ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/270593760

Clustering attributed graphs: Models, measures and methods

Article · January 2015

DOI: 10.1017/nws.2015.9 · Source: arXiv

CITATIONS
81

READS
640

4 authors, including:

Cécile Bothorel
Institut Mines-Télécom
57 PUBLICATIONS **279** CITATIONS

SEE PROFILE

Juan David Cruz
Institut Mines-Télécom
16 PUBLICATIONS **165** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Quality of communities in Social Networks View project

Project Cooperación en Sistemas Multi-Agente View project

All content following this page was uploaded by Juan David Cruz on 21 May 2015.

The user has requested enhancement of the downloaded file.

arXiv:1501.01676v1 [cs.SI] 7 Jan 2015

Clustering attributed graphs: models, measures and methods∗

CECILE BOTHOREL, JUAN DAVID CRUZ
Department of Logics in Uses, Social Science and Information Science,
UMR CNRS 3192 Lab-STICC,
Télécom Bretagne, Institut Mines-Télécom, Brest, France
(*e-mail:* cecile.bothorel, juan.cruzgomez@telecom-bretagne.eu)

MATTEO MAGNANI†
Computing Science Division, IT Department, Uppsala University, Sweden
(*e-mail:* matteo.magnani@uu.se)

BARBORA MICENKOVÁ
Data Intensive Systems, Department of Computer Science,
Aarhus University, Denmark
(*e-mail:* barbora@cs.au.dk)

Abstract

Clustering a graph, i.e., assigning its nodes to groups, is an important operation whose best known application is the discovery of communities in social networks. Graph clustering and community detection have traditionally focused on graphs without attributes, with the notable exception of edge weights. However, these models only provide a partial representation of real social systems, that are thus often described using node attributes, representing features of the actors, and edge attributes, representing different kinds of relationships among them. We refer to these models as *attributed graphs*. Consequently, existing graph clustering methods have been recently extended to deal with node and edge attributes. This article is a literature survey on this topic, organizing and presenting recent research results in a uniform way, characterizing the main existing clustering methods and highlighting their conceptual differences. We also cover the important topic of clustering evaluation and identify current open problems.

Contents

1 Introduction 2
1.1 Current trends in attributed graph analysis and mining 3
1.2 Clustering attributed graphs 4

∗ This version has been submitted to the Network Science journal and has been subsequently accepted for publication subject to minor revisions. It will appear in a revised form subsequent to peer review and/or editorial input by Cambridge University Press and/or the journal's proprietor (http://journals.cambridge.org/NWS).
† The author has been partly supported by the Italian Ministry of Education, Universities and Research FIRB grant RBFR107725.

2 Clustering edge-attributed graphs 4
2.1 Single-layer approaches 6
2.2 Extension of modularity 8
2.3 Clique-finding methods 10
2.4 Emerging clusters 12
3 Clustering node-attributed graphs 13
3.1 Data representation 14
3.2 Weight modification according to node attributes 15
3.3 Linear combination of attributes and structural dimensions 17
3.4 Walk-based approaches 19
3.5 Methods based on statistical inference 19
3.6 Subspace-based methods 20
3.7 Other methods 21
4 Practical aspects 22
4.1 Evaluation 22
4.2 Applicability 29
5 Open problems and discussion 31

1 Introduction

Graphs represent one of the main models to study human relationships. For example, structural properties of social systems can be measured by representing individuals and their relationships as graphs and computing the centrality or prestige of their nodes (Wasserman & Faust, 1994). Similarly, once a social graph is available, groups of strongly connected individuals (communities) can be identified using clustering algorithms. The application of graphs to the study of social systems motivated and is now a part of a broader discipline called *network science*, focused on the modeling and analysis of relationships between generic entities. This discipline provides a set of tools (methodologies, methods and measures) to improve our understanding of complex systems, including social and technological environments, transport and communication networks and biological systems. The wide applicability of network science largely relies on the adoption of graph-based models, that thanks to their generality can be applied to a diverse range of scenarios.

However, researchers in social network analysis (SNA) and social sciences have long been aware of the potential value in representing additional information on top of the social graph, and of the potential loss in accuracy when simple nodes and edges are used to represent complex social interactions. For example, according to Wasserman & Faust (1994) social networks contain at least three different dimensions: a *structural* dimension corresponding to the social graph, e.g. actors and their relationships, a *compositional* dimension describing the actors, e.g. their personal information, and an *affiliation* dimension indicating group memberships. The existence of multiple relationship types, e.g., *working together*, *being friends* or *exchanging text messages*, has also been studied for a long time, as recently reported by Borgatti *et al.* (2009). This last aspect has been referred to as *multiplexity* in the SNA tradition, and can be related to Goffman's concept of *context*, well exemplified by the metaphore of individuals acting on multiple stages depending on their

audience (Goffman, 1974). As an example, Figure 1(b) highlights how an *attributed* graph may lead to a deeper understanding of social interactions if compared to the corresponding graph without attributes in Figure 1(a).

