered the most appropriate for alleviating sensitivity to extreme weights. The definition of *wCC* implies that for a random distribution of weights in the network, *wCC* equals to *CC*. Here, for each network we calculate the ratio of *wCC* to *CC* and observe the performance of the algorithms when this ratio is greater or lower than 1.

4 Experiments and Results

The proposed approaches, WSCAN and AutoWSCAN, are compared with their unweighted counterparts, SCAN and AutoSCAN, in terms of their performance on the LFR benchmark framework. Our aim is to determine the validity of the proposed methods and their suitability for graphs that exhibit real world features. Since disregarding the variability of the intensity of interactions in real world networks is a common approach, here we try to identify how it affects performance and in which situations it can be safely followed. We also apply AutoWSCAN to a user interaction network from Twitter, focused on a real world event-related topic, and identify features of the detected communities that can be leveraged to gain insights regarding the event's impact.

4.1 Synthetic Networks

To evaluate the algorithms' performance, we apply them on four weighted LFR graphs, whose complexity is governed by the *topological mixing* (μ_t) and the network's *weighted mixing* (μ_w) parameters [10]. Since μ_t is the ratio of the number of a node's external neighbors to the node's total degree, its increasing values indicate mixed and difficult to separate communities. μ_w has a similar effect, since it is the ratio of the sum of the weights of the edges between a node and its neighbors in different communities to the sum of the all nodes' incident edges. Table 1 outlines the parameter combination for each generated benchmark graph. Benchmarks 1 and 3 refer to graphs with smaller communities (10-50 nodes per community) compared to Benchmarks 2 and 4 (20-100 nodes nodes per community). Also, graphs of Benchmarks 1 and 2 (with $\mu_t = 0.5$) have a more apparent community structure compared to Benchmarks 3 and 4 (with $\mu_t = 0.8$). Since we are interested in how weights affect the community detection results, we perform runs of SCAN, AutoSCAN, WSCAN and AutoWSCAN for varying values of μ_w. Fig. 2 depicts NMI scores for all runs on the four benchmark graphs (with $\mu = 4$).

Table 1: Synthetic Benchmark Graph Specification

	n	k	k_{max}	min_c	max_c	μ_t
Benchmark 1	5000	20	50	10	50	0.5
Benchmark 2	5000	20	50	20	100	0.5
Benchmark 3	5000	20	50	10	50	0.8
Benchmark 4	5000	20	50	20	100	0.8

As expected, the performance of (Auto)SCAN is invariable with respect to μ_t for all benchmarks, since its operation is not affected by changes at the edges' weights. The performance of (Auto)WSCAN is satisfactory for the NMI score, since its starts to decay at $\mu_w \approx 0.5$. Lower NMI values are expected for high μ_w values, since, then, the algorithms characterize more nodes as outliers/hubs and assign them to communities based on the workaround described in Sect. 3.3. For Benchmarks 1 and 2 the weighted algorithms perform better than (Auto)SCAN for $0.1 \leq \mu_t \leq 0.4$, while the corresponding set of graphs exhibit $wCC/CC > 1$ (as depicted in Fig. 3). For $\mu_t \geq 0.5$ unweighted graphs maintain a good performance for Benchmark 1, whereas they perform poorly for all graphs of Benchmark 2 (with bigger communities and $CC < 0.1$). On the contrary, larger community sizes do not significantly affect (Auto)WSCAN's performance, since NMI scores for Benchmarks 1 and 2, as well as for Benchmarks 3 and 4 are similar. NMI scores from Benchmarks 3 and 4 indicate that the weighted algorithms perform better for $\mu_t = 0.8$, rather than for $\mu_t = 0.5$ (Benchmarks 1 and 2). This may seem contradictory, however, as explained in [10], when $\mu_t < \mu_w$ inter-communities edges carry on average more weight, rather than when $\mu_t > \mu_w$. This is inconsistent with most

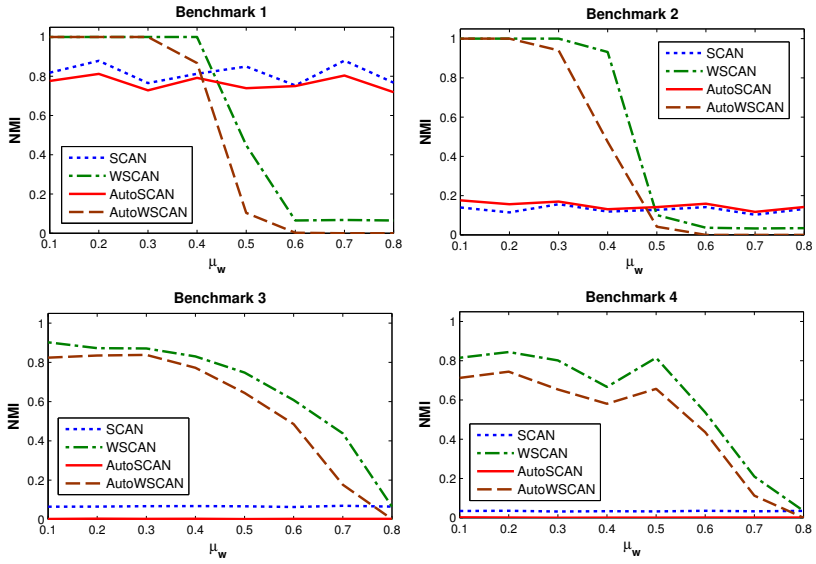


Fig. 2: NMI scores for the algorithms' benchmarks with varying value of μ_w

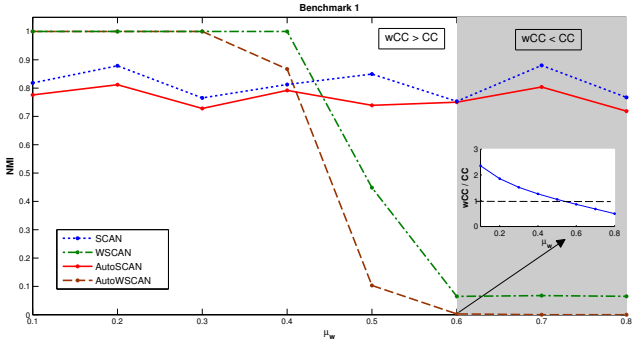
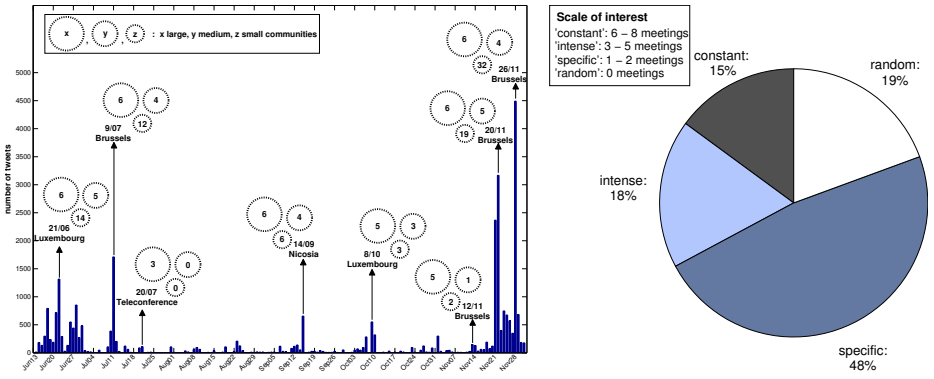


Fig. 3: NMI scores for Benchmark 1 combined with the evolution of the ratio of wCC to CC for increasing value of μ_w (depicted in the embedded plot)

community detection algorithms' hypothesis that intra-community nodes are connected with highly-weighted edges. For all graphs of Benchmarks 3 and 4 the unweighted algorithms fail to detect the community structure. An important observation is that for all these graphs $wCC/CC > 1$ (except for when $\mu_t = 0.8$, where $wCC/CC = 1$). Our results indicate that the decision of whether to apply (Auto)SCAN or (Auto)WSCAN on a given network could be based on the ratio of wCC to CC, selecting the first when it is $<$ 1, or the second otherwise. In all cases, the automatic algorithms follow closely the best performance of their unweighted counterparts. This is a significant outcome considering the temporal cost induced by the search of the (ϵ) parameter space in (W)SCAN. In our experiments, while the selected value for SCAN is ~0.2 for all graphs, in WSCAN it increases for rising value of μ_w with no common pattern over all graphs. The selected $epsilon$ value for all runs where WSCAN performs satisfactorily ($NMI > 0.5$) ranges from 0.04 to 0.28, it thus appears difficult to estimate it in advance. AutoWSCAN emerges as a good alternative to WSCAN, since it is independent of this parameter and performs similarly to WSCAN under the parameter setting leading to the best results.

4.2 Real-World Networks

Our target case-study is to apply community detection in real-world user interaction networks and identify emergent community structure's features for the characterization of real world events based on their impact. For experimentation we have generated a network based on Twitter user interactions, (i.e. *mentions*, *replies*, *retweets*), extracted from data collected via the Twitter Streaming API with topic-related keywords. Our selected topic refers to the official Eurogroup meetings (of Eurozone's finance ministers), which have attracted major interest due to the recent financial crisis and the Eurogroup's role in important decision taking. Our EUROGROUP dataset (covering 8 meetings from 13/06/12 to 30/11/12) acts as an exemplary case study of a series of events held at different time instances, having the same participants with a common generic context (i.e. the Eurozone's monetary issues), but different focus (depending on the agenda). The dataset spans 227 days and comprises: 29529 tweets, 10305 interactions and 3015 different users. Regarding the interactions' type, retweets span more than 50% of the total interactions, thus they affect considerably the networks' shape (star-like forms). Statistical features such as tweet frequencies, depicted in Fig. 4(a), can be used to obtain some initial insights for an event's popularity in Twitter (e.g. more intense activity towards late November). Here, we are mostly interested in the users' clustering around such periods claiming that communities' emergent features reveal finer aspects of events.



(a) (b)

Fig. 4: EUROGROUP meetings, tweets, and communities: (a) depicts the daily number of tweets and is annotated by the meetings' dates and locations. The number of active communities per meeting is depicted above its corresponding day; (b) shows a distribution of the communities in a scale of users' interest based on their members' activity on the events' dates

Before we apply community detection, we normalize all weights and calculate *wCC* and *CC* for the user interaction network, resulting at a ratio of $wCC/CC = 1.22$. *wCC* is, thus, larger than *CC* implying that the intensities of user interactions are not random in this network, but play indeed an important role in communities' formation. Therefore, based on the observations of Section 4.1 we opt to apply AutoWSCAN for the detection of communities. AutoWSCAN reveals 67 communities which we further analyze on three feature axes: size, topic diversity, and time span. *Size* simply refers to the number of users that are members of a given community, and is indicative of the community's popularity. By analyzing the tweets corresponding to intra-community user interactions, we can infer more refined topics that interest each community's members. This analysis involves extracting the text of all inter-community tweets containing interactions between its members, and applying LDA [3] to detect topics within them.

Since LDA requires specifying the number of topics to be detected, we empirically set this parameter to 100. Each document in LDA is a mixture of various topics with different probabilities. Here, due to the small length of tweets' text, a tweet is most likely to belong to a single topic, thus we assign it to the most probable (topic). Then, each community is characterized by the set of different topics expressed within its relevant tweets, and is associated with the feature of *topic diversity* which refers here to the size of its topic set. Finally, each community's *time span* is simply derived by taking the length of temporal duration covered by its corresponding tweets (at a day granularity).

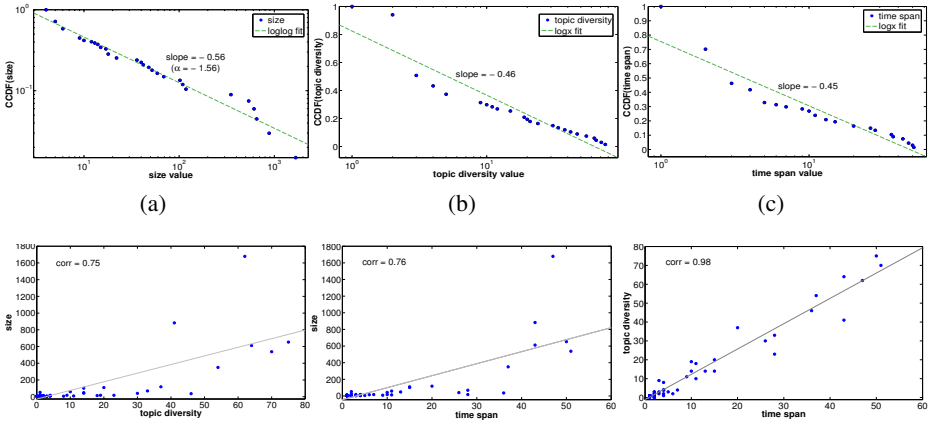


Fig. 5: Estimated distributions of the detected communities' size, topic diversity, and time span are depicted in (a), (b), and (c), respectively. Scatter plots for pairs of features are depicted in (d) for size and topic diversity, (e) for time span and size, and (f) for time span and topic diversity.

Figs. 5(a), (b), (c) depict an estimated distribution of communities' size, topic diversity, and time span, using their Complementary Cumulative Distribution Functions (CCDF), which represents for feature f_i the probability $P(f_i > x)$. Size's CCDF exhibits a slow, power law decay with exponent 0.56, and a p-value of 0.78, indicating good fit. Thus, it can be derived that communities' size also follows a power law distribution with an exponent of $\alpha = 1 + 0.56 = 1.56$ [13]. Similar results for community size have been documented in [2]. Careful observation of Fig. 5(a) reveals a knee at ~538, beyond which CCDF decays faster. This indicates that there are fewer communities of very large size in the dataset compared to these dictated by the power law distribution that fits communities of less that 538 members. Topic diversity and time span do not exhibit a power law distribution, but they have a logarithmic relationship, since their CCDFs can be both fitted best by a exponentially decaying line with slope ~0.45 in a lin-log plot. Fig. 5(b) reveals that 94% and 50% of the communities cover more that 2 and 3 topics, respectively, while the intra-community topics' number is best fitted by the exponential distribution after $f_{topic} = 3$. Fig. 5(c) shows that only 46% of the communities span more than 2 days, indicating that roughly half of the communities are short-lived.

Figs. 5(d), (e), and (f) depict the scatter plots generated for the features of size & topic diversity, size & time span, and time span & topic diversity, respectively. By plotting the least-squares line, we get a strong correlation of ~0.75 for both the size-topic diversity and size-time span feature pairs, as well as a very strong correlation of 0.98 for topic diversity and time span. These results indicate that larger communities cover, in general, more topics which was up to a point expected, but they are also active

for a longer duration of time. This might be explained by the assumption that interest in small communities is focused on specific topics which correspond to a limited time period, whereas larger communities interact more frequently since they are interested in multiple relevant events. From both Figs. 5(d) and (e) we can observe that there is an outlier point at the largest community (f_{size} = 1670), which does not exhibit the expected magnitude in topic diversity and time span. This behavior could be attributed to the effect of retweets that may cause a significant increase in a community's size, but are focused on a single topic and are usually relevant to a single time-limited event.

To understand each Eurogroup meeting's impact, we associate them with the discovered communities and their features. We assume that each community expresses interest in an event, thus it is *active* on it, given that interactions between its members are observed on the current/previous/next day of the event. The number of active communities identified for each meeting are depicted in Fig. 4(a). To qualitatively characterize active communities, we further classify them as *small* (< 50 members), *medium* (50 ≤ members < 200), and *large* (≥ 200 members), and present their distribution for each event in the same figure. Since in total 6 large communities have been detected, we can observe that they are all active in 5 out of 8 events, which are also the events with the most tweets on the day they took place. This observation is inline with our previous analysis which indicated that larger communities generally cover more topics. Examination of the most popular events with respect to the tweets' number (20/11 and 26/11 in Brussels), reveals that although the latter one has attracted the most tweets, the earlier has more medium active communities. The meeting of 20/11 corresponds to the failure of European leaders to *reach an understanding of how to restructure Greece's aid package, thus delaying the next aid tranche*, whereas this of 26/11 to the *IMF's and eurozone's €40 billion debt-reduction agreement for Greece*[1]. Although apparently more buzz was generated on the day of the later event, it seems that the previous, a long critical meeting building up tension and failing to reach a result, has attracted the interest of more large and medium communities combined. The later event, on the contrary, has been of interest to more small communities, probably focused on its decision. By comparing the summer meetings of 21/6 and 9/7, we can observe that although the first has attracted less tweets than the second, it is related to more active communities. June meeting's target was *to discuss the latest developments in the eurozone, mainly in Greece, Spain, Portugal and Ireland*, whereas July's meeting aimed at *discussing EU/IMF's rescue programs for Spain, Greece and Cyprus*[2]. More topics seem to be involved in the first event which may, up to a point, explain interest's dispersion in more communities. Some communities active on June's meeting might also be interested in a related topic: the announcement of the successful formation of a new government in Greece (which happened after a critical long election period associated with the question of Greece's continued eurozone membership), which took place one day before the event. Communities are also characterized in terms of their interest in the "Eurogroup" topic based on the number of meetings on which they are active. We assess interest expressed within a community in the following scale: *constant*, *intense*, *specific*, *random*, based on whether the community is active on 6-8, 3-5, 1-3, or 0 meetings, respectively. Most communities appear to have *specific* interest on few meetings, thus, a considerable percentage of them are indeed *focused* on the topic (with intense or constant interest). To identify the most popular topics within tweets, we resort to the following approach. We form 3 orderings of topics by ranking each topic based on: A) the number of tweets

[1] http://blogs.cfainstitute.org/investor/2011/11/21/european-sovereign-debt-crisis-overview-analysis-and-timeline-of-major-events/

[2] http://www.consilium.europa.eu/

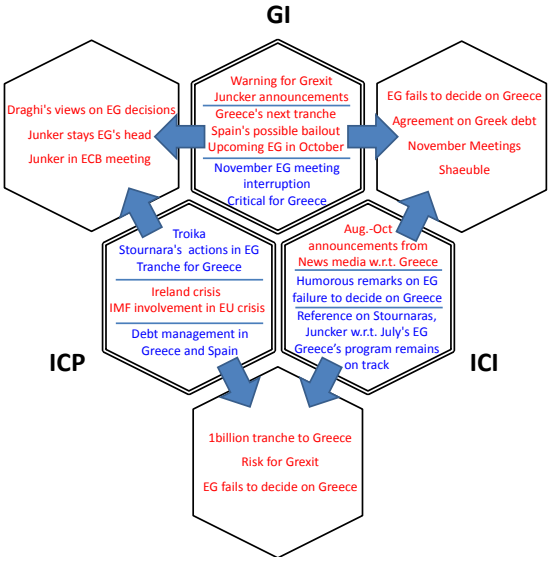


Fig. 6: Classification of the most significant topics based on interest intensity and diffusion. Horizontal lines separate different topics, while topics in red/blue correspond to the English/Greek language. Greek topics have been translated in English. (*Best viewed in color*)

that express it over all communities, B) the number of communities that are related to at least one tweet that expresses it, C) the number of communities that are *strongly-related* to it. For ordering C, we assign each community to a single topic, i.e. the one expressed in most of its members' interactions, and then we rank each topic by the number of communities assigned to it. A set of 12 unique topics is generated by taking the top-5 from each ordering. We define 3 topic features: *General Intensity* (GI), *Inter-Community Popularity* (ICP), and *Inter-Community Intensity* (ICI), which characterize topics that rank high (here, in the top-5) in ordering A, B, and C, respectively. In our set, there exist: 3 GI topics (which have the most intense user interest overall), 3 ICP topics (which reach out to the most communities), and 3 ICI topics (which play a major role in the most communities). There also exist 3 topics that combine two features, GI & ICP (attracting intense general interest while also being diffused in several communities), GI & ICI (attracting intense interest while also being major in several communities), and ICP & ICI (spanning several dedicated communities). Summaries of all 12 topics are depicted in Fig. 6, where the central hexagons correspond the GI, ICP, and ICI features, whereas the hexagons adjacent to two central ones represent the corresponding intersection. Topics are also divided based on their terms' language in English and Greek, since they are the ones represented in the set. It can be easily observed that all topics that combine two features (thus are more significant), are in English, indicating their significant impact on more users and communities. The most important illustrated topics along with their borderlines highlight users interest permutations.

5 Conclusions

In this work we propose a community detection approach for topic-focused interaction networks of social media users, which leverages both the structural properties of the network and the interactions' intensity. We investigate the role of weights in community

detection approaches based on structural similarity and the possibility to combine them with automatic parameter selection. Our approach's correctness is validated on a series of synthetic networks. Moreover, its application on a real world network combined with a community meta-analysis process enables us to better understand the dual relationship between real world events / topics, and community formation.

Acknowledgements This research was supported by Qualia, an on-line media monitoring and business intelligence company, situated in Athens, Greece, and the Research Committee of the Aristotle University of Thessaloniki, Greece.

References

1. Ankerst, M.; et al. 1999. Optics: ordering points to identify the clustering structure.In SIGMOD, 4960.
2. Arenas,A.; et al. 2004. Community analysis in social networks. European Physics Journal B, 38(2), 373-380.
3. Blei, D.M.; et al. 2003. Latent Dirichlet allocation. Journal of Machine Learning Research 3 (4-5), 9931022.
4. Bortner, D., and Han, J. 2010. Progressive clustering of networks using structure-connected order of traversal. In ICDE, 653656.
5. Danon, L.; et al. 2005. Comparing community structure identification. J. Stat. Mech. P09008(0505245).
6. Ester, M.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In KDD, 226-231.
7. Girvan, M., and Newman, M. E. J. 2002. Community structure in social and biological networks. Proc. of the National Academy of Sciences USA 99(12).
8. Jurgens, D., and Lu, T.-C. 2012. Friends, Enemies, and Lovers: Detecting Communities in Networks Where Relationships Matter. In WebScience.
9. Kamath, K. Y., and Caverle, J. 2011. Transient crowd discovery on the real-time social web. In WSDM, 585-594.
10. Lancichinetti, A., and Fortunato, S. 2009. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Phys. Rev. E 80(016118).
11. Lancichinetti, A.; et al. 2011. Finding statistically significant communities in networks. PLoS ONE 6, e18961.
12. Morrison, D.; et al. 2012. Evolutionary clustering and analysis of user behaviour in online forums. In ICWSM, 519-522.
13. Newman, M. E. J. 2005. Power laws, Pareto distributions and Zipf's law. Contemporary Physics 46, 323.
14. Newman, M. E. J., and Girvan, M. 2004. Finding and evaluating community structure in networks. Phys. Rev. E 69.
15. Newman, M. E. J. 2006. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74(0602124v1).
16. Opsahl, T., and Panzarasa, P. 2009. Clustering in weighted networks. Social Networks 31 (2), 155-163.
17. Papadopoulos, S.; et al. 2012. Community detection in social media. Data Mining & Knowledge Discovery 24:515-554.
18. Sander, J.; et al. 2003. Automatic extraction of clusters from hierarchical clustering representations. In PAKDD, 75-87.
19. Sun, J.; et al. 2007. Graphscope: parameter-free mining of large time-evolving graphs. In KDD, 687-696.
20. Sun, H.; et al. 2010. gskeletonclu: Density-based network clustering via structure-connected tree division or agglomeration. In ICDM, 481-490.
21. Ugander, J.; et al. 2011. The anatomy of the facebook social graph. CoRR abs/1111.4503.
22. Xu, X.; et al. 2007. Scan: a structural clustering algorithm for networks. In KDD, 824-833.