Chapter 12
On the Evaluation of Community Detection Algorithms on Heterogeneous Social Media Data

Antonela Tommasel and Daniela Godoy

Abstract One fundamental problem in social networks is the identification of groups of elements (also known as communities) when group membership is not explicitly available. Community detection has proven to be valuable in diverse domains such as biology, social sciences and bibliometrics. Thus, several community detection techniques have been developed. Nonetheless, as real networks are very heterogenous, the question of how communities should be assessed remains open. Whilst there are several works that have analysed the performance of diverse community detection algorithms over artificial graph benchmarks, the evaluation over real social networks has received comparatively less attention. Motivated by the lack of such studies, this chapter focuses on the analysis of the performance of community detection algorithms over social media networks, and the quantification of the structural properties of the discovered communities.

12.1 Introduction

Social networking and microblogging sites have increased their popularity in recent years attracting millions of users, who spend an increasing amount of time sharing personal information and making new friends. For example, sites like *Flickr*, *YouTube*, *Facebook* or *Twitter* allow users to create content, publish photographies, comment on content other users shared, tag content and socially connect with other users in the form of subscriptions or friendships. Consequently, social networking sites affect how people communicate and interact, leading to the formation of relationships of heterogeneous nature, origin and strength. Users might choose their friends because they publish interesting information, share common interests or common friends, or just because they are celebrities, amongst other possible explanations. Thereby, topological relations could lead to the existence of

A. Tommasel (✉) · D. Godoy
ISISTAN, CONICET-UNICEN, Tandil, Buenos Aires, Argentina
e-mail: antonela.tommasel@isistan.unicen.edu.ar

Deepak P, A. Jurek-Loughrey (eds.), *Linking and Mining Heterogeneous and Multi-view Data*, Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-01872-6_12

casual links. In this context, the significance and importance of relations should not be only analysed based on topological information, but in conjunction with other information sources or data views, which might implicitly define connections between social media users. For example, whether two users use the same terms or hashtags, or post on the same topics. It is worth noting that the content users consume or post might depend, for example, on their mood and environment [8].

One fundamental problem in social networks is the identification of groups of elements (users, posts or other elements) when group membership is not explicitly available. A group or community can be defined as a set of elements that interact more frequently or are more similar to other community members than to outsiders. Community detection has proven to be valuable in diverse domains such as biology, social sciences and bibliometrics. For example, community detection techniques can be used for identifying groups of users with similar purchase history on *Amazon* to create more efficient product recommendation systems, detecting topics in collaborative systems, identifying real-world landmarks in *Flickr* by clustering photos, detecting events on *Twitter* streams or for studying the information diffusion problem by solving the influence maximisation problem in *Foursquare*.

In the context of multidimensional networks, considering only one information source might be insufficient for accurately capturing community structure [40]. For example, in *Twitter*, social relations might be sparse, and users might belong to the same community even if there are no explicit friendship relations amongst them. Relations can also be noisy. As it is easier to connect with other users online than in the real world, users might have thousands of online friends. Hence, the correct identification of communities might be hindered if only friendship interactions are considered. Conversely, other users might have a few friends, but frequently engage in posting or commenting activities, which could reveal valuable information for discovering communities, despite the fact that social media content might be topically diverse and noisy. Thus, the integration of multiple information sources could help to overcome the problem caused by incomplete or noisy information in each dimension, as well as obtaining more accurate and reliable community partitions. Nonetheless, combining multiple and possibly heterogeneous data views poses new challenges; for example, how to fuse the different views for performing an integrated analysis.

Several community detection methods have been developed based on techniques from a variety of disciplines, such as statistical physics, biology, applied mathematics, computer science or sociology [44], mostly relying on similarity measurements amongst the nodes in the network [38], which might not be simple. Interestingly, most of them only focus on one data view. Moreover, although all methods aim at identifying meaningful communities, as they might rely on different notions of communities, their results might not be always directly comparable. In most real-world applications, a unique correspondence between nodes and communities (i.e. a ground truth) might not be available, which hinders the reliability assessment of community detection techniques. As a result, community detection algorithms are traditionally tested on a few real or artificial networks [30]. As real-world social networks are very heterogeneous, the question regarding over which data evaluate the algorithms remains open. On the other hand, artificial networks rely on various statistics like average degree, degree distribution and shortest path average, amongst others, which are not possible to control in a real environment. Hence, in both cases, algorithms are usually evaluated over networks with very specific and limited set of characteristics, which might not match the typical features of real-world networks. Whilst there are several works that have analysed the performance of diverse community detection algorithms over artificial graph benchmarks [12, 18, 30, 44], their evaluation over real social networks has received comparatively less attention [20].

Considering the increasing amount of available information in social networks, the necessity of integrating such heterogeneous information and the lack of studies analysing the problem of community detection over real-world networks, this chapter addresses three challenges. First, the definition and extraction of multiple sources of information regarding user interactions and activities that can be inferred from social media data. Second, the assessment of the performance of community detection algorithms in the context of two real-world social media networks. Third, the exploration and evaluation of diverse similarity measures that could be considered during the community detection process. To that end, it is also explored how to quantify the structural properties of the discovered communities in terms of several quality metrics. The final goal of this study is to provide some insights regarding the integration of diverse information sources and user interactions, as well as the selection of both algorithms and metrics for performing and assessing the community detection process.

The rest of this chapter is organised as follows. Section 12.2 discusses related research. Section 12.3 describes background concepts regarding the definition and extraction of graphs from social media sites, and presents the community detection techniques and the similarity metrics evaluated in this study. Section 12.4 describes the experimental evaluation performed over two real-world networks from *Twitter* and *Flickr*. Section 12.5 presents the observed results. Finally, Sect. 12.6 summarises the results and conclusions drawn from the analysis.

12.2 Related Work

According to graph theory [22], communities have also been defined as cliques (every node is adjacent to each other) or connected components (every pair of nodes is connected by at least a path). In this context, the goal of community detection techniques (also known as graph clustering techniques) is to divide the nodes into communities (or clusters), such that the nodes of a particular community are similar or connected in some predefined sense [38]. Several works have been dedicated to formalise the intuition that a community is a set of nodes that has more or better connections between its members than with the remainder of the network [20]. For example, in some cases it might be desirable to obtain communities of similar order and/or density. Interestingly, not every graph presents a structure with natural communities. In the case of a uniform graph structure in which edges are evenly distributed over the set of nodes, clustering results will be rather arbitrary.

Community detection techniques can be either local or global. Generally [31], the definition of global communities relies on the number of edges falling between them (cut size), the profoundness of their separation, modularity (i.e. the extent to which a given community partition deviates from the hypothetical state in which the network would be randomly rewired under the constraint of same-degree for each node [27]) or on the similarity between nodes. Community detection can be performed either by considering all data elements at once, or by iteratively assigning one element at a time to the appropriate cluster. Approaches that require the entire graph to be simultaneously accessible do not scale for large graphs [38].

On the other hand, local community detection techniques provide an alternative to alleviate scalability challenges of global techniques as they only focus on a portion of the network under study. Thus, they are expected to circumvent the memory bottleneck faced by global methods. Since it is not feasible to study the community structure as a whole in terms of space and computational complexity, communities can be progressively discovered by means of the explicit or implicit relations defined between the nodes. This type of technique starts the network exploration process from a set of seed nodes and progressively adds adjacent nodes to the community as long as those node additions lead to the increment of some local community quality measure [31].

In general, local techniques present limitations that need to be addressed in order to be effectively applied on large-scale social networks. First, the performance of local techniques is affected by the density (the number of links of the interconnected communities and the total number of links of the network) and the size of networks [29]. In this regard, techniques based on optimising modularity might fail to identify communities that are smaller than a size that depends on the number of nodes in the network and the link density of communities, even when communities are unambiguously discovered [13]. Fortunato and Barthélemy [13] found that the detection of communities based on modularity is not consistent with the modularity optimisation, which might favour network partitions in large communities. According to the authors, by enforcing modularity optimisation, the different possible partitions of the network are explored at a coarse level, so that communities that are smaller than a determined scale might not be resolved. The origin of the resolution scale lies in the fact that modularity is a sum of terms, where each term corresponds to a community. Thus, finding the maximal modularity is equivalent to look for the ideal trade-off between the number of terms in the sum. An increment in the number of communities does not necessarily imply an increment in modularity, as communities would be smaller so each term of the sum would also be smaller. Furthermore, modularity optimisation results in communities with similar sizes, which causes modularity to have a peak. The problem is that the supposedly optimal partition imposed by mathematics does not necessarily capture the actual community structure of the network, in which communities might have heterogeneous sizes. As a result, alternative measures for analysing community partitions have to be devised.

The remainder of this section presents relevant works that have aimed at defining benchmarks for comparing the performance of community detection techniques (Sect. 12.2.1), and metrics for comparing the performance of community detection techniques (Sect. 12.2.2).

12.2.1 Benchmarks for Community Detection

For newly designed techniques, it is necessary to assess their performance and compare it with that of other techniques. Nonetheless, the evaluation of techniques has received little attention in the literature [12]. As a result, it might be difficult to determine which technique is most reliable in the context of a certain domain or application. Generally, evaluations consist in applying the new techniques to a small set of simple benchmark graphs, whose community structure is known or easy to recover; for example, the social network of Zachary's karate club [45], the social network of bottlenose dolphins living in Doubtful Sound [21] or the American college football teams [14]. Zachary's karate club is one of the most used graphs and comprises two communities. In the American college football team graph, there are 115 nodes representing the teams, which are connected if they have played against each other. The natural partition of the graph comprises 12 communities, each representing a geographical area. Note that both networks are undirected and non-overlapping, as it is very difficult to find directed graph datasets with known community structures and sufficient size [22]. When considering real networks, it is worth noting that there is no guarantee that meaningful communities, defined on the basis of non-structural information, will match those detected by methods solely based on graph structure [12]. Moreover, in most cases, graph datasets are of small scale, which hinders their usefulness for assessing the performance of techniques at large scales. Thereby, it is crucial that the scientific community agrees on a standard evaluation procedure. In this context, several works have focused on the design of artificial benchmark graphs.

Condon and Karp [6] proposed one of the first graph benchmarks based on the planted ℓ-partition model with $n = g \cdot \ell$ nodes divided in ℓ communities with g nodes each. In this model, nodes of the same community are connected with a probability p_{in}, whilst nodes belonging to different communities are connected with a probability p_{out}. In this regard, each community represents a random Erdös-Rengy graph with a connection probability $p = p_{\text{in}}$. The modelled graph will have a community structure when the intracluster edge density is higher than the intercluster edge density, i.e. $p_{\text{in}} > p_{\text{out}}$. Following this model, Girvan and Newman [14] introduced one of the most known benchmarks, which is parametrised so that each network has 128 nodes divided into 4 groups, implying that p_{in} and p_{out} are not independent [12].

Although the ℓ-partition model is widely used, all nodes have approximately the same degree, and all communities have the same size by construction. These two features might not reflect the characteristics of real networks, in which degree distributions might be skewed with many nodes with low degree coexisting with a few nodes with high degree (for example, in online social networks). Brandes et al. [3] proposed a modification to the model named Gaussian random partition generator in which community sizes have a Gaussian distribution. This variation on community sizes also introduces heterogeneity in the degree distribution, as the expected degree of a node will depend on the number of nodes in its community. However, the introduced variations might still not be enough to represent real networks. In all cases, the hypothesis that the connecting probabilities of each node with the other nodes in the community or even with other communities are constant might not represent the actual properties of networks.

Lancichinetti et al. [19] proposed the LFR benchmark, in which the distributions of degree and community sizes are assumed to be governed by independent power laws. Graphs are built as follows. First, community sizes are defined based on a predefined power law distribution. Second, each node in a community is assigned a degree, which is defined considering another predefined power law. Third, all stubs of vertices of the same community are randomly connected to each other to maintain the predefined degree distribution. Fourth, each node is connected to nodes in the other communities.

It is worth noting that the described benchmarks correspond to undirected networks. Although the problem of detecting communities on directed networks has received comparatively less attention than on undirected networks, several benchmarks have been introduced to deal with special types of graphs and community structures. For example, Arenas et al. [1] extended Girvan and Newman's [14] benchmark to build graphs with embedded hierarchical structures. Additionally, Lancichinetti and Fortunato [18] proposed a modification of the LFR benchmark to create weighted, directed, and unweighted and overlapping networks. The weighted graph is built based on an unweighted graph by assigning positive real numbers to each edge. To that end, two new parameters are defined. The first parameter is used to assign a strength to each node such that the power law relation between the strength and the node degree is frequently observed in real networks. The second parameter is used to assign the internal strength, which is defined as the sum of the weights of the connections between a node and all its neighbours belonging to the same community. Then, a greedy algorithm is applied to the graph so that weights are consistent with the connection probabilities. For directed networks, changes are imposed in the degree definition. Whilst in-degrees are defined based on a power law, out-degrees are defined based on a δ-distribution. Finally, connections are established by preserving both distributions.

In the case of unweighted and overlapping graphs, a new topological mixing parameter is introduced to define the number of neighbours of a node that have at least one membership in common. The generating procedure is equivalent to the generation of a bipartite network where the two classes are the communities, and nodes comply to the requirement that the sum of community sizes equals the sum of node memberships. Although this benchmark aims at simulating the features observed in real-world networks, the requirement that overlapping nodes interact with the same number of embedded communities might be unrealistic [42]. Simple generalisations were proposed in which each overlapping node might belong to different numbers of communities [25], or communities are converted to fuzzy associations by adding a belonging coefficient to the occurrence of nodes [15]. Similarly, Sawardecker et al. [37] also proposed an extension of Girvan and Newman [14]'s benchmark in which the probability of an edge to be present in the network is a non-decreasing function based on the set of co-memberships of its vertices.

12.2.2 Comparing Community Detection Techniques

In addition to considering graphs with a known community structure, the quality of the communities discovered by a technique should be compared to that of other techniques, aiming at selecting the most accurate one. This implies the selection of criteria depending on the known structure of the network to define how similar the discovered communities are. Several metrics have been used for this purpose, which can be divided into three categories [12]: metrics based on pair counting, community matching or information theory. Metrics based on pair counting depend on the number of pairs of nodes that are assigned to the same or different communities. Similarity metrics based on community matching aim at finding the largest overlap between pairs of communities belonging to different partitions. A common problem of this type of metrics is that some communities might not be considered if the overlap with other communities is not large enough. The third category is based on casting the problem of comparing communities as the problem of message decoding in the context of information theory. The rationale of these metrics is that if two community partitions are similar, little information is needed to infer one partition, given the other. Thus, such extra information can be used as a measure of dissimilarity. Evaluating the quality of the discovered communities is non-trivial. The problem worsens when analysing overlapping communities [42], as extending the evaluation metrics from disjoint to overlapping communities is rarely straightforward.

Generally, there are two criteria to analyse the goodness of a community partition [20]. First, the number of edges or links amongst the nodes in each community, and second, the number of edges amongst the members of each community in relation to the nodes outside of it. These criteria characterise the connectivity structure of a given community, built on the assumption that communities comprise sets of nodes with many inner connections and few outer connections. Table 12.1 presents the definitions of the used metrics, where S represents the set of nodes of a community, E represents the set of edges of a community, n_S is the number of nodes in community S, c_S is the number of edges in the boundary of the community, i.e. $c_S = |\{(u, v) : u \in S, v \notin S\}|$, m_S is the number of edges inside the community, i.e. $m_S = |\{(u, v) : u, v \in S\}|$, $d(u)$ is the degree of node u, σ_{sw} is the number of shortest paths from node s to w and $\sigma_{sw}(v)$ is the number of shortest paths from

Table 12.1 Community quality measures

Cut ratio (↓)	Measures the fraction of existing edges (out of all possible edges) leaving the community. $\frac{c_S}{n_S(n-n_S)}$
Triangle separation ratio (TPR) (↑)	Measures the fraction of nodes inside a community that belong to a triad. $1/n_S *$ $\|\{u : u \in S, \{(v,w) : v, w \in S, (v,w) \in E, (u,w) \in E, (v,w) \in E\} \neq \emptyset\}\|$
Conductance (↓)	Measures the fraction of total edge volume that points outside the community. $\frac{c_S}{2 * m_S + c_S}$
Flake out degree fraction (FlakeODF) (↓)	Measures the fraction of nodes in the community that has fewer edges pointing inside the community than outside. $1/n_S * \|\{u : u \in S\}, \|\{(u,v) \in E, v \in S\}\| < d(u)/2\|$
Betweenness centrality (↑)	Measures how often a node appears on shortest paths between nodes in the community. $1/n_S * \sum_{u,v,w \in S, u \neq v \neq w} \frac{\sigma_{sw}(v)}{\sigma_{sw}}$
Closeness centrality (↑)	Measures the average distance between every pair of nodes in the community. $1/n_S * \sum_{u,v \in S, u \neq v} \frac{1}{\text{distance}(u,v)}$
Eccentricity (↓)	Averages the distance from each node to the farthest node in the community. $1/n_S * \sum_{u \in S} \max\{\text{distance}(u,v) : v \in S\}$
Density (↑)	Measures the fraction of edges (out of all possible edges) that appear between the nodes in the community. It is based on the supposition that good communities are well connected. $\frac{2 * m_S}{n_S(n_S - 1)}$
Clustering coefficient (↑)	It is based on the supposition that communities are manifestations of locally inhomogeneous distributions of edges as pairs of nodes with common neighbours are more likely to be connected with each other. $1/n_S * \sum_{u \in S} \frac{\|(v,w) \in E : v, w \in S, (u,v) \in E \wedge (u,w) \in E\|}{d(u) * (d(u) - 1)}$
Separability (↑)	Measures the ratio between the internal and external number of edges in the community. It is based on the supposition that good communities are well-separated from the rest of the network, i.e. communities have relatively few edges pointing to other communities. $\frac{m_S}{c_S}$

node s to w that pass through v. The arrows indicate whether a higher (↑) or a lower (↓) score is preferable.

Considering the described types of metrics, Lancichinetti and Fortunato [18] analysed the performance of 12 algorithms in the context of undirected and unweighted, and directed and weighted networks. Performance was evaluated in terms of computational time and normalised mutual information. Yang et al. [44] evaluated 8 algorithms by quantifying their accuracy using complementary measures and their computing time. The authors aimed at studying the dependency between network size, computing time and the predicting power of techniques. Finally, closely related to this study, Leskovec et al. [20] evaluated community detection algorithms over real networks: a bipartite authors-papers networks from

DBLP,[1] the Enron e-mail network,[2] a co-authorship network from Arxiv[3] and a social network from Epinions.[4] Note that no social media data network was selected for the analysis. The authors aimed at understanding the biases in the communities discovered by the selected algorithms by analysing several objective functions.

12.3 Community Detection on Social Media

Considering the increasing amount of available information in social networks, the development of a large number of community detection techniques, the necessity of integrating heterogeneous data from multiple source, and the lack of studies analysing either the performance assessment of community detection techniques, or the problem of community detection over real-world networks, this chapter focuses on four aspects. First, the definition of a graph in the context of social media data (Sect. 12.3.1). Second, the analysis of the performance of several community detection techniques (presented in Sect. 12.3.2). Third, the exploration and evaluation of diverse similarity metrics that could be considered during the community detection process (defined in Sect. 12.3.3). Fourth, the quantification of the properties and characteristics of communities that determine the goodness of algorithms (presented in Sect. 12.3.4). The final goal of this study is to provide some insights regarding the integration of diverse information sources and user interactions for community detection, as well as the selection of both the algorithms and the metrics for assessing the community detection process.

12.3.1 Graph Derivation on Social Media

To apply a community detection algorithm, the information on which the underlying graph structure is going to be based on has to be defined. Multiple and diverse information sources can be extracted from social media data, and hence multiple graph structures can be defined. Nodes might not only represent real people but also other entities such as neighbourhoods, Web pages or tweets, amongst others, depending on the task at hand to perform [23]. Then, once communities are found, they can be integrated in diverse learning tasks such as topic detection, text classification or clustering, link prediction or even feature selection.

Most community detection techniques are purely based on explicit social relations; however, in the context of social media data, both the social relations between

[1] http://snap.stanford.edu/data/com-DBLP.html.

[2] http://snap.stanford.edu/data/email-Enron.html.

[3] http://snap.stanford.edu/data/cit-HepPh.html.

[4] http://snap.stanford.edu/data/soc-Epinions1.html.

users and the characteristics of the published content are important for improving the quality of the discovered communities [40]. Hence, besides the relations between posts derived from the actual social relations between their authors (i.e. two posts are socially connected if their authors are socially related), posts' content resemblance or common categories (in case they are available) could also help to establish relations between them. It is worth noting that social information and content-based relations offer complementary views of data, in this case, posts. Thus, no individual relation might be sufficient for accurately determining community memberships [39]. For example, social information might be sparse and noisy, whilst content-based information could be irrelevant or redundant. Hence, it is important to adequately combine the different types of relations for performing community detection in social networks.

Content-based relations could be used either to establish new relations between posts that are not socially related (named *Independent* graph derivation) or to reinforce the social relations already found amongst posts (named *Weighted* graph derivation). In the former case, social and content relations are assumed to be independent from each other, i.e. edges in the graph represent not only social links but also separated content ones. Hence, when considering both types of relations independently, two nodes might be connected even when there is no explicit social connection between them. In this graph derivation, the different relationships are integrated by adding their corresponding matrices, as $A_{Rels} = \sum_{i \in Rels} A_i$, where A_{Rels} represents the aggregated adjacency matrix, $Rels$ is the set of selected relationships and A_i are the adjacency matrices. Note that no differentiation is made between the social and content-based relationships.

On the *Weighted* derivation, the graph only includes edges representing the social relation between nodes, whose strength or relevance is given by the content features. Thus, in this case, the quality of the social ties between nodes depends on an adequate definition of the content-based features, which should allow to fully exploit the social media data information. The final adjacency matrix for this derivation can be defined as $A_{Rels} = A_{Social} \cdot \sum_{i \in Rels_W} A_i + \sum_{i \in \{Rels - Social - Rels_W\}} A_i$, where A_{Social} represents the adjacency matrix for the *Social* relation and $Rels_W$ the set of relationships chosen for weighting the *Social* relationship. Note that this graph derivation also allows the integration of independent relationships, as showed by the second term in the Equation.

By definition, all content-based relations are symmetric, i.e. they do not have directionality. However, the same does not necessarily apply to the social or friendship relations. For example, when considering the Follower/Followee relationship in *Twitter* or the social relations in *Instagram*, the fact that user A follows user B does not imply that user B also follows user A. Whilst the diverse social networks exhibit different reciprocity levels, most community detection techniques only leverage on undirected (and perhaps weighted) graphs. It is worth noting that developing community detection techniques for directed graphs might be a difficult task [12], and that several concepts that are theoretically well defined for undirected graphs have not been yet extended to directed ones [22].

12.3.2 *Algorithms to Compare*

Both global and local community detection algorithms were selected for the comparison, which are described as follows.

Cobweb It is an incremental clustering algorithm [11]. The goal is to learn from observations, in opposition to learning from examples. Cobweb incrementally organises observations into a classification tree, in which each node is labelled by a probabilistic concept that summarises the attribute-value distributions of the elements under such particular node. Search is guided by a heuristic measure called Category Utility, which can be regarded as a trade-off between the intraclass similarity and interclass dissimilarity of elements.

Edge Betweenness It is a hierarchical decomposition process in which edges are removed in the decreasing order of their Edge Betweenness score [14, 27]. This technique focuses on the least central edges that connect separated communities. Hence, communities are created by progressively removing edges from the original graph. The computational complexity of the algorithm is $\Theta\left(v \cdot e^2\right)$, where v represents the number of nodes in the graph and e the number of edges, which might make it impractical for large and dense graphs. One disadvantage of this algorithm is that it builds a full dendrogram and does not provide any guidance about where to cut it to obtain the final community partition, hence other measures are needed to find that optimal partition.

Expectation Maximisation (EM) It is an iterative algorithm that alternates between two steps, expectation (E) and maximisation (M) [9]. When applied to clustering, expectation maximisation (EM) uses finite Gaussian mixture models and iteratively estimates a set of parameters until a desired convergence value is achieved. In the E step, for each instance it computes its membership possibility to each cluster based on the initial parameters. In the M step, parameters are recomputed based on the new membership possibilities. Theoretically, the running time is not bounded.

Farthest First It is a variant of the K-means algorithm that places each cluster centre aiming at maximising the cluster radius [16]. The algorithm operates in two steps: centroid selection and cluster assignment. The centroid selection step begins by selecting a random data point as the original cluster centre. Then, it iteratively chooses the next centres as the data points that are farthest from the previously selected one, until the desired number of centroids has been selected. In the cluster assignment step, all other data points are assigned to the nearest centroid. The computational complexity of this algorithm is $\Theta\left(v \cdot k\right)$, where k represents the number of desired clusters, making it suitable for large-scale applications.

Fast Greedy It is a bottom-up hierarchical approach that tries to optimise modularity in a greedy manner [5]. At first, each node belongs to a separated community. Then, communities are iteratively merged such that each merge is locally optimal. The merges stop when it is infeasible to increase modularity. The computational complexity of the algorithm is $\Theta\left(e \cdot d \cdot \log\left(v\right)\right)$, where d represents the depth of the dendrogram describing the community structure. This algorithm faces the resolution limit problem, caused by inefficiencies derived from merging communities with unbalanced sizes [41].

Infomap This algorithm optimises the map equation [36] exploiting the information-theoretic duality between the problem of data compression, and the problem of how to extract significant patterns or structures from such data. The map equation specifies the theoretical limit of the trajectories of a random walker on the network. Community structure is represented through a two-level nomenclature based on Huffman coding. The rationale behind this optimisation is that on partitions containing few intercommunity paths, the walker will probably stay longer inside the communities, leading to compact representations. The computational complexity of the algorithm is $\Theta\left(e\right)$.

Label Propagation In this algorithm [34], each node is assigned one of k labels. Then, labels are iteratively reassigned such that nodes take the most frequent label of its neighbours synchronously. The process is repeated until each node is labelled with the most frequent label in its neighbourhood. This algorithm is solely based on network structure and does not require neither optimisation of an objective function nor prior information about the communities. Although it is a fast technique, it provides unstable results, as they depend on the initial label configuration and the random decision of breaking ties. The computational complexity of the algorithm is $\Theta\left(v + e\right)$.

Leading Eigenvector It is a top-down hierarchical approach that optimises modularity in terms of a matrix eigenspectrum [26], leading to a centrality measure that identifies those vertices that occupy central positions within the communities to which they belong. The algorithm starts by computing the leading eigenvector of the modularity matrix. Then, it iteratively splits the graph into two parts to maximise the modularity improvement based on the leading eigenvector and stops once the modularity of the network subdivision is negative. The running time of the algorithm is $\Theta\left(v^2 + v \cdot e\right)$, or $\Theta\left(v^2\right)$ for sparse graphs.

Louvain Algorithm It is one of the most known and easy to implement algorithms, based on a greedy optimisation of modularity [2]. The algorithm is divided into two steps that are iteratively performed until there are no more changes. First, each node in the graph is assigned to a different community. For each node, it is moved to the community for which the positive modularity gain is maximum. This process is sequentially and repeatedly applied until modularity cannot be improved. The second step builds a new network whose nodes are the communities found during the first step. By definition, the number of communities decreases at each pass, thus most of the running time is concentrated on the first iterations. The computational time of the algorithm is $\Theta\left(v \cdot \log\left(v\right)\right)$.

Spinglass This algorithm aims at finding communities via a Spinglass model and simulated annealing based on statistical physics [35]. In this model, each particle (i.e. node) can be in one of c spin states (i.e. the maximum number of communities), and the interactions amongst particles (i.e. edges between nodes) define which nodes might stay on the same state, and which are likely to have different states. Then, the model is simulated and the final spin states of particles define the final community partition. Note that spin states could remain empty, hence reducing the number of found communities. Due to the nature of simulations, the algorithm is not deterministic, but it has parameters that allow determining the sizes of the communities to be found. The computational complexity of the algorithm for sparse graphs is approximately $\Theta\left(v^{3.2}\right)$.

Walktrap Similarly to Infomap, this algorithm is based on random walks [33]. The rationale is that short distance random walks tend to stay in the same community as they are assumed to have few edges outside them. At first, all nodes belong to different communities, and the distances between all adjacent nodes are computed. Then, two adjacent communities are iteratively chosen and merged, updating the corresponding distances. The output of the algorithm is a full dendrogram. The computational complexity of the algorithm is $\Theta\left(v^2 \cdot e\right)$, and $\Theta\left(v^2 \cdot \log(v)\right)$ for sparse graphs.

X-Means This algorithm aims at overcoming three K-means shortcomings [32]: the scaling inefficiency, the manual definition of the number of clusters, and the proneness to remain on local minima. It works as K-means until all instances have been assigned to their corresponding community or cluster. Then, it attempts to split each community into two separate communities by computing the Bayesian information criterion (BIC) on both the original community and the two newly created ones. In those cases the BIC for the new communities is higher than that for the previously defined communities, the new communities are retained and the total number of communities is increased. This process is iteratively repeated until convergence. Once all K values are tested, the best community partition is chosen.

12.3.3 Vertex Similarity

Most community detection techniques are based on computing the similarity amongst nodes [38]. For example, in a graph in which nodes represent posts, the similarities amongst them could be used to group together nodes that are not only well connected but also similar to each other. However, assessing node similarity might not be computationally simple, and even more complex than clustering the graph once all similarities are known.

The quality of the discovered communities might be greatly affected by the selection of the similarity metric. Hence, such metric has to be carefully chosen. Furthermore, as the different metrics assess node similarity from different points of view, they could be combined to perform a more comprehensive assessment. The Harmonic mean, which is less biased to the presence of outliers than the Arithmetic mean, adds a possibility for combining similarity scores. Note that for combining the scores, they must be normalised to the same range.

Metrics for computing similarity amongst nodes can be divided into three groups [38], which are summarised in Table 12.2:

Distance/Similarity Metrics Traditionally, these metrics are based on some distance property. Distance metrics should comply with three criteria: the distance between a node and itself is zero, distances are symmetrical and the triangle inequality holds. Similarity metrics resulting from the adaptation of distance metrics also comply with the three criteria. A great number of distance metrics have been defined and used in the literature [10, 17]. Examples of these metrics include Euclidean Distance, Manhattan Distance and Tanimoto Coefficient, amongst others.

Adjacency-Based Metrics In several environments, nodes lack of properties that allow computing their similarity [38]. The most straightforward manner for determining whether two nodes are similar only using adjacency information is to analyse the overlap of their neighbourhoods. Nonetheless, correlation analyses could also be applied to determine community structure based on adjacency information. Examples of these metrics include [10, 17]: Jaccard Similarity, Common Neighbours and Pearson Correlation, amongst others.

Connectivity Metrics Communities in graphs can also be defined through node connectivity by computing the number of paths between each pair of nodes [38]. In this regard, nodes belonging to the same community should be highly connected to each other. Connectivity metrics could be used to define the similarity amongst nodes. For example, nodes could be regarded as similar if they are connected by a number of paths higher than a predefined threshold, or similarity could be defined proportionally to the number of paths connecting the nodes. However, defining the threshold might be difficult, as its selection often involves knowing the diameter of the graph a priori. Choosing a large threshold in relation to the diameter of the graph might result in communities containing large portions of the graph, whereas choosing a small threshold might split natural communities into two or more subcommunities.

12.3.4 Quality Metrics

The criteria proposed by Leskovec et al. [20] (summarised in Table 12.1) characterise the connectivity structure of a given community built on the assumption that communities comprise sets of nodes with many inner connections and few outer connections. Nonetheless, connectivity structure might not be the only important characteristic of communities. In this regard, two additional functions were considered. First, a function characterising communities' content cohesiveness defined as the average Cosine Similarity amongst all node pairs in the community (named *ContentCohesiveness*). Second, as *Twitter* trending topics and *Flickr* photos

Table 12.2 Vertex similarity metrics

Statistic-based	Pearson Correlation (p_i, p_j)	$\frac{n\left(\sum_{k=1}^{n} A_{p_i,p_k} * A_{p_j,p_j}\right) - \lvert\Gamma(p_i)\rvert * \lvert\Gamma(p_j)\rvert}{\sqrt{\lvert\Gamma(p_i)\rvert * \lvert\Gamma(p_j)\rvert (n - \lvert\Gamma(p_i)\rvert) * (n - \lvert\Gamma(p_j)\rvert)}}$
Adjacency-based	$S\phi rensen\,(p_i, p_j)$	$\frac{2\lvert\Gamma(p_i) \cap \Gamma(p_j)\rvert}{\lvert\Gamma(p_i)\rvert + \lvert\Gamma(p_j)\rvert}$
Adjacency-based	$Leicht - Holme - Newman - LHN\,(p_i, p_j)$	$\frac{\lvert\Gamma(p_i) \cap \Gamma(p_j)\rvert}{\lvert\Gamma(p_i)\rvert \times \lvert\Gamma(p_j)\rvert}$
Adjacency-based	$Hub\,Promoted\,Index - HPI\,(p_i, p_j)$	$\frac{\lvert\Gamma(p_i) \cap \Gamma(p_j)\rvert}{\min\{\lvert\Gamma(p_i)\rvert, \lvert\Gamma(p_j)\rvert\}}$
Adjacency-based	$Hub\,Depressed\,Index - HDI\,(p_i, p_j)$	$\frac{\lvert\Gamma(p_i) \cap \Gamma(p_j)\rvert}{\max\{\lvert\Gamma(p_i)\rvert, \lvert\Gamma(p_j)\rvert\}}$
Adjacency-based	$Jaccard\,(p_i, p_j)$	$1 - \frac{\lvert\Gamma(p_i)\cap\Gamma(p_j)\rvert}{\lvert\Gamma(p_i)\cup\Gamma(p_j)\rvert} = \frac{\lvert\Gamma(p_i)\cup\Gamma(p_j)\rvert - \lvert\Gamma(p_i)\cap\Gamma(p_j)\rvert}{\lvert\Gamma(p_i)\cup\Gamma(p_j)\rvert}$
Distance-Similarity	$Euclidean\,(p_i, p_j)$	$\sqrt{\sum_{t=0}^{m} (p_{i,t} - p_{j,t})^2}$
Distance-Similarity	$Manhattan\,(p_i, p_j)$	$\sqrt{\sum_{t=0}^{m} \lvert p_{i,t} - p_{j,t}\rvert}$
Distance-Similarity	$Tanimoto\,(p_i, p_j)$	$\frac{p_i \cdot p_j}{\lVert p_i\rVert^2 + \lVert p_j\rVert^2 - p_i \cdot p_j}$
Distance-Similarity	$Cosine\,(p_i, p_j)$	$\frac{p_i \cdot p_j}{\lVert p_j\rVert \lVert p_j\rVert} = \frac{\sum_{t=0}^{m} (p_{i,t} \times p_{j,t})}{\sum_{t=0}^{m} p_{i,t}^2 \times \sum_{t=0}^{m} p_{j,t}^2}$

Where p_i and p_j denote the post for which the similarity score is computed, $\Gamma(p_i)$ denotes the set of neighbours of p_i, $\lvert\Gamma(p_i)\rvert$ denotes the degree of post p_i, A denotes the adjacency matrix, n denotes the number of posts contained in the graph and m denotes the dimension of posts, i.e. the total number of features contained in all posts are assigned to classes, the *Entropy* of the classes given the community assignments was also analysed. As scores were computed for each individual community, they were averaged to obtain the score corresponding to a given community partition. To ensure metrics' comparability, all results were normalised to the range [0; 1], and adjusted so that the highest scores represent the best ones.

12.4 Experimental Evaluation

This section presents the experimental evaluation performed to assess the effectiveness of the selected community detection algorithms over social media data, and is organised as follows. First, Sect. 12.4.1 describes the data collections used. Then, Sect. 12.4.2 describes the implementation details. Finally, Sect. 12.4.3 details the graphs derived from social media used to perform the evaluation. The final goal of this study is to provide some insights regarding the selection of both the algorithms and the metrics for performing and assessing the community detection process.

12.4.1 Dataset

The performance of the technique was evaluated considering two real-world datasets collected from *Twitter*[5] [46] and *Flickr*[6] [24]. Table 12.3 summarises the main characteristics of both datasets. The *Twitter* dataset includes the content of more than 500,000 tweets belonging to 1036 trending topics, which were manually assigned to one of four categories: News, Ongoing Events, Memes (trending topics triggered by viral ideas) and Commemoratives (the commemoration of a certain person or event that is being remembered in a given day, for example birthdays or memorials). For the purpose of the experimental evaluation, each trending topic was regarded as a node in the graph, i.e. each node grouped the tweet set associated to the corresponding trending topic.

The *Flickr* dataset comprises the metadata associated to original images from the NUS-WIDE dataset [4]. For each photo, the dataset included information regarding the owner, description, title, comments, tags, manually annotated labels and the groups in which the photo was posted. Labels were considered as the category of photos, and hence the community ground truth. Photos could be assigned to 81 different concepts (belonging to different categories such as, scene, object, event, program, people and graphics), which were extracted from frequently used tags, representing either general (e.g. "animal") or specific (e.g. "dog") concepts. Only photos containing at least one tag or description were kept. The dataset also provides

[5] http://www.twitter.com/

[6] http://snap.stanford.edu/data/web-flickr.html.

Table 12.3 Data collection's main characteristics

(a) Twitter data collection	
Number of instances	1036
Number of features	226,043
Number of classes	4
Number of following relations	251,522,840
Average number of followees	816
Average number of features per instance	1084
Average number of instances per class	259
(b) *Flickr* data collection	
Number of instances	190,339
Number of features	947,829
Number of classes	81
Number of taggers	58,144
Number of commenters	569,765
Pairs of photos posted by the same user	77,909
Pairs of photos posted by users who are friends	8,825,738
Average number of features per instance	5
Average number of instances per class	1007

information regarding the topological relations between the users and their photos, including an indicator for whether both photos were taken by the same user, and an indicator for whether two users were socially related. For the purpose of the experimental evaluation, each photo was considered as a node in the graph.

12.4.2 Experimental Settings

The performance of the selected community detection and clustering techniques was evaluated for different graph sizes ranging between 50 and 1000 posts. In order to make the results comparable, the implementations of the algorithms provided by three widely used libraries were used: Gephi Toolkit,[7] WEKA[8] and Igraph.[9] Both Gephi Toolkit and WEKA are implemented in Java, whilst Igraph is available in R, C++ and Python. For the purpose of this evaluation, the Python implementation was

[7]https://gephi.org.

[8]https://www.cs.waikato.ac.nz/ml/weka/.

[9]http://igraph.org/.

chosen. For Cobweb, EM, Farthest First and X-Means clustering algorithms, their WEKA implementation was used. Specifically, the density-based implementations were selected to evaluate the generated partitions in a cross-validated manner. The graph was represented in the *arff* format in which each node was considered an instance, and features also represented nodes. The value of each feature represented the weight of the edge between the corresponding nodes. On the other hand, for Edge Betweenness, Fast Greedy, Infomap, Label Propagation, Spinglass and Walktrap, it was used their Igraph implementation. Finally, as regards the Louvain algorithm, the Gephi implementation was used. The number of communities or clusters to detect was automatically discovered in all cases. For those algorithms returning a full dendrogram, the chosen community partition corresponded to the one maximising modularity.

12.4.3 Graph Creation

Evaluation was performed considering the *Social* relationship and the combinations of relationships obtaining the best results in [40], regarding the independent and weighted graph derivations, which are summarised in Tables 12.4 and 12.5 for the *Twitter* and *Flickr* datasets, respectively. Additionally, the table shows for each of the analysed relationships the average and standard deviation of the node's degrees for each of the graph sizes tested. As most algorithms are designed to work with undirected graphs, a symmetrisation strategy was applied to the directed graphs resulting from considering the asymmetrical *Social* relationship. Particularly, a simple symmetrisation in which the new adjacency matrix U can be defined as $U = A + A^T$ was applied. As a result, in the case a pair of nodes is connected with edges in both directions, the weight of the edge in the symmetrised graph will correspond to the sum of the weight of the directed edges.

12.5 Experimental Results

Interestingly, not every tested algorithm was able to find a meaningful number of communities (i.e. a number between 1 and the number of nodes) for each of the analysed relationships, hence those results are not reported. Considering the results obtained for each of the evaluated graph sizes, it was analysed whether the different samples achieved similar quality; in other words, whether the algorithms behaved stable across different graph sizes. Data normality was evaluated by performing both the Shapiro and the Anderson–Darling tests [7]. As data was shown not to be normal, the Kruskal–Wallis test for unrelated samples was applied to the results obtained for each metric and analysed combinations of relations. Particularly, each of the results obtained for a determined graph size was regarded as a sample. The confidence value was set to 0.01. To perform the test, the null and the alternative hypothesis were

Table 12.4 Evaluated combinations of social and content-based relationships for the *Twitter* dataset

	50 nodes	100 nodes	200 nodes	500 nodes	1000 nodes
Independent graph derivation					
Social	36.80 ± 8.21	79.94 ± 14.62	153.30 ± 36.27	383.74 ± 89.20	763.96 ± 179.79
SimilarContent-0.6	0.08 ± 0.27	0.06 ± 0.24	0.02 ± 0.14	0.10 ± 0.33	0.21 ± 0.55
SharedClass	18.88 ± 7.72	43.72 ± 21.13	80.20 ± 38.21	195.57 ± 94.09	407.47 ± 193.51
Social & *SimilarContent-0.6*	36.84 ± 8.22	79.94 ± 14.62	153.30 ± 36.27	383.74 ± 89.20	763.96 ± 179.79
Social & *SharedClass*	40.68 ± 6.20	87.82 ± 11.21	171.22 ± 26.02	425.15 ± 66.32	854.87 ± 129.11
Social & *SharedClass* & *SharedTag* & *SimilarContent-0.6*	42.08 ± 5.43	89.19 ± 10.16	176.00 ± 23.87	435.63 ± 61.77	875.18 ± 119.09
Social & *SharedClass* & *SharedTag* & *SimilarContent*	49.00 ± 0.00	98.00 ± 0.00	198.76 ± 1.24	496.44 ± 11.91	993.19 ± 16.88
Weighted graph derivation					
Social-W-SimilarContent-0.6	0.04 ± 0.20	0.06 ± 0.24	0.02 ± 0.14	0.09 ± 0.33	0.21 ± 0.55
Social-W-SharedClass	15.00 ± 7.24	35.84 ± 17.87	62.28 ± 32.41	154.15 ± 83.49	316.56 ± 168.83
Social-W-SimilarContent-0.6 & *SharedClass*	18.88 ± 7.72	43.74 ± 21.13	80.20 ± 38.21	0.09 ± 0.33	407.50 ± 193.49
Social-W-SharedClass & *SimilarContent-0.6*	15.04 ± 7.27	35.86 ± 17.88	62.28 ± 32.41	154.16 ± 83.49	316.59 ± 168.81
Social-W-SharedTag & *SharedClass*	27.24 ± 8.84	60.67 ± 19.61	125.82 ± 38.56	296.62 ± 94.60	606.40 ± 188.85

Table 12.5 Evaluated combinations of social and content-based relationships for the *Flickr* dataset

	50 nodes	100 nodes	200 nodes	500 nodes	1000 nodes
Independent graph derivation					
Social	29.00 ± 3.41	72.50 ± 8.51	122.25 ± 17.88	299.73 ± 17.88	371.42 ± 78.85
SimilarContent-0.6	16.62 ± 2.14	36.93 ± 4.51	73.06 ± 8.02	176.88 ± 8.02	349.88 ± 16.51
TaggedSameUser	57.59 ± 3.58	363.12 ± 34.31	418.04 ± 26.05	486.29 ± 26.05	618.71 ± 35.11
Social & SimilarContent-0.6	24.93 ± 3.22	55.40 ± 6.77	109.60 ± 12.03	265.34 ± 12.03	524.87 ± 24.75
Social & TaggedSameUser	90.13 ± 7.52	376.35 ± 33.66	475.95 ± 27.24	635.66 ± 27.24	857.08 ± 35.82
Weighted graph derivation					
Social-W-SharedClass & SimilarContent-0.6	19.94 ± 2.57	44.32 ± 5.42	87.67 ± 9.62	212.25 ± 9.62	419.86 ± 19.81
Social-W-SimilarContent & SimilarContent-0.6	79.65 ± 11.29	226.40 ± 50.50	271.67 ± 40.86	240.33 ± 40.86	489.21 ± 53.73
Social-W-TaggedSameUser & SimilarContent-0.6	16.62 ± 2.14	36.93 ± 4.51	73.06 ± 8.02	176.88 ± 8.02	349.88 ± 16.51
Social-W-CommentedSameUser & SimilarContent-0.6	18.28 ± 2.36	40.63 ± 4.96	80.37 ± 8.82	194.57 ± 8.82	384.87 ± 18.15
Social-W-SharedTag & SimilarContent-0.6	19.11 ± 2.47	42.47 ± 5.19	84.02 ± 9.22	203.41 ± 9.22	402.36 ± 18.98

defined. The null hypothesis stated that no difference existed amongst the results of the different samples, i.e. the alternatives behaved stable over the different graph sizes. On the contrary, the alternative hypothesis stated that changes in the size of the graph caused changes in the behaviour of the algorithms. In all cases, the null hypothesis could not be rejected, due to a p-value higher than the confidence value, or a critical value higher than the obtained statistic value. In consequence, it could be assumed that the algorithms did not change their behaviour as the size of the graph changed. Hence, for clarity of presentation, the results across different graph sizes are summarised by their mean value. The statistical invariability of results confirms the findings in [44] that stated that the performance of the algorithms on artificial networks is independent of the network size.

12.5.1 Evaluation of Quality Metrics

A correlation test was applied to the results obtained for all metrics to assess the relationship between them. For all the obtained community structures, the scores corresponding to all the metrics in Table 12.1 were computed. The correlation between the metrics' results was evaluated according to the definitions and methods proposed in [7]. The normality of results was evaluated by analysing their skewness, kurtosis, and performing both the Shapiro and the Anderson–Darling tests. As the normality tests failed for at least one result sample, correlation results had to be evaluated by means of non-parametric tests. Thus, correlation was measured by means of the Spearman Rank Order correlation. The confidence value for considering a correlation statistically significant was set to 0.05. The minimum correlation value for two metrics to be considered highly correlated was set to 0.7. Figure 12.1 summarises the obtained results for the *Twitter* dataset by depicting the significant relations found. Interestingly, although *TPR* is not highly correlated with any other metric (the highest correlation was 0.52 with the *ClusteringCoefficient*), the obtained results showed that the metric has no sufficient discriminative power amongst the different evaluated community detection alternatives. Particularly, its standard deviation was 0.27. Consequently, *TPR* was not considered for assessing the quality of communities. As the figure shows, metrics can be grouped into four clusters, out of which a representative metric can be chosen. These results are in agreement with those in [43], showing that although there are different quality measures for structurally assessing the quality of communities, such definitions are heavily correlated. Nonetheless, the results obtained for *TPR*, *CutRatio* and *Density* (i.e. the representative metrics for each group) did not show statistical differences amongst the results for the different combinations of alternatives. Consequently, results are only reported for *FlakeODF*. On the other hand, no correlation was found amongst *ContentCohesiveness* and *Entropy*. Similar results were observed for the *Flickr* dataset.

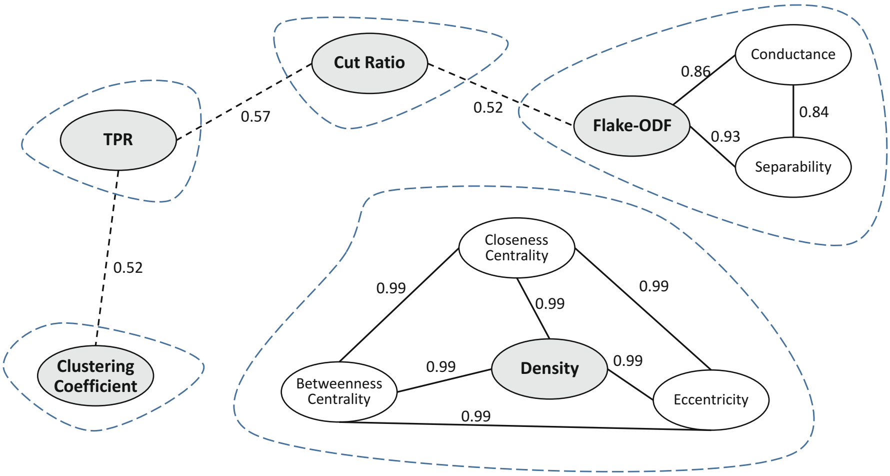

Fig. 12.1 Statistical correlation between quality assessment metrics

12.5.2 Evaluation of Community Detection Algorithms

This section presents the evaluation of the selected community detection algorithms for the *Twitter* (Sect. 12.5.2.1) and *Flickr* (Sect. 12.5.2.2) datasets. For both datasets, a statistical analysis was performed to determine whether the differences amongst results were statistically significant. As data was shown not to be normal, the Friedman test for related samples was applied to the results obtained for each community detection algorithm and combination of relations. Particularly, the results obtained for a determined node relationship across all community detection algorithms were regarded as a sample. To perform the test, two hypotheses were defined: the null and the alternative hypothesis. The null hypothesis stated that no difference existed amongst the results of the different samples, i.e. the discovered communities are independent from the algorithm used for discovering them, and the observed differences are due to chance. On the contrary, the alternative hypothesis stated that the observed differences amongst the different community structures are incidental, and not due to chance.

12.5.2.1 Results for the Twitter Dataset

The reported results include all the community detection algorithms previously described with the exception of Edge Betweenness as for each evaluated graph structure this algorithm required more than 2 h of execution, thereby being only suitable for small networks, and not for real-time processing applications. Additionally, the experimental evaluation showed that the algorithms that best scaled as the network size increased were Louvain and X-Means. On the other hand, Infomap, Walktrap

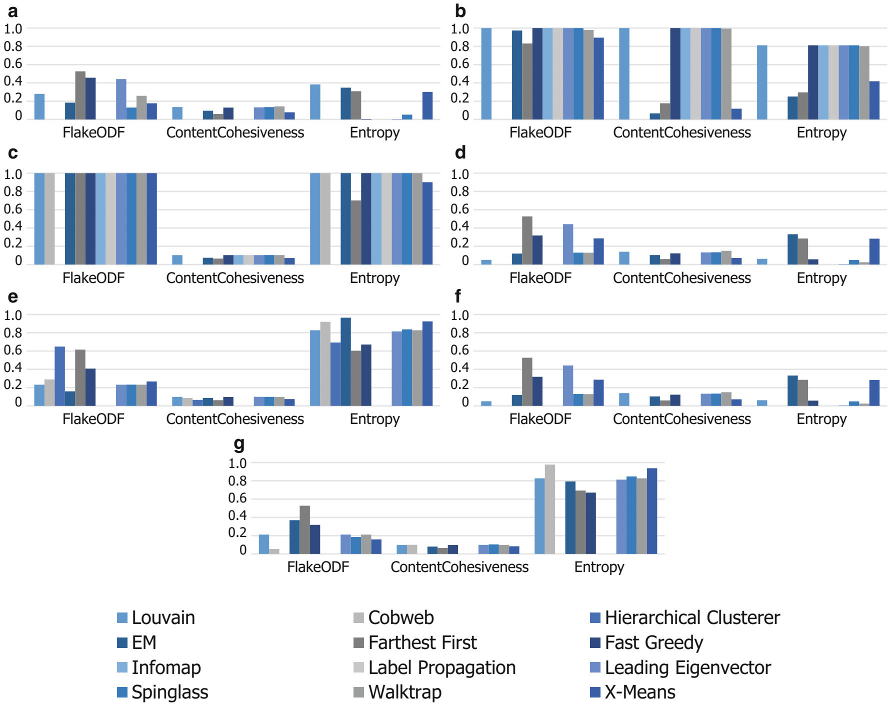

Fig. 12.2 Community detection results for the independent social and content views for the *Twitter* dataset. (**a**) *Social*. (**b**) *SimilarContent-0.6*. (**c**) *SharedClass*. (**d**) *Social & SimilarContent-0.6*. (**e**) *Social & SharedClass*. (**f**) *Social & SharedClass & SharedTag & SimilarContent-0.6*. (**g**) *Social & SharedClass & SharedTag & SimilarContent*

and Spinglass did not scale well, hindering their applicability on large networks or real-time applications. These findings are in agreement with those in [44].

Independent Social and Content Views Figure 12.2 shows the obtained results for the different combinations of node relationships for the independent derivation of the social graph. As it can be observed, there was no clear dominance of neither the traditional clustering techniques nor the algorithms specifically designed for community detection, as the quality of the obtained communities for each algorithm varied according to the node relationships under consideration.

The highest quality communities were found for high density networks. Moreover, most algorithms did not obtain stable results across the evaluated relationships. For example, Farthest First was amongst the best performing algorithms for *Social*, but also was the worst performing one for *SharedClass*. Similarly, Spinglass was one of the best performing algorithms for *SharedClass*, but the worst one for *Social*. These results show the instability and lack of robustness of algorithms when varying the underlying graph structure. Moreover, results exposed the sensitiveness (as shown in Table 12.4) of algorithms to the network degree, as described in [28]. For example, the quality of the communities found by Spinglass and X-Means increased as the average degree of the network increased. On the other hand, for Leading Eigenvector the effect was the inverse, i.e. the quality of the found communities decreased as the average node degree increased. Finally, Walktrap consistently obtained communities of lower quality than other algorithms, independently of network density. Label Propagation only obtained results for the combinations of relationship with the lowest average node degree.

Interestingly, in the case of *Social*, several community detection algorithms found community partitions with higher *ContentCohesiveness* than when considering *SharedClass*, even though explicit content information was not included. Moreover, those same algorithms obtained the highest *ContentCohesivess* for *SimilarContent-0.6*, with competitive *Entropy* and *FlakeODF* results. These differences were maintained when both relationships were combined. The obtained results might also expose the redundancy of relationships, as combining either two of four relationships obtained similar results for most algorithms. Particularly, results were similar for *Social* & *SharedClass* (combination of two relationships) and *Social* & *SharedClass* & *SharedTag* & *SimilarContent-0.6* (combination of four relationships). In those cases, all algorithms discovered partitions with low *Content-Cohesiveness* even though, the individual relationships allowed discovering content cohesive communities.

As regards the differences between the clustering and the community detection algorithms, a curious phenomenon appeared when analysing the results for *Social*, *SharedClass* and their combination. Whilst for *Social*, results did not show a clear dominance of any type of technique, for *SharedClass*, the best average results were obtained by the community detection techniques. However, when combining both relations the tendencies were reverted, and clustering techniques obtained the best results. This phenomenon emphasises the instability and sensitivity of algorithms.

Finally, it is worth noting that the Louvain algorithm was able to find relatively high-quality communities across almost every analysed individual and combinations of relationships, regardless of the density of the analysed graph. Particularly, the performed Wilcoxon test showed with a confidence of 0.01 the statistical superiority of Louvain regarding EM and Spinglass for the three quality metrics. When solely considering *FlakeODF* and *Entropy*, Louvain obtained statistically superior results than EM, Spinglass, Walktrap, Hierarchical Clusterer, Fast Greedy, Label Propagation, X-Means and Leading Eigenvector. These results showed the capabilities of the algorithm and its stability. Moreover, the results agree with those in [44], which stated the superiority of the communities found by the Louvain algorithm, especially for large graphs.

Weighted Social View Figure 12.3 shows the obtained results of the weighted derivation of the social graph for the different combinations of node relationships. Similarly to the results obtained for the independent graph derivation, there is no clear dominance of any type of technique, excepting for those relations considering *SimilarContent-0.6*, for which clustering algorithms consistently obtained the worst results. Interestingly, for this graph derivation, Label Propagation was able to

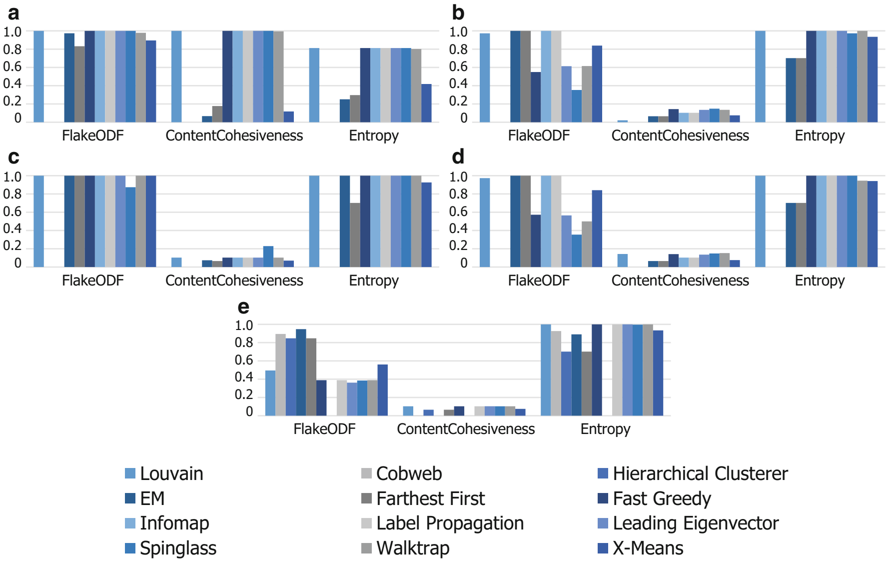

Fig. 12.3 Community detection results for the weighted social views for the *Twitter* dataset. (**a**) *Social-W-SimilarContent-0.6*. (**b**) *Social-W-SharedClass*. (**c**) *Social-W-SimilarContent-0.6 & SharedClass*. (**d**) *Social-W-SharedClass & SimilarContent-0.6*. (**e**) *Social-W-SharedTag & Shared-Class*

find a meaningful number of communities for every analysed relation. Its results were competitive regarding other techniques in all cases, but for the relationship yielding the highest node average degree. These results showed the sensitivity of the algorithm to the network degree distribution. As regard the diverse evaluated relations, results evidenced that techniques were dominated by the information provided by *SharedClass*, i.e. algorithms obtained almost equal results in every combination of relationships including *SharedClass*.

EM and Fast Greedy were shown to have contrasting results. Whilst EM was amongst the best performing techniques for *Social-W-SharedTag & SharedClass* and the worst one for *Social-W-SimilarContent-0.6*, Fast Greedy obtained exactly the reverse results. These results continued to expose the sensitivity of techniques towards the underlying graph distribution. Additionally, Farthest First was shown to improve the quality of the found communities as the density of the graph increased. Walktrap continued to exhibit poor performance for this graph derivation. Unlike for the independent graph derivation, the quality of the communities found by Spinglass decreased as the node average degree increased. The results for the Louvain algorithm continued to be the most stable ones across all relationships, confirming the robustness of the algorithm for finding high-quality communities. Finally, similarly as for the other graph derivation, a Wilcoxon test confirmed with a confidence of 0.01 the superiority of Louvain regarding other evaluated algorithms.

12.5.2.2 Results for the Flickr Dataset

Similarly as for the *Twitter* dataset, no results are reported for Edge Betweenness due to the excessive execution time. Additionally, no results are reported for Coweb as it did not allow to obtain a significant number of communities for any of the selected combinations of relationships. As for the *Twitter* dataset, the algorithms that best scaled as the network size increased were Louvain and X-Means. Moreover, unlike for the *Twitter* dataset, Label Propagation obtained a meaningful number of communities for every analysed combinations of relationships.

Independent Social and Content Views Figure 12.4 shows the obtained results for the different combinations of node relationships for the independent derivation of the social graph. As it can be observed, there was no clear dominance of neither the traditional clustering techniques nor the algorithms specifically designed for community detection, as the quality of the obtained communities for each algorithm varied according to the node relationships under consideration. The highest differences were observed for *ContentCohesiveness*, which achieved the lowest results for the *Social* relationship.

Similarly as for the *Twitter* dataset, the best results were found for high density networks. Nonetheless, none relationship or combination of them was able to obtain high results for the three metrics simultaneously. For example, *FlakeODF* and *Entropy* were high for *Social*, *TaggedSameUser* and *Social* & *TaggedSameUser*,

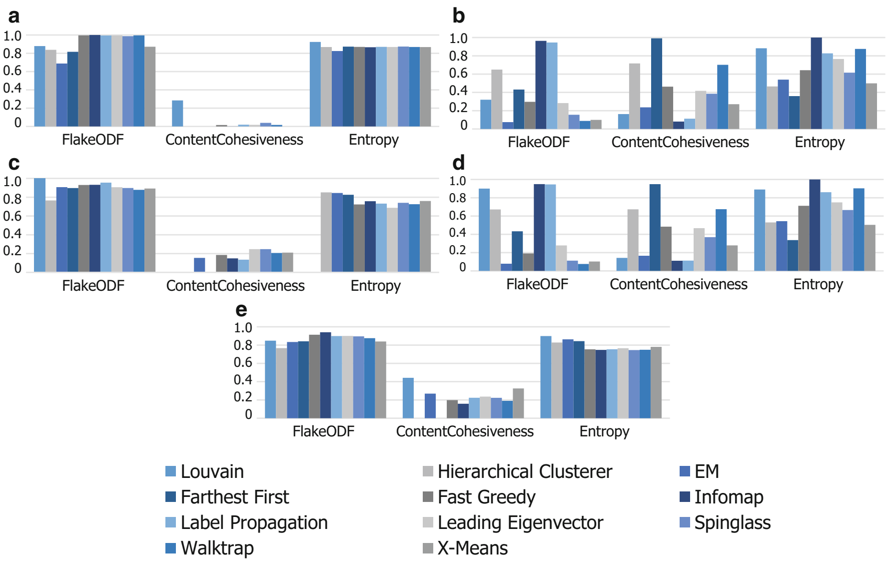

Fig. 12.4 Community detection results for the independent social and content views for the *Flickr* dataset. (**a**) *Social*. (**b**) *SimilarContent-0.6*. (**c**) *TaggedSameUser*. (**d**) *Social* & *SimilarContent-0.6*. (**e**) *Social* & *TaggedSameUser*

whilst *ContentCohesiveness* was high for *Social* & *SimilarContent-0.6*. Moreover, most algorithms did not obtain stable results across the evaluated relationships. For example, EM was amongst the best performing algorithms for *TaggedSameUser* and *Social* & *TaggedSameUser*, but also was the worst performing one for the other three combinations of relationships. Similarly, Infomap was the best performing algorithm for *SimilarContent-0.6* and *Social* & *SimilarContent-0.6*, but one of the worst for *Social* & *TaggedSameUser*. These results show the instability and lack of robustness of algorithms when varying the underlying graph structure. The sensitiveness of algorithms to network degree observed for the *Twitter* dataset was also observed for this dataset (as shown in Table 12.5); for example, the relative quality of the communities found by X-Means and EM increased as the average degree of the network increased. Interestingly, the quality of Spinglass results changed independently of the underlying node degree.

The obtained results might also expose the redundancy of relationships, as combining two relationships obtained similar results to those obtained for the individual relationships for several algorithms. For example, results were similar for *Social* and *Social* & *SimilarContent-0.6*, and for *TaggedSameUser* and *Social* & *TaggedSameUser*. These results might imply that the characteristics of the discovered communities are dominated by only one relationship of the pair.

As regards the differences between the clustering and the community detection algorithms, results did not show a clear dominance of any type of technique, as both community detection and clustering techniques achieved both high- and low-quality results. Nonetheless, in all cases the best results were obtained by community detection techniques (Infomap and Louvain), followed, in some cases, by clustering techniques, whilst the worst results were obtained by clustering techniques (EM and Hierarchical Clusterer). This phenomenon emphasises the instability and sensitivity of algorithms. Finally, it is worth noting that the Louvain algorithm obtained the best average results for four of the analysed individual and combinations of relationships, regardless of the density of the analysed graph. These results continue to show the capabilities of the algorithm and its robustness.

Weighted Social View Figure 12.5 shows the results obtained for the weighted derivation of the social graph for the different combinations of node relationships. Unlike the results observed for the independent graph derivation, the worst quality communities were consistently obtained by two clustering techniques X-Means and EM. On the other hand, the best results were obtained by community detection techniques in all cases.

For this graph derivation, the differences of average node degree were lower than for the independent derivation, excepting for *Social-W-SimilarContent* & *SimilarContent-0.6* (the relationship combination showing the highest average node degree), whose average degree was a 98% higher than the second highest one (*Social-W-SharedClass* & *SimilarContent-0.6*). In this regard, most of the techniques showed stable results for four of the five combinations of relationships analysed. Then, in the case of EM, Spinglass and X-Means, the quality of the

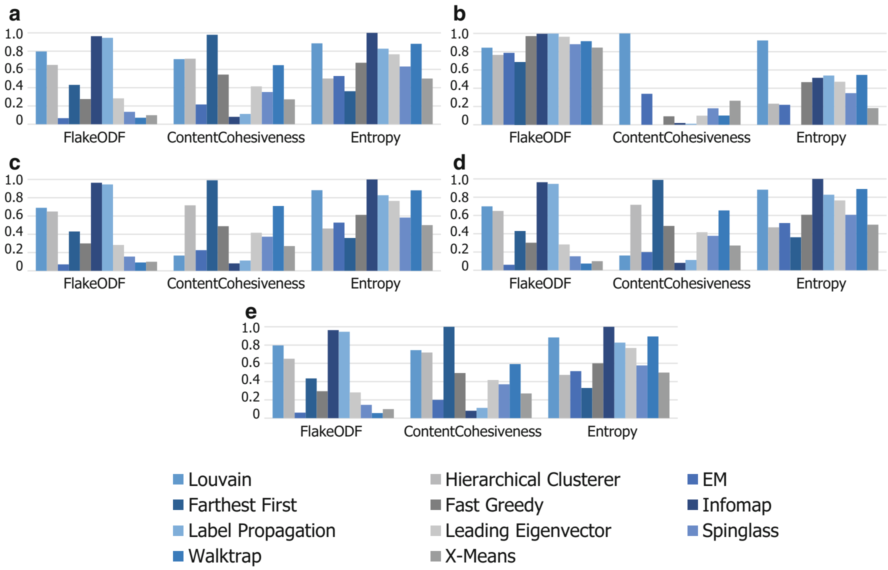

Fig. 12.5 Community detection results for the weighted social views for the *Flickr* dataset. (**a**) *Social-W-SharedClass* & *SimilarContent-0.6*. (**b**) *Social-W-SimilarContent* & *SimilarContent-0.6*. (**c**) *Social-W-TaggedSameUser* & *SharedClass*. (**d**) *Social-W-CommentedSameUser* & *SimilarContent-0.6*. (**e**) *Social-W-SharedTag* & *SimilarContent-0.6*

discovered communities increased when the average node degree increased, whilst the quality of Hierarchical Clusterer, Farthest First, Infomap and Label Propagation decreased.

The results for Louvain, Fast Greedy and Leading Eigenvector were the most stable ones across all relationships. These results confirm the robustness of Louvain for community detection on heterogeneous graph structures. Finally, similarly as for the independent graph derivation, a Wilcoxon test confirmed with a confidence of 0.05 the superiority of Louvain regarding other evaluated algorithms.

12.5.3 Evaluation of Vertex Similarity Metrics

This section presents the evaluation of the selected vertex similarity metrics for the *Twitter* (Sect. 12.5.3.1) and *Flickr* (Sect. 12.5.3.2) datasets. Similarly to the evaluation of the community detection techniques, the stability of the vertex similarity metrics across the different evaluated graph sizes was statistically analysed. As data was shown not to be normal, the Kruskal–Wallis test for unrelated samples was applied to the results obtained for each metric and each of the analysed combinations of relations. The results obtained for each graph size were regarded as a sample. The confidence value was set to 0.01. Results showed that the hypothesis that no difference existed amongst the results of the different samples could not be rejected, as either the p-value was higher than the set confidence value or the critical value was higher than the obtained statistic value. Hence, the diverse vertex similarity metrics could be assumed to behave stable across the different graph sizes. For clarity of presentation, the results across different graph sizes are summarised by their mean value.

Finally, a statistical analysis was performed to determine whether the differences amongst results were statistically significant. As data was shown not to be normal, the Friedman test for related samples was applied to the results obtained for each vertex similarity metric and combination of relations. Particularly, the results obtained for a determined node relationship across all vertex similarities were regarded as a sample. The null hypothesis stated that no difference existed amongst the results of the different samples, i.e. discovering communities based on the full node set yielded the same quality than iteratively extracting one node from the set and inserting it using vertex similarity metrics. On the contrary, the alternative hypothesis stated that inserting a node in an already computed community structure does not lead to the same community quality than using the full node set.

12.5.3.1 Results for the Twitter Dataset

As the following subsections show, no differences were observed between the results for the different graph derivations. This evaluation continued to expose the redundancy amongst relationships.

Independent Social and Content Views Figure 12.6 shows the obtained results for the different combinations of node relationships for the independent derivation of the social graph. As it can be observed, results obtained for each vertex similarity metric are similar to those obtained when considering all nodes in the community detection process (named *Full Communities*). Interestingly, differences are only observable after the fourth decimal place.

The biggest differences were found for *SharedClass* (Fig. 12.6c) and *SimilarContent-0.6* (Fig. 12.6b). When considering *SharedClass*, the communities discovered for the full set of nodes had higher *ContentCohesiveness* and *Entropy* than those obtained for the vertex similarity metrics. Conversely, in the case of *SimilarContent-0.6*, the *ContentCohesiveness* obtained for the full set of nodes was lower than that observed for the vertex similarity metrics. Note that the highest differences were observed for the lowest density graph. These results could imply that community structures are very sensitive to small changes in the node set. For example, it could occur that removing one node could alter the strength of links forcing the separation of a community into two or more communities. Then, when the node is inserted in the community structure, communities are not merged, altering their quality.

The results of *Social & SharedClass* (Fig. 12.6e), *Social & SharedClass & SharedTag & SimilarContent-0.6* (Fig. 12.6d) and *Social & SharedClass & SharedTag & SimilarContent* (Fig. 12.6g) are identical. These results reinforce the fact

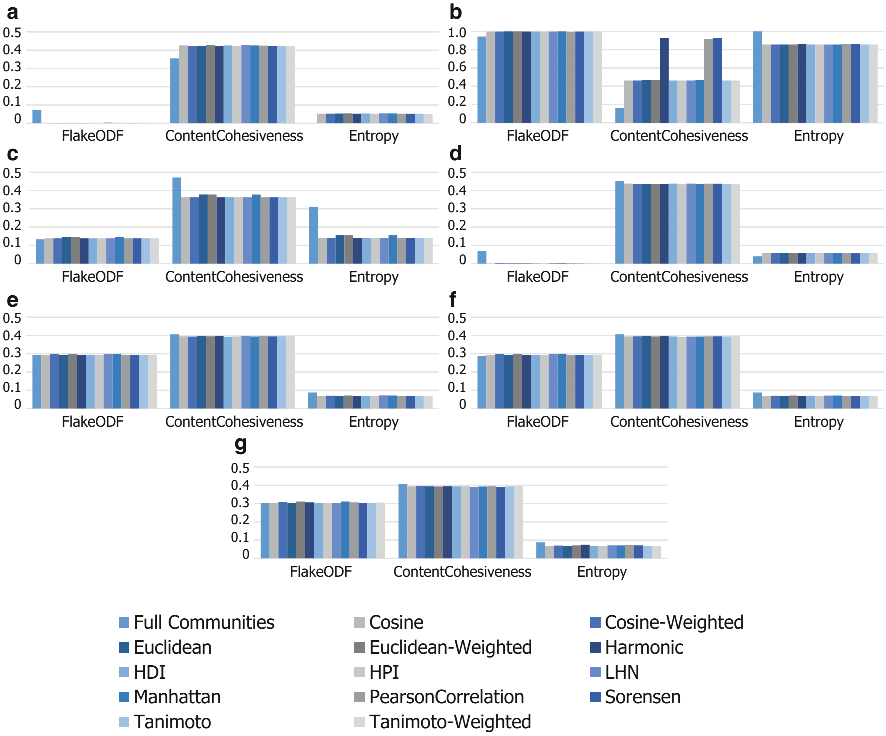

Fig. 12.6 Vertex similarity results for the independent social and content views for the *Twitter* dataset. (**a**) *Social*. (**b**) *SimilarContent-0.6*. (**c**) *SharedClass*. (**d**) *Social & SimilarContent-0.6*. (**e**) *Social & SharedClass*. (**f**) *Social & SharedClass & SharedTag & SimilarContent-0.6*. (**g**) *Social & SharedClass & SharedTag & SimilarContent*

that in some cases, adding more information does not imply an improvement of results. Instead, information sources might be redundant. As observed, for low-density graphs, $S\phi rensen$, $Pearson\ Correlation$ and their combination achieved higher results than the other metrics.

Table 12.6 summarises the results obtained for each vertex similarity metric averaged for every node relationship analysed. The best three results obtained for each metric are highlighted in bold. Interestingly, in average, considering the full node set achieved the lowest *ContentCohesiveness*, but the highest *Entropy* and *FlakeODF*. Note that $Harmonic$ obtained high results for the three evaluated metrics, showing the highest *ContentCohesivenness* values.

The performed Friedman test showed with a confidence of 1.229e−07 that the null hypothesis should be rejected, meaning that there is a difference between any of the analysed pair of results. To discover the pairs for which a statistical difference existed, the Wilcoxon test was applied defining the same hypotheses. Wilcoxon results showed with a confidence of 0.05 the existence of significant differences amongst the diverse vertex similarity metrics. For example, $Cosine\ Similarity$

Table 12.6 Summary of vertex similarity results for the independent graph derivation of the *Twitter* dataset

	FlakeODF	*ContentCohesiveness*	*Entropy*
Full Communities	**0.30**	0.38	**0.23**
Cosine	0.29	0.41	0.19
Cosine-Weighted	0.29	0.41	0.19
Euclidean	0.29	0.41	0.19
Euclidean-Weighted	0.29	0.41	0.19
Harmonic	**0.31**	**0.48**	**0.20**
HDI	0.29	0.41	0.19
HPI	0.29	0.41	0.19
LHN	0.29	0.41	0.19
Manhattan-Weighted	0.29	0.41	0.19
PearsonCorrelation	0.30	**0.48**	**0.20**
Sorensen	**0.31**	**0.48**	0.19
Tanimoto	0.29	0.41	0.19
Tanimoto-Weighted	0.29	0.41	0.19

results were shown to be statistically different and lower than most of those of the other metrics. On the other hand, *Pearson Correlation* and *Harmonic* were shown to be statistically superior to the other metrics. Finally, no statistical difference was found amongst the binary and weighted variations of the metrics.

Weighted Social View Figure 12.7 shows the obtained results for the different combinations of node relationships for the weighted derivation of the social graph. As it can be observed, the behaviour of the vertex similarity metrics did not change regarding the other graph derivation strategy. For three combinations of relationships, considering the full node set allowed to obtain higher *Entropy*, and a slight improvement of *ContentCohesiveness*. These results also exposed the redundancy amongst node relationships as *Social-W-SharedClass* (Fig. 12.7b), *Social-W-SimilarContent-0*.6 & *SharedClass* (Fig. 12.7c), *Social-W-SharedClass* & *SimilarContent-0.6* (Fig. 12.9b) and *Social-W-SharedTag* & *SharedClass* (Fig. 12.7e) achieved similar results.

Table 12.7 summarises the results obtained for each vertex similarity metric averaged for every node relationship analysed. The best results obtained for each metric are in bold. Interestingly, in average, considering the full node set achieved the lowest *ContentCohesiveness* but the highest *Entropy*. Unlike for the other graph derivation, using the full node set did not discover the most structurally cohesive communities. Similarly to the previous case, *Harmonic* was amongst the best performing vertex similarity metrics.

The same statistical analyses performed for the independent derivation results were performed for this graph derivation. The Friedman test showed with a confidence of $1.049e{-}11$ that the null hypothesis should be rejected, meaning that there was a difference between any of the analysed pair of results. Then, the

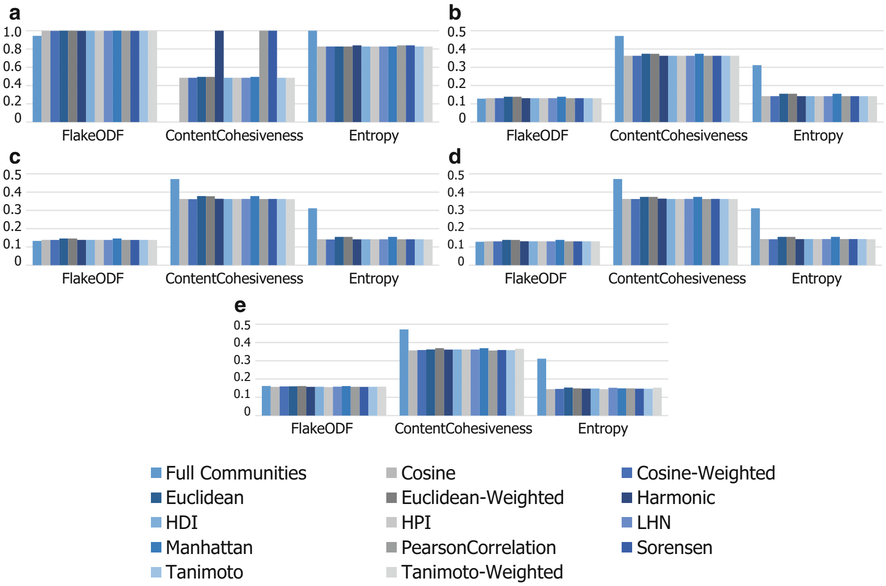

Fig. 12.7 Vertex similarity results for the weighted social views for the *Twitter* dataset. (**a**) *Social-W-SimilarContent-0.6*. (**b**) *Social-W-SharedClass*. (**c**) *Social-W-SimilarContent-0.6 & SharedClass*. (**d**) *Social-W-SharedClass & SimilarContent-0.6*. (**e**) *Social-W-SharedTag & SharedClass*

Table 12.7 Summary of vertex similarity results for the weighted graph derivation of the *Twitter* dataset

	FlakeODF	*ContentCohesiveness*	*Entropy*
Full Communities	0.30	0.38	**0.45**
Cosine	0.31	0.39	0.28
Cosine-Weighted	0.31	0.39	0.28
Euclidean	**0.32**	**0.40**	0.29
Euclidean-Weighted	**0.32**	**0.40**	0.29
Harmonic	**0.33**	**0.49**	**0.30**
HDI	0.31	0.39	0.28
HPI	0.31	0.39	0.28
LHN	0.31	0.39	0.28
Manhattan-Weighted	**0.32**	**0.40**	0.29
PearsonCorrelation	0.31	**0.49**	**0.30**
Sorensen	0.31	**0.49**	**0.30**
Tanimoto	0.31	0.39	0.28
Tanimoto-Weighted	0.31	0.39	0.28

Wilcoxon test was applied. Results showed with a confidence of 0.05 the existence of significant differences amongst the diverse vertex similarity metrics. For example, *Cosine Similarity* results were shown to be statistically different and lower than most of those of the other metrics, whilst *Pearson Correlation* and *Harmonic* were shown to be statistically superior to the other metrics.

12.5.3.2 Results for the Flickr Dataset

As the following subsections show, the results for this dataset present some differences regarding those observed for the *Twitter* dataset. Such differences could be due to the intrinsic and particular characteristics of each social network under analysis. Additionally, some differences were observed between the analysed graph derivations.

Independent Social and Content Views Figure 12.8 shows the obtained results for the different combinations of node relationships for the independent derivation of the social graph. As it can be observed, in terms of *FlakeODF* results obtained for each vertex similarity metric are similar to those obtained for *Full Communities*. In most cases, differences are only observed after the third decimal place. Discov-

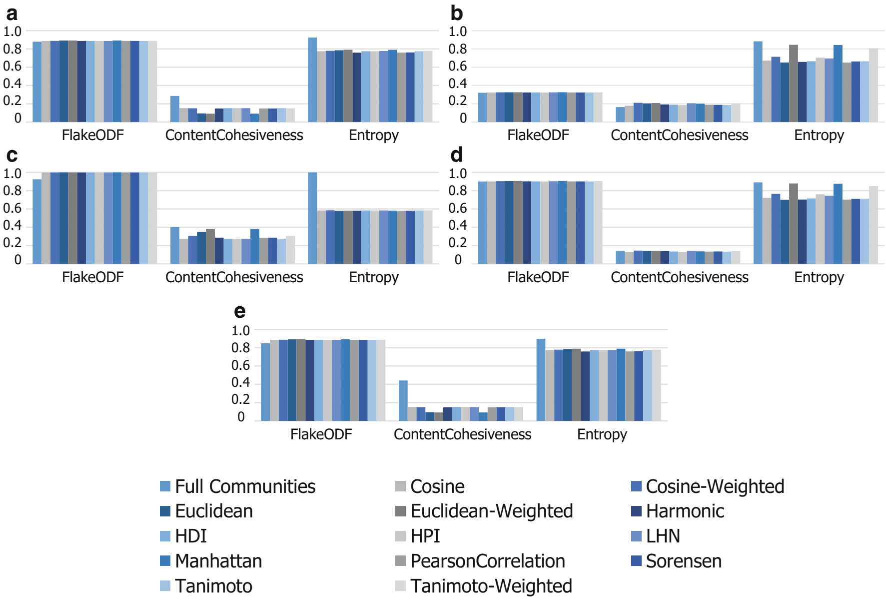

Fig. 12.8 Vertex similarity results for the independent social and content views for the *Flickr* dataset. (**a**) *Social*. (**b**) *SimilarContent-0.6*. (**c**) *TaggedSameUser*. (**d**) *Social* & *SimilarContent-0.6*. (**e**) *Social* & *TaggedSameUser*

ering communities for the full node set obtained the best average results for every combination of relationships, excepting *SimilarContent-0.6*.

In terms of *Entropy*, the biggest differences were observed for *SimilarContent-0.6* (Fig. 12.8b) and *Social* & *SimilarContent-0.6* (Fig. 12.8d). In those cases, the best average results were obtained when considering *Full Communities*, followed by the results of *Euclidean-Weighted*, *Manhattan-Weighted* and *Tanimoto-Weighted*. As observed for the community detection algorithms, the results for those combinations of relationships are similar, which might imply that the characteristics of the discovered communities are dominated by only one relationship of the pair. For the remaining three combinations of relationships, *Full Communities* obtained the highest *Entropy* results, with differences up to a 71% regarding the different similarity metrics. As regards *ContentCohesiveness*, for those alternatives considering *SimilarContent-0.6*, *Full Communities* did not discover the highest quality communities.

Table 12.8 summarises the results obtained for each vertex similarity metric averaged for every node relationship analysed. The best results obtained for each metric are highlighted in bold. Note that for *FlakeODF* average results were almost equal for every analysed similarity metric. Interestingly, in average, considering *Full Communities* achieved the lowest *FlakeODF*, but the highest *ContentCohesiveness* and *Entropy*. These results differ from those observed for the *Twitter* dataset, in which *FullCommunities* achieved the lowest *ContentCohesiveness* and the highest *FlakeODF*. Moreover, unlike for the *Twitter* dataset, the weighted versions of *Cosine*, *Manhattan* and *Euclidean* obtained higher results than *Harmonic*.

Table 12.8 Summary of vertex similarity results for the independent graph derivation of the *Flickr* dataset

	FlakeODF	*ContentCohesiveness*	*Entropy*
Full Communities	0.77	**0.29**	**0.92**
Cosine	0.79	0.18	0.71
Cosine-Weighted	**0.80**	**0.19**	0.72
Euclidean	0.80	0.18	0.70
Euclidean-Weighted	**0.80**	0.18	**0.78**
Harmonic	0.80	0.18	0.69
HDI	0.79	0.18	0.70
HPI	0.79	0.18	0.72
LHN	0.80	0.18	0.72
Manhattan-Weighted	**0.80**	0.18	**0.78**
PearsonCorrelation	0.80	0.18	0.69
Sorensen	0.80	0.18	0.70
Tanimoto	0.79	0.18	0.70
Tanimoto-Weighted	0.80	**0.19**	0.76

The performed Friedman test showed with a confidence of 1.229e−07 that the null hypothesis should be rejected, meaning that there is a difference between any of the analysed pair of results. To discover the pairs for which a statistical difference existed, the Wilcoxon test was applied defining the same hypotheses. Wilcoxon results showed with a confidence value of 0.05 the existence of significant differences amongst the diverse vertex similarity metrics. For example, $Euclidean$ results were shown to be statistically different and lower than most of those of the other metrics, in terms of *FlakeODF* and *ContentCohesiveness*. On the other hand, $Tanimoto$-$Weighted$ was shown to be statistically different than most of the other metrics in terms of *Entropy*. Interestingly, $FullCommunities$ showed to be statistically superior than several metrics, in terms of *ContentCohesiveness* for HPI, LHN, $Manhattan - Weighted$, $Tanimoto$, $Pearson$ and $Euclidean$. No significant differences were observed in terms of *FlakeODF* and *Entropy*. Finally, even though differences were observed for the binary and weighted variations of the metrics, such differences were not statistically significant.

Weighted Social View Figure 12.9 shows the obtained results for the different combinations of node relationships for the weighted derivation of the social graph. For every combination of relationships, considering the full node set allowed to obtain the highest *Entropy* results. Moreover, for every combination

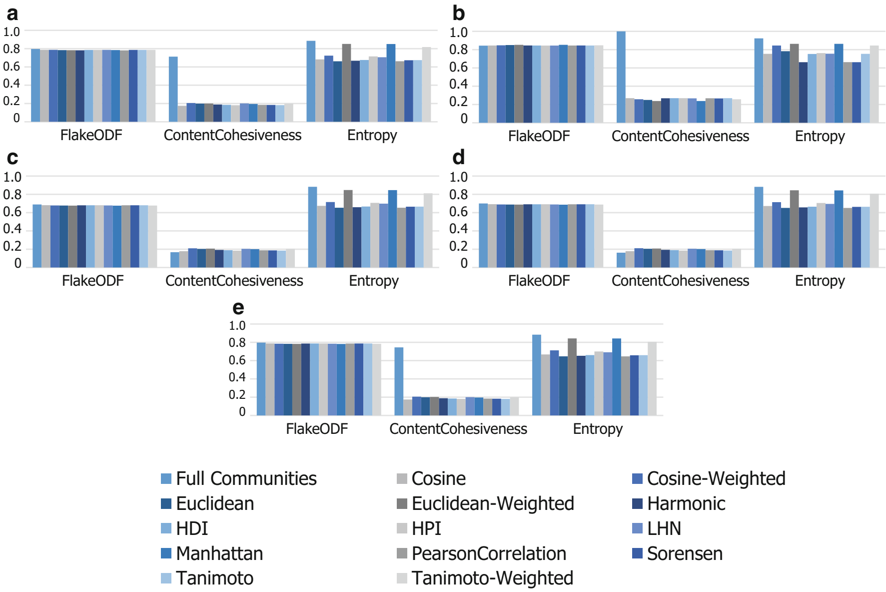

Fig. 12.9 Vertex similarity results for the weighted social views for the *Flickr* dataset. (**a**) *Social-W-SharedClass & SimilarContent-0.6*. (**b**) *Social-W-SimilarContent & SimilarContent-0.6*. (**c**) *Social-W-TaggedSameUser & SharedClass*. (**d**) *Social-W-CommentedSameUser & SimilarContent-0.6*. (**e**) *Social-W-SharedTag & SimilarContent-0.6*

Table 12.9 Summary of vertex similarity results for the weighted graph derivation of the *Flickr* dataset

	FlakeODF	*ContentCohesiveness*	*Entropy*
Full Communities	**0.77**	**0.56**	**0.89**
Cosine	0.76	0.20	0.69
Cosine-Weighted	0.76	**0.22**	0.74
Euclidean	0.76	0.21	0.68
Euclidean-Weighted	0.76	0.21	**0.85**
Harmonic	0.76	0.21	0.66
HDI	0.76	0.20	0.68
HPI	0.76	0.20	0.72
LHN	0.76	**0.22**	0.71
Manhattan-Weighted	0.76	**0.21**	**0.85**
PearsonCorrelation	0.76	0.20	0.66
Sorensen	0.76	0.20	0.66
Tanimoto	0.76	0.20	0.68
Tanimoto-Weighted	0.76	0.21	0.82

excepting *Social-W-TaggedSameUser* & *SimilarContent-0.6*, *FullCommunities* obtained the highest *ContentCohesiveness,* with differences up to a 300%. These results also exposed the redundancy amongst node relationships as *Social-W-SharedTag* & *SimilarContent-0.6* and (Fig. 12.9e) *Social-W-SharedClass* & *SimilarContent-0.6* (Fig. 12.9a) obtained similar results. The same applies for *Social-W-TaggedSameUser* & *SimilarContent-0.6* (Fig. 12.9c) and *Social-W-CommentedSameUser* & *SimilarContent-0.6* (Fig. 12.9d).

Table 12.9 summarises the results obtained for each vertex similarity metric averaged for every node relationship analysed. The best results obtained for each metric are in bold. Similarly as for the independent graph derivation, all metrics obtained similar average *FlakeODF*. However, unlike for the independent graph derivation, *Full Communities* obtained the highest average results. The highest differences were observed for *ContentConhesiveness*. The same statistical analyses performed for the independent derivation results were performed for this graph derivation. Results showed the same tendencies than for the independent graph derivation.

12.6 Conclusions

Social networking and microblogging sites have increased their popularity in recent years attracting millions of users, who spend an increasing amount of time on those sites sharing personal information and making new friends. For example, sites like *Flickr*, *YouTube*, *Facebook* or *Twitter* allow users to create content, publish photos, comment on content other users shared, tag content and socially connect with other users in the form of subscriptions or friendships. Consequently, social networking sites affect how people communicate and interact with each other.

One fundamental problem in social networks is the identification of groups of elements (users, posts or other elements) when group membership is not explicitly available. A group, or community, can be defined as a set of elements that interact more frequently or are more similar to other community members than to outsiders. Community detection has proven to be valuable in diverse domains such as biology, social sciences and bibliometrics. As a result, several community detection methods have been developed based on techniques from a variety of disciplines. Given the heterogeneity of real-world networks, one question that arises is how to effectively evaluate the algorithms.

Motivated by the lack of studies analysing the problem of community detection over real-world social media networks, this chapter focused on the analysis of the performance of community detection algorithms over such type of networks (particularly over *Twitter* and *Flickr*) including the effect of diverse metrics for assessing community membership. To that end, it was also explored how to quantify the structural properties of the discovered communities in terms of several quality metrics. The obtained results exposed the sensitivity of community detection algorithms to the density and structure of the underlying graph distribution, and hence their lack of robustness. Results showed that the Louvain algorithm achieved high-quality communities for almost every analysed combination of relationships, reinforcing its capabilities and stability for accurately discovering community structures, as claimed by [44]. Moreover, the study showed the relation and dependence of several quality metrics. Finally, as regards community membership, most of the analysed metrics obtained similar results. Nonetheless, those results varied according to the analysed dataset, highlighting the importance of considering the intrinsic characteristics of the social network under analysis for the community detection process.

References

1. Arenas, A., Díaz-Guilera, A., Pérez-Vicente, C.J.: Synchronization reveals topological scales in complex networks. Phys. Rev. Lett. **96**, 114102 (2006)
2. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech: Theory Exp. **2008**(10), P10008 (2008)
3. Brandes, U., Gaertler, M., Wagner, D.: Experiments on graph clustering algorithms. In: Di Battista, G., Zwick, U. (eds.) Algorithms – ESA 2003, pp. 568–579. Springer, Berlin/Heidelberg (2003)
4. Chua, T.-S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: NUS-WIDE: a real-world web image database from National University of Singapore. In: Proceedings of the ACM International Conference on Image and Video Retrieval, CIVR '09, pp. 48:1–48:9. Association for Computing Machinery, New York (2009)
5. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**, 066111 (2004)

6. Condon, A., Karp, R.M.: Algorithms for graph partitioning on the planted partition model. Random Struct. Algoritm. **18**(2), 116–140
7. Corder, G.W., Foreman, D.I.: Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach. Wiley, Hoboken (2009)
8. Cui, P., Zhu, W., Chua, T.-S., Jain, R.: Social-sensed multimedia computing. IEEE MultiMedia **23**(1), 92–96 (2016)
9. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B **39**, 1–38 (1977)
10. Deza, M.M., Deza, E.: Encyclopedia of Distances. Springer, Berlin/Heidelberg (2009)
11. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. Mach. Learn. **2**(2), 139–172 (1987)
12. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010)
13. Fortunato, S., Barthélemy, M.: Resolution limit in community detection. Proc. Natl. Acad. Sci. **104**(1), 36 (2007)
14. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. **99**(12), 7821–7826 (2002)
15. Gregory, S.: Fuzzy overlapping communities in networks. J. Stat. Mech. Theory Exp. **2011**(02), P02017 (2011)
16. Hochbaum, D.S., Shmoys, D.B.: A best possible heuristic for the k-center problem. Math. Oper. Res. **10**(2), 180–184 (1985)
17. Huang, A.: Similarity measures for text document clustering. In: Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, pp. 49–56 (2008)
18. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. Phys. Rev. E **80**(5), 056117 (2009)
19. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**, 046110 (2008)
20. Leskovec, J., Lang, K.J., Mahoney, M.: Empirical comparison of algorithms for network community detection. In: Proceedings of the 19th WWW, pp. 631–640. Association for Computing Machinery, New York (2010)
21. Lusseau, D.: Evidence for social role in a dolphin social network. Evol. Ecol. **21**(3), 357–366 (2007)
22. Malliaros, F.D., Vazirgiannis, M.: Clustering and community detection in directed networks: a survey. Phys. Rep. **533**(4), 95–142 (2013). Clustering and Community Detection in Directed Networks: A Survey.
23. Marin, A., Wellman, B.: Social network analysis: an introduction. In: Carrington, P., Scott, J. (eds.) Handbook of Social Network Analysis. Sage, London (2010)
24. McAuley, J., Leskovec, J.: Image Labeling on a Network: Using Social-Network Metadata for Image Classification, pp. 828–841. Springer, Berlin/Heidelberg (2012)
25. McDaid, A., Hurley, N.J.: Detecting highly overlapping communities with model-based overlapping seed expansion. In: 2010 International Conference on Advances in Social Networks Analysis and Mining, pp. 112–119 (2010)
26. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74**(3), 036104 (2006)
27. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
28. Orman, G.K., Labatut, V.: A Comparison of Community Detection Algorithms on Artificial Networks, pp. 242–256. Springer, Berlin/Heidelberg (2009)
29. Orman, G.K., Labatut, V., Cherifi, H.: On accuracy of community structure discovery algorithms. CoRR, abs/1112.4134 (2011)
30. Orman, G.K., Labatut, V., Cherifi, H.: Towards realistic artificial benchmark for community detection algorithms evaluation. Int. J. Web Based Communities **9**(3), 349–370 (2013)
31. Papadopoulos, S., Kompatsiaris, Y., Vakali, A., Spyridonos, P.: Community detection in social media – performance and application considerations. Data Min. Knowl. Discov. **24**(3), 515–554 (2012)

32. Pelleg, D., Moore, A.W.: X-means: extending k-means with efficient estimation of the number of clusters. In: Proceedings of the 17th ICML, pp. 727–734. Morgan Kaufmann Publishers, San Francisco (2000)
33. Pons, P., Latapy, M.: Computing Communities in Large Networks Using Random Walks, pp. 284–293. Springer, Berlin/Heidelberg (2005)
34. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E. **76**(3), 036106 (2007)
35. Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. Phys. Rev. E **74**, 016110 (2006)
36. Rosvall, M., Axelsson, D.T., Bergstrom, C.T.: The map equation. Eur. Phys. J. Spec. Top. **178**(1), 13–23 (2009). https://doi.org/10.1140/epjst/e2010-01179-1
37. Sawardecker, E.N., Sales-Pardo, M., Amaral, L.A.N.: Detection of node group membership in networks with group overlap. Eur. Phys. J. B **67**(3), 277–284 (2009)
38. Schaeffer, S.E.: Graph clustering. Comput. Sci. Rev. **1**(1), 27–64 (2007)
39. Tang, J., Wang, X., Liu, H.: Integrating Social Media Data for Community Detection. LNAI, vol. 7472, pp. 1–20 (2012)
40. Tommasel, A., Godoy, D.: Multi-view community detection with heterogeneous information from social media data. Neurocomputing **289**, 195–219 (2018)
41. Wakita, K., Tsurumi, T.: Finding community structure in mega-scale social networks. In: Proceedings of the 16th WWW, pp. 1275–1276. Association for Computing Machinery, New York (2007)
42. Xie, J., Kelley, S., Szymanski, B.K.: Overlapping community detection in networks: The state-of-the-art and comparative study. ACM Comput. Surv. **45**(4), 43:1–43:35 (2013)
43. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. In: Proceedings – IEEE International Conference on Data Mining, pp. 745–754. IEEE Computer Society (2012)
44. Yang, Z., Algesheimer, R., Tessone, C.J.: A comparative analysis of community detection algorithms on artificial networks. Sci. Rep. **6**, 30750 (2016)
45. Zachary, W.W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. **33**(4), 452–473 (1977)
46. Zubiaga, A., Spina, D., Martínez, R., Fresno, V.: Real-time classification of twitter trends. J. Assoc. Inf. Sci. Technol. **66**(3), 462–473 (2015)