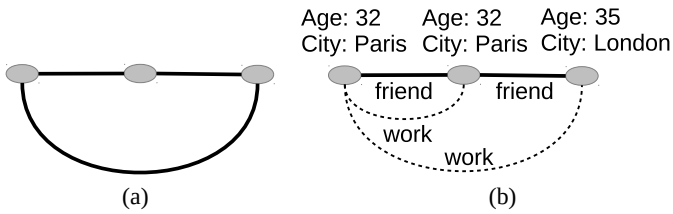


Fig. 1. A graph (a) provides a simplified representation of a social system which can be easy to understand but may prevent a deep understanding of its structural and compositional dimensions (b)

1.1 Current trends in attributed graph analysis and mining

Attributed graphs have been used for decades to study social environments and it has been long recognized that the structure of a social network may not be sufficient to identify its communities (Freeman, 1996; Hric *et al.*, 2014). However, recent years have witnessed a renewed attention towards these models, partially motivated by the availability of real data from on-line sources. One interesting aspect of real attributed graphs is the observed *dependency* between who the actors are and how they interact, i.e. between the structural and compositional dimensions. For example, La Fond & Neville (2010) have observed the coexistence of social influence and homophily. Social influence states that people who are linked are likely to have similar attributes, thus node attribute values can be interpreted as a result of interactions with other nodes. At the same time, homophily implies that people with similar attributes are likely to build relationships. These two related phenomena have been observed in real networks by Kossinets & Watts (2006), and the dependency between attributes and connectivity has been studied mathematically (Kim & Leskovec, 2012).

With this in mind, researchers have focused on *attributed graph generators*. Artificially grown graphs are useful to experiment algorithms and run simulations when real data are difficult to collect. They are relevant in testing *what if* scenarios, providing forecasts on future evolutions, and can be used to design graph sampling algorithms when the size of original graphs would otherwise make the analysis impractical (Leskovec *et al.*, 2005).

Prior models, as the well-known preferential attachment mechanism by Barabási & Albert (1999), have focused on the social structure. Now the challenge is to generate datasets as close as possible to real-world social graphs, as done by Zheleva *et al.* (2009) where affiliation information is also generated. This model captures previously studied properties (e.g. power-law distribution for social degree) but also provides new interesting insights regarding the processes behind group formation. More recently Gong *et al.* (2011) have proposed a generative *social-attribute network* model based on their empirical observations of Google+ growth. Here attributes describe user characteristics like name of attended school and group membership. Nan Du *et al.* (2010); Magnani & Rossi (2013a) have instead focused on the generation of graphs with interdependent attributes on the edges.

The idea that attributes and connections are generated in an interdependent way has led to the development of specialized analysis methods. Several graph mining tasks have been extended to attributed graphs, like link prediction (Getoor & Diehl, 2005; Rossetti *et al.*, 2011; Gong *et al.*, 2011; Sun *et al.*, 2012) or attribute inference (Li & Yeung, 2009; Gong *et al.*, 2011; Yang *et al.*, 2011). This survey is dedicated to one of the most relevant and studied operations on graphs and complex networks: graph clustering, often referred to as *community detection* when social graphs are involved. We believe that this is an important and timely effort to facilitate research in this still young area, in particular considering that the discussed approaches have been introduced in different disciplines, often unaware of each other.

1.2 Clustering attributed graphs

Although several surveys on graph clustering have been written (Schaeffer, 2007; Fortunato, 2010; Aggarwal & Wang, 2010; Coscia *et al.*, 2011), most of the approaches to cluster attributed graphs are more recent and have not been included in these works. At the same time, there is a large literature on (multi-dimensional) clustering of tabular data (Moise *et al.*, 2009; Han *et al.*, 2011), but existing surveys in this area have not addressed extensions for graph data. Attributed graph clustering can be seen as the confluence of these two fields, the former focusing on the structural and the latter on the compositional aspects. In this article we focus on recent works resulting from this promising combination.

The article is organized in three main parts: a review of methods for edge-attributed graphs, a review of methods for node-attributed graphs, and a section on practical issues including the evaluation of clusterings and the applicability of different approaches. We conclude by summarizing the status of the research and discussing the open problems that are more promising according to our view of the area. Attributed graph clustering has been independently studied in different disciplines, therefore it is important to know how different terms have been used in the literature. In Table 1 we have indicated and briefly explained the terms used in this article.

2 Clustering edge-attributed graphs

One way to extend a graph model and to provide additional information to the clustering algorithm is to represent the different kinds of edges among individuals. As an example, in Figure 1(b) we can see that the relationship between the two left-most nodes consists of a *friendship* and a *working* edge.

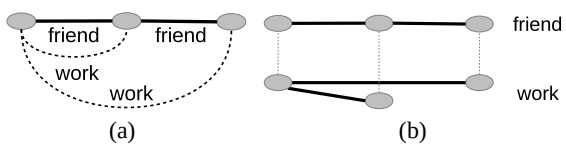


Fig. 2. Two alternative representations of the different edge types in a multigraph

Different models have been used to represent this scenario (Minor, 1983; Lazega & Pattison, 1999; Skvoretz & Agneessens, 2007; Kazienko *et al.*, 2010; Berlingerio *et al.*, 2011b), some-

Table 1. Terminology used in this article and synonyms used in the literature

main term	**synonyms**	**meaning**
Node	Vertex, site, actor	Basic component of a graph. As an example, a node may indicate that a user has an account on the social media site whose social network is represented by that graph.
Edge	Link, arc, tie, connection, bond, relation(ship)	A relationship between two nodes, e.g., a *following* relationship between two Twitter accounts. When there is an edge between two nodes we say that they are directly connected.
Graph	Network, social network, layer	A graph without attributes, neither on nodes nor on edges, with the exception of an optional numerical weight on edges indicating the strength of the connection. Edges may be directed or indirected.
Edge-attributed graph	Multiplex network, multi-layer graph, multidimensional network, edge-labeled multi-graph	Attributes indicate connections of different kinds or inside different graphs. With this term we do not indicate the presence of weights, in which case we explicitly talk of weighted graph/edges.
Node-attributed graph	Node-labeled graph, graph with feature vectors	A feature vector is associated with each node and contains information about it, e.g., age, nationality, language, income.
Attributed graph	Attribute graph, social and affiliation network, relational data, multidimensional network	An edge-attributed graph, or a node-attributed graph, or both.
Layer	Aspect, dimension	Sometimes all the edges with the same attribute value in an edge-attributed graph are indicated as a *layer*, e.g., the *Facebook friendship*, *spacial proximity*, *Twitter following*, *colleague* or *family* layers in an attributed graph indicating different types of social relationships.
Clustering	Community structure	Assignment of each node to one or more groups of nodes, called clusters. Different criteria can be used to determine whether two nodes should belong to the same cluster.
Partition	Non-overlapping clustering	A clustering where each node is assigned to exactly one cluster.

times emphasizing the different roles played by individuals with respect to different networks (Magnani & Rossi, 2011), including different kinds of nodes (Cai *et al.*, 2005) or providing a more general data model to mathematically represent a graph with attributes on both nodes and edges (Kivelä *et al.*, 2014). In Figure 2 we can see two alternative representations of the same data, as a multigraph (a) and as a set of interconnected graphs (b). The former, sometimes referred to as a *multiplex network*, focuses on a single set of nodes that may have complex relationships between them:

Definition 1 (*Multi-relational edge-attributed graph*)
Given a set of nodes N and a set of labels L, an edge-attributed graph is a triple $\{G = (V,E,l)\}$ where $V \subseteq N$, (V,E) is a multi-graph and $l : E \to L$. Each edge $e \in E$ in the graph has an associated label $l(e)$.

The latter emphasizes how the same node can belong to multiple (social) graphs, also known as *layers*:

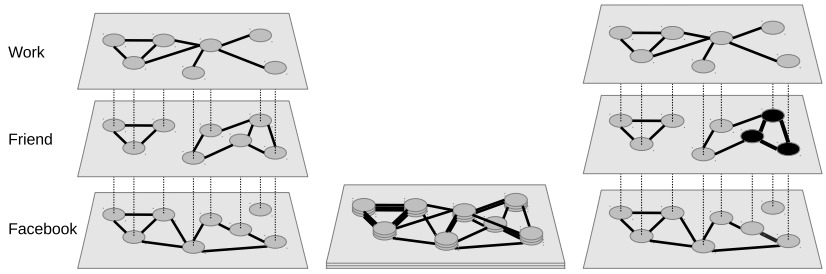
Definition 2 (*Multi-layer edge-attributed graph*)
Given a set of nodes N and a set of labels L, an edge-attributed graph is defined as a set of graphs $G_i = (V_i,E_i)$ where $V_i \subseteq N$, $E_i \subseteq V_i \times V_i$. Each graph G_i has an associated unique name $l_i \in L$.

Although very similar, and in this specific example equivalent, these two representations emphasize different aspects of an edge-attributed graph. It is important to understand that the methods covered in the remaining of this section have been developed starting from specific models, influencing their features. Researchers using the first model have mainly focused on the reduction of different edge types to single edges, while researchers using the second model have looked for clusters spanning different layers and nodes belonging to multiple clusters depending on the edge type. With this difference in mind, in the following we will formally represent both scenarios using the second (more general) model, where a family of graphs possibly containing common nodes represent the different kinds of edges. A larger working example is shown in Figure 3(a).

More general definitions have been provided in the literature, where one node in one graph can correspond to multiple nodes in another. This includes the case of online social media, where the same user can open multiple accounts on some services (Magnani & Rossi, 2011), and the case of non-social networks containing different kinds of nodes, such as a power grid and a control network, where one node in a network can be related to multiple nodes in another (Gao *et al.*, 2011). Similarly, the model introduced by Kivelä *et al.* (2014) allows the presence of attributes both on nodes and edges. For the sake of simplicity we focus on the simpler definitions above, because they are the ones used by almost all works on clustering social networks to date. Also, notice that we focus on nominal attributes, e.g. *work* and *friendship*: the case where attributes are only numeric, that is, weighted graphs, has already been treated in depth in existing surveys. However, we will deal with numeric weights when these are used inside algorithms for nominal attributes.

2.1 Single-layer approaches

A basic approach to deal with edge-attributed graphs is to *flatten* them: to reconstruct a single weighted graph so that existing clustering methods can be indirectly applied. This



(a) (b) (c)

Fig. 3. An edge-attributed graph, corresponding to a set on interconnected graphs defined on a common superset of individuals (a). An indirect way to process it is to reduce it to a single weighted graph, then apply classical clustering algorithms (b). A significantly different approach is to look at exclusive connections (c)

approach, exemplified in Figure 3(b), is not restricted to clustering but can be applied to any operation defined on weighted graphs. Weights can be computed straightforwardly so that an edge between two nodes has a weight proportional to the number of graphs where the two nodes are directly connected.

Definition 3 (*Flattening*)
A flattening of an edge-attributed graph $(\{G_i\})$ is a weighted graph (E_f, V_f, w_f) where $E_f = \bigcup E_i, V_f = \bigcup V_i$ and $w(u,v) = \frac{|\{i \mid (u,v) \in E_i\}|}{N}$ (where N is the total number of graphs).

Berlingerio *et al.* (2011a) follows this approach. However, the same authors point out how this solution may discard relevant information, e.g., the fact that some attribute values (or graph layers) are more important than others to define a cluster. Tang *et al.* (2011) propose a more general framework where the information about the multiple edge types is considered during one of the four different components of the community detection process, network flattening being one of them. Nevertheless, the authors point out that this kind of integration requires that edges of different types share the same community structure. Therefore, it is not suitable for cases where the structures significantly vary in different dimensions.

An antithetic approach acknowledging the importance of edge-attributed models but still not considering clusters that can span several graphs is introduced by Bonchi *et al.* (2012). While flattening tends to assign nodes directly connected on multiple graphs to the same group because they get connected by a strong edge in the flattened graph, Bonchi *et al.* (2012) consider a set of nodes as a good cluster if their relationships are as specific and homogeneous as possible, i.e., they are mainly connected through the same edge type. An example is presented in Figure 3(c) where the three nodes marked in black are connected with each other in the middle layer but only share one single edge on all other layers, representing a good cluster according to this approach[1].

[1] Please notice that this specific example is not compatible with the original model by Bonchi *et al.* (2012) where individuals are allowed to be directly connected only on one of the layers. However, it retains its underlying intuition. While this work was not originally intended to be applied to this domain, it still presents a worth-mentioning alternative point of view.

The next sections are devoted to methods aiming at identifying clusters spanning multiple layers. They are mostly extensions of quality measures traditionally used in graph clustering, modularity and quasi-cliques being two prominent examples.

2.2 Extension of modularity

Modularity is a measure of how well the nodes in a graph can be separated into dense and independent components (Newman & Girvan, 2004). Figure 4 shows four graphs with their nodes assigned into two communities (black and white) and the modularities resulting from these assignments. In these examples it clearly appears how the assignments putting together highly interconnected nodes and separating groups of nodes with only a few connections between them get a higher value of modularity. It is worth noticing that modularity is not a method to find communities, but only a quality function. However, it can be directly optimized or used inside community detection methods to guide the clustering process.

Although this measure suffers from some well known pitfalls (Fortunato & Barthélemy, 2007; Lancichinetti & Fortunato, 2011), it has recently been at the basis of several graph clustering methods and it has also been extended to deal with attributed graphs. Let us briefly introduce it[2], to later simplify the explanation of its extension. The modularity is thus expressed as

$$Q = \frac{1}{2m} \sum_{ij} \left(a_{ij} - \frac{k_i k_j}{2m} \right) \delta(\gamma_i, \gamma_j), \tag{1}$$

where $\delta(\gamma_i, \gamma_j)$ is the Kronecker delta which returns 1 when nodes i and j belong to the same cluster, 0 otherwise. Therefore, the sum is computed only for those pairs of nodes that are inside the same cluster. For each of these pairs, the presence of an edge between them improves the quality of the assignment: a_{ij} equals 1 when there is an edge between i and j, 0 otherwise. As we are dividing everything by m (the number of edges in the graph), edges between nodes belonging to different clusters negatively affect modularity because they are not considered in the numerator (as $\delta(\gamma_i, \gamma_j) = 0$), but are counted in the denominator (m). Finally, the formula considers the fact that two nodes with high degree would be more likely to end up in the same cluster by chance, therefore their contribution is reduced ($-\frac{k_i k_j}{2m}$, where k_i and k_j are the degrees of i and j).

Now it should be easier to understand the extension of modularity proposed by Mucha *et al.* (2010) for edge-attributed graphs. Let us consider Figure 5: here we have emphasized how the same individual i can be present in multiple graphs at the same time. For example, i and j are directly connected on graphs r and s, where r and s represent two different edge types. Notice that in this example we have three graphs, i.e., three edge types, and that j is assigned to two different clusters in graphs r (gray) and s, t (white).

[2] Please notice that modifications of this formula have been proposed to make it more adaptable to different datasets. One typical addition is a *resolution parameter*, that we have omitted from the following equations because it is orthogonal to our discussion.

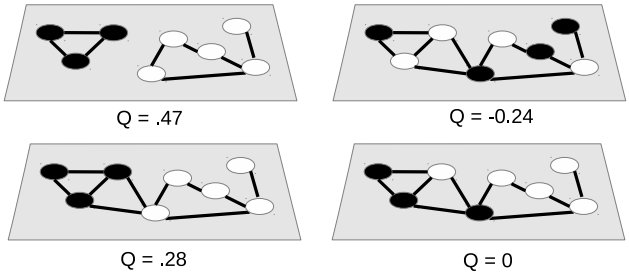


Fig. 4. Modularity of four graph clusterings: nodes in each graph are assigned to two clusters (black and white); the modularity of each assignment is reported under the graph

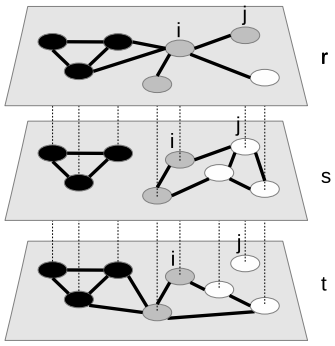


Fig. 5. An edge-attributed graph with three kinds of edges, represented as three interconnected graphs. Nodes have been assigned to three clusters (black, gray and white)

Thus, the extended version of the modularity can be expressed as

$$Q_m = \frac{1}{2\mu} \sum_{ijsr} \left[\left(a_{ijs} - \frac{k_{is}k_{js}}{2m_s} \right) \delta(s,r) + c_{jsr}\delta(i,j) \right] \delta(\gamma_{i,s}, \gamma_{j,r}). \tag{2}$$

This extended quality function involves not just all pairs of nodes (i, j) but also all pairs of graphs (s, r). μ and $\delta(\gamma_{i,s}, \gamma_{j,r})$ correspond respectively to m and $\delta(\gamma_i, \gamma_j)$ in the modularity formula, where μ also considers the connections between different graphs: we say that there is a connection between two graphs r and s whenever they contain a common node j, which increases μ by c_{jsr}. $\delta(\gamma_{i,s}, \gamma_{j,r})$ allows to assign the same node to different clusters inside different graphs. The sum is now made of two components. One is only computed when two nodes in the same graph are considered (because of $\delta(s,r)$), corresponding to modularity. In fact, here $a_{ijs} = 1$ when i and j are directly connected in graph s and k_{is} is the degree of node i in the same graph. The second component, c_{jsr}, is only computed when we are considering the same node j inside two different graphs r and s. This term increases the quality function by c_{jsr} (typically, a constant value ranging from 0 to 1) whenever we assign the same individual to the same cluster on different graphs.

One practical problem in using this measure is to set the c_{jsr} parameter. Setting it to 0 for all nodes and graphs, clusters are identified on each single graph independently of each other. If c_{jsr} is high, e.g., 1, it becomes unlikely to assign the same individuals to different clusters on different graphs. Other practical aspects to consider are the fact

that the part of the formula corresponding to traditional modularity can give a negative contribution, which is not true for the part taking care of inter-network relationships, and also the fact that the contribution of inter-network relationships grows quadratically on the number of networks while the modularity part only grows linearly. However, while the choice of appropriate parameters deserves more research, this extended definition of modularity can be directly used to find clusters by using any modularity-optimization heuristics, as done by Mucha *et al.* (2010), or paired with a concept of betweenness to extend the Girvan-Newman algorithm. The definition of betweenness for edge-attributed graphs follows directly from any definition of distance involving multiple graphs (Brodka *et al.*, 2011; Magnani *et al.*, 2013).

Figure 6 shows the values of modularity for four different multi-graphs and three different settings for the inter-graph parameter c_{jsr} (which is kept constant for all nodes and graphs). The figure emphasizes the different components of this measure. On the top we can see two clusterings aligned with both the single-graph and multi-graph structure. In particular, groups of nodes sharing several edges belong to the same cluster, and the same nodes on different graphs tend to belong to the same cluster. However, the top-right example shows that we can assign a node to different clusters in different graphs.

Modularities computed using different values of c_{jsr} cannot be compared: increasing c_{jsr} also increases the absolute value of modularity. However, we can see how the increase in the top-right figure is proportionally lower than the one on the left (from .48 to .68 and from .54 to .62, respectively). This is determined by the nodes assigned to multiple clusters.

The two lower figures show examples of lower modularity, i.e., clusterings not following the structure of the graphs. The lower-left image has a low overall intra-graph modularity which can be seen when $c_{jsr} = 0$ and thus inter-graph connections are not considered. When we also consider them ($c_{jsr} = .5$ and $c_{jsr} = 1$) we can see that modularity is increasing in the lower-left graph much more than in the lower-right one, where every node belongs to both clusters on different layers.

2.3 Clique-finding methods

Another concept used to discover clusters in graphs is the clique, i.e., a complete (sub)graph. Although this is one of the basic concepts in graph theory and it is thus well known, we briefly recall it.

Definition 4 (*Clique*)
A clique is a set of nodes directly connected to all other nodes in the clique.

Definition 5 (*Maximal clique*)
A maximal clique is a clique that is not contained in a larger clique.

Figure 7(a) shows an example of a clique. Any three nodes in Figure 7(a) still make a clique, but not a maximal one because we can add the fourth node and still have a clique.

A (maximal) clique clearly corresponds to a cluster. However, large cliques are difficult to find in real data because it is sufficient for one edge not to be present to break the clique, and in social graphs edges can be missing for many reasons, e.g., because of unreported data or just because even in a tight group there can be two individuals that do not get well

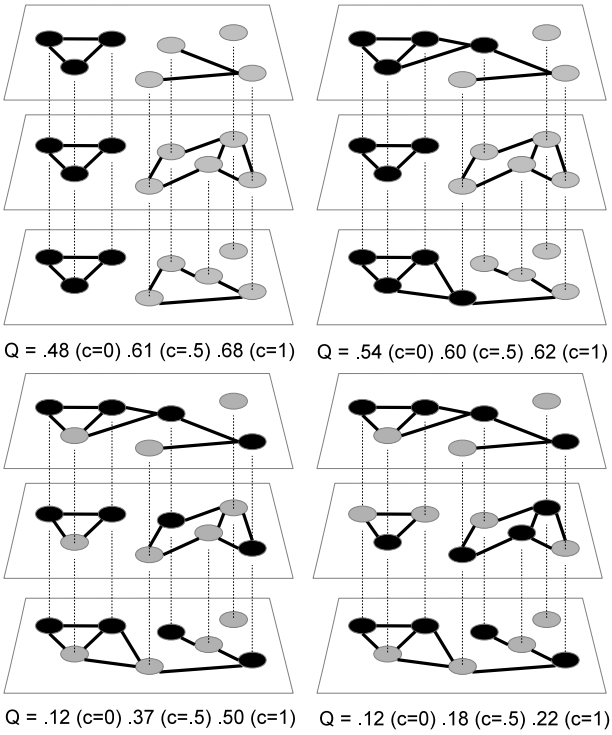


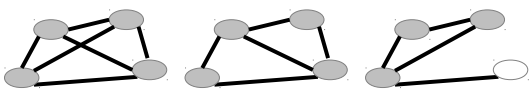
Fig. 6. Multi-layer modularity of four graph clusterings: nodes in each graph are assigned to two clusters (black and gray); the modularity of each assignment is reported under the graph using three settings: $c_{jsr} = 0$, $c_{jsr} = .5$ and $c_{jsr} = 1$

together. Therefore, when clustering is applied to social graphs, it is wiser to look for more relaxed structures called quasi-cliques.

For example, Freeman (1996) studies the cliques gathered from interviews to a group of individuals and acknowledges that they are not enough for defining communities.

Definition 6 (*Quasi-clique*)
A quasi-clique is a set of nodes where each node is directly connected to at least γ% of the other nodes in the quasi-clique.

Algorithms to discover quasi-cliques take γ as a parameter. Please notice that similar alternative definitions are possible, e.g., using a strict $>$ or considering the percentage over all nodes in the quasi-clique — the underlying concept remains the same. In Figure 7(b), we have illustrated a .5-quasi-clique, and in Figure 7(c), we have four nodes that do not constitute a .5-quasi-clique because the white node is directly connected to only one third of the other nodes.



(a) (b) (c)

Fig. 7. A clique (a), a quasi-clique (b) and four nodes not making a .5-quasi-clique (c)

The problem of finding quasi-cliques in a graph is NP-hard. According to common beliefs, this implies that no algorithm can exactly solve this problem in a reasonable amount of time even for small graphs. However, efficient algorithms which do not guarantee the identification of all quasi-cliques have been proposed.

As previously mentioned, the most common interpretation of clusters in edge-attributed graphs states that multiple kinds of edges between two individuals strengthen their relationship. Therefore, Pei *et al.* (2005) have introduced algorithms to discover quasi-cliques in all graphs and Wang *et al.* (2006); Zhiping Zeng (2006) to identify quasi-cliques in at least a given percentage of graphs (where this threshold is called *support*).

While not based on quasi-cliques, the ABACUS algorithm by Berlingerio *et al.* (2013) also applies a similar definition, coming from the *frequent itemset mining* problem. First, clusters are identified in each graph, then those individuals being in the same cluster in at least a given percentage of graphs are also included into a global cluster in the final result.

It is worth noticing that quasi-clique clustering methods were first developed for generic graph databases without focusing on the application domain of social graphs. In this specific domain, while we may agree that a cluster spanning all the graphs represents a strong global cluster, a group of nodes sharing a large number of edges on a few specific graphs may also identify a cluster of interest. For example, we might find that a group of individuals goes to the same school and plays in the same basketball team. This is a strong relationship that should not be negatively affected by the existence of other relationships where they do not form a group. However, adding other edge types to the attributed graph (which corresponds to adding new graphs to the multi-layer graph structure) would reduce their support.

The approach proposed by Boden *et al.* (2012) starts from this consideration and looks for sets of nodes that make a cluster in each single graph of *any subset* of the graphs in an edge-attributed model. This work also considers the case of weighted graphs, but this is peculiar to this method and we will not provide additional details here.

2.4 Emerging clusters

We conclude this section presenting a hypothesis still unverified in the literature that in our opinion might lead to the development of new clustering methods. The hypothesis is that clusters can emerge when a specific combination of graphs is considered, and disappear when more graphs are added to the model.

In Figure 8, the idea is illustrated on a simple example. The analysis of the three graphs together (right hand side of the figure) does not reveal any interesting patterns as there are too many edges in the graph. The same can be observed for each single graph (on the left). However, choosing two specific layers, some more evident clusters emerge (center, clusters denoted by black and white nodes). None of the previously presented approaches seems to be able to find such clusterings, because they require every cluster to be present in at least one of the single graphs.

This hypothesis would also provide an answer to the difficulty in finding good clusterings in real social graphs. In fact, although several clustering algorithms exist, in practice they achieve good results when some more or less well-separated clusters exist. This is strictly related to the way in which community detection algorithms have been defined:

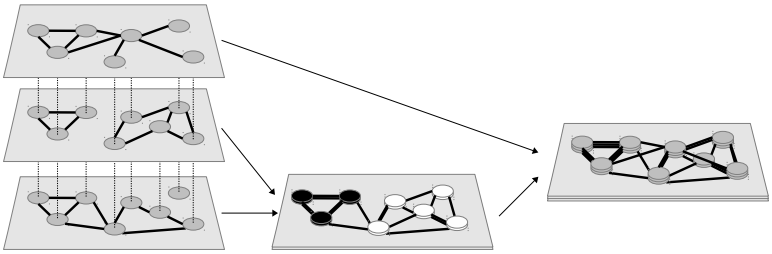


Fig. 8. Emerging clusters: well separated clusters appear when a specific subset of the graphs is used, but disappear when less or more networks are added

some try to maximize modularity, favoring well separated clusters, some use random walk approaches, where the probability that a walker crosses two clusters is proportional to the number of edges between them, some exploit measures like betweenness, that is high when few other edges connect distinct portions of the graph (Fortunato, 2010). However, when we deal with on-line relationships, clustering becomes extremely hard. According to our hypothesis, this depends on the fact that a large number of semantically different layers are considered all-together, determining the co-existence of several overlapping clusters, and a case of information overload.

In summary, if we consider Figure 8 (right side), we would not expect any clustering algorithm to find evident clusters. However, in theory clusters may appear when the multi-layer organization of the edges is unfolded in specific ways, e.g., by only retaining the two layers in Figure 8 (center). Therefore, the problem shifts from being purely algorithmic (e.g., how do we find the best cut?) toward aspects like the choice of the data model, data preprocessing and feature selection.

A preliminary work in this direction that can be seen as a conjunction between the idea of emerging clusters and the flattening approach is discussed by Rocklin & Pinar (2011). This work proposes an algorithm to find a vector that weights the layers to aggregate them such that the clustering of the resulting flattened graph is as similar to a given ground-truth clustering as possible (the clustering algorithm and a similarity measure between weighted single-layer graphs are given for this problem). The second half of the paper deals with the rich clustering structure that the multi-typed edges can provide. Generating random aggregates of the graph, the authors explore the space of possible clusterings and study, e.g., if good graph clusterings are *clustered* in this space. The final problem that they tackle is how to give an efficient representation of this resulting meta-clustering. Their approach is to reduce each meta-cluster (of clusterings) into a single representative clustering and select a small number of them to cover the meta-clustering space. In this way, they provide a set of diverse and non-redundant clusterings as output.

3 Clustering node-attributed graphs

According to the taxonomy presented by Getoor & Diehl (2005), node-attributed graph clustering aims at detecting groups of nodes sharing common characteristics considering both their attributes and their position in the graph. Most of the works addressing this problem are based on *partitioning* and *homophily*: nodes can belong to one and only one

group, and nodes in the same group must have homogeneous values on their attributes. A few other methods, also covered here, generate overlapping clusters, e.g., by considering different combinations of the attributes. This last approach is usually known as *subspace* clustering.

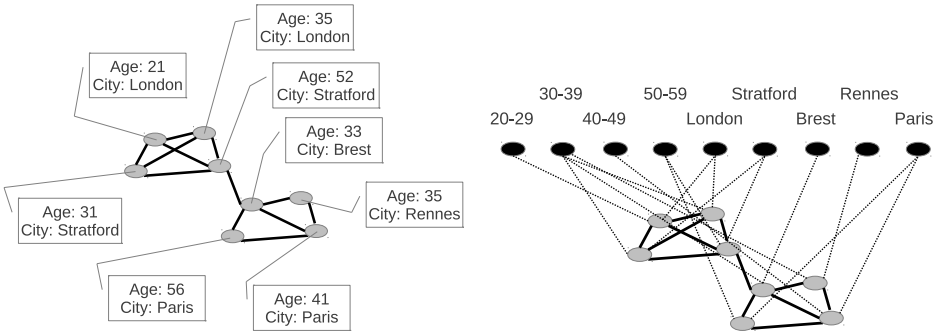
3.1 Data representation

Like in the case of edge attributes, also when attributes on nodes are considered, the literature abounds with terminologies and models depending on the research field or the finality of the work, making it difficult to provide a unified view. However, we can see some main options emerging.

As previously mentioned, Wasserman & Faust (1994) describe multiple dimensions that can be represented in a social network model: a *structural* dimension (relationships among actors), a *compositional* dimension (attributes of the single actors), and an *affiliation* dimension (representing group memberships). Affiliation information often refers to known groups such as clubs or companies, but it can also represent the cluster memberships discovered through a clustering process.

Two main options to represent such a model are shown in Figure 9. The first one, Figure 9(a), consists in extending a structural graph with tuples describing node properties. This can be formally expressed as a triple $G = (V, E, F)$ where each node v is associated with a set of a attributes (or a *feature vector*) $[f_1(v), \dots f_a(v)]$, storing its compositional dimension. Note here that the affiliation information may be stored in the same way, by adding attributes dedicated to memberships. The second option, Figure 9(b), consists in superimposing one or more graphs where additional nodes represent either specific attribute values or groups. Structurally, this superimposed graph is *bipartite* because it connects individuals to groups, without edges between groups or between users (the latter are stored in the original social network). More formally, a graph $G_p = (V_p, E_p)$ is augmented by a bipartite graph $G_a = (V_p \cup V_a, E_a)$, connecting nodes of V_p to attribute nodes of V_a, with no links between attributes: $E_a \subseteq V_p \times V_a$. This defines an augmented graph $G = (V, E)$ with $E = E_p \cup E_a$ and $V = V_p \cup V_a$.

Several terms have been used in the literature to refer to the options presented in Figures 9(a) and 9(b), or even for their intermediate variations. To make access to the existing literature easier, in Table 2 we report the main terms together with the references to where they appear and the indication of which modeling option has been adopted. Our objective here is not to be exhaustive: we aim at capturing the relationships between different approaches. For example when Tong *et al.* (2007) refer to an *attribute graph*, they imply that they have previously grouped the nodes with common attributes, and propose a meta-graph where meta-nodes reflect those groups and edge weights represent group-to-group similarity. Zheleva *et al.* (2009) study *social and affiliation networks* keeping two distinct graphs and observing the co-evolution of these two graphs via their common nodes, retrieved from Flickr groups. In the machine learning field, in the late 1990s and early 2000s, workshops dedicated to *link mining* referred to *relational data* (Neville *et al.*, 2003). In a more recent data warehousing context, Zhao *et al.* (2011) introduced an OLAP graph cube for *multidimensional networks*.



(a) (b)

Fig. 9. (a) Attributes represented as tuples describe node properties. The similarity/distance between tuples can be integrated into the graph and used during the clustering process. (b) New nodes representing the additional information are added to the original graph, resulting in a heterogeneous structure with multiple node types.

Table 2. Some terminology used in the literature to refer to node-attributed graphs

term	**references**	**option**
Social-attribute network	(Yin *et al.*, 2010a,b)	(b)
Attribute augmented graph	(Zhou *et al.*, 2009, 2010)	(b)
Attributed graph	(Zhou *et al.*, 2009; Cruz *et al.*, 2013; Cruz & Bothorel, 2013)	(a)
Feature-vector graph Vertex-labeled graph	(Günnemann *et al.*, 2013)	(a)

In summary, there has not been a consensus on the model yet. While different formats are useful to emphasize different aspects, all models include both structural and compositional data and one can be derived from another. Therefore, to introduce existing methods, we will use a common model consisting of an attributed graph $G = (V, E, F)$ where nodes are associated with an attribute vector $F(v)$.

3.2 Weight modification according to node attributes

The first class of methods we present is based on the following idea: first the node-attributed graph is reduced to a single weighted graph, where weights represent attribute similarity. Then, any clustering algorithm for weighted graphs can be applied in principle. Different methods use alternative functions to compute node similarity and to update edge weights when similarities have been computed. However, in all these approaches the change of weights influences the clustering algorithm to privilege the creation of groups in which the nodes are not only well connected but also similar.

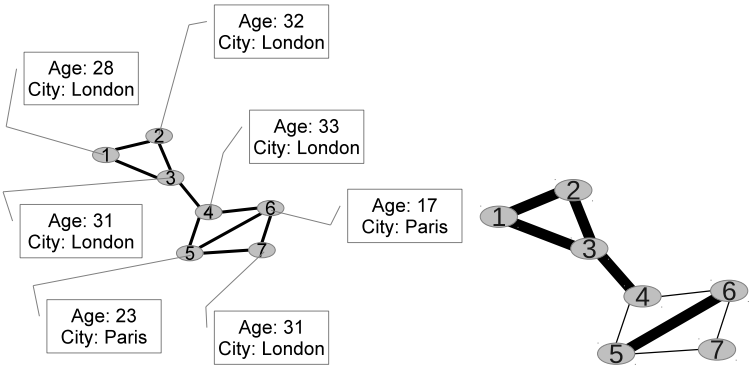


Fig. 10. A node-attributed graph (a) and an attribute-free representation of the same graph (b) where attribute similarities are stored in the edge weights (b). Thicker edges indicate a higher weight, i.e., a stronger connection

Table 3. Variations of the *weight modification* approach

reference	**similarity**	**clustering**
(Neville *et al.*, 2003)	Matching coefficient	Karger's Min-Cut MajorClust Spectral
(Steinhaeuser & Chawla, 2008)	Extended matching coefficient	Assign u and v to the same cluster when the weight of (u,v) is above a given threshold
(Cruz *et al.*, 2011b) (Cruz *et al.*, 2012)	Self-organizing maps	Louvain

As an example, consider Figure 10. Focusing solely on the attributes, nodes $\{1,2,3,4,7\}$ would form a homogeneous cluster, well separated from nodes $\{5,6\}$. If we only consider the structure of the graph, two clear clusters emerge (nodes $\{1,2,3\}$ and nodes $\{4,5,6,7\}$). These two pieces of information are summarized in the weighed graph in (b). While the specific final clusters depend on the assigned weights, we can see the emergence of a cluster made of nodes $\{1,2,3,4\}$, presenting both structural and compositional similarities and otherwise difficult to identify. Table 3 summarizes the main works adopting this strategy, and the measures mentioned in the table are reported in the following.

For example, Neville *et al.* (2003) use the *matching coefficient* similarity metric S_{ij} quantifying the number of attribute values (k) the nodes have in common. This similarity metric is expressed as

$$S_{ij} = \begin{cases} \sum_k s_k(i,j) & \text{if } e_{ij} \in E \text{ or } e_{ji} \in E \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where

$$s_k(i,j) = \begin{cases} 1 & \text{if } k_i = k_j \\ 0 & \text{otherwise} \end{cases}.$$

Once the weights have been changed, the graph is clustered using one of the three methods reported in Table 3: Karger's Min-Cut (Karger, 1993), MajorClust (Stein & Niggemann, 1999) or spectral clustering with a normalized cut objective function (Shi & Malik, 2000). Experimenting with artificial datasets, spectral clustering appears to be robust to irrelevant attributes and graphs with low linkage.

Steinhaeuser & Chawla (2008) extend the matching coefficient computation to take both discrete and continuous attributes into account: for discrete attributes, each common attribute shared by two nodes increments the weight of $e(u,v)$ by 1; for continuous attributes, the idea is to add the normalized distance between the attributes. Once the weights have been changed and normalized, all nodes, connected by an edge whose weight is greater than a threshold t, are assigned to the same cluster. In this specific work the quality of the final partition is evaluated using modularity (Newman & Girvan, 2004).

The approach presented by Cruz *et al.* (2011b, 2012) deals with the fact that not all attributes may be relevant to determine the similarity between nodes. When too many attributes are involved in the computation of traditional distance functions, e.g. Euclidean distance, we lose the ability to discriminate between different nodes. In fact, the so-called *curse of dimensionality* materializes in that all distances tend to converge to the same value. In addition, some attributes may need to be combined/transformed to become relevant. Therefore, the authors use a classical machine learning approach developed by Kohonen (1997) and known as self-organizing map (SOM)[3], to find the latent information worth to establish the similarity between the nodes. An edge between two nodes from the same cluster gets its weight strengthened proportionally to a given constant $\alpha \gg 1$. The resulting weighted graph is finally clustered using the Louvain method (Blondel *et al.*, 2008) and the overall complexity is linear $\mathcal{O}(n) + \mathcal{O}(fn) + \mathcal{O}(m)$, where n is the number of nodes, f the number of attributes or features and m the number of edges. Additionally, the authors introduce the notion of *point of view*: by manually selecting subsets of attributes, it becomes possible to analyze the social network from different perspectives.

It is worth noticing that this family of techniques produces new edge weights according to node attributes. If the original social graph is also weighted the two kinds of weights must be combined is some way, e.g., by multiplying them.

3.3 Linear combination of attributes and structural dimensions

The previous family of methods removes node attributes by storing their information inside the edges of the graph. Some studies adopt an opposite approach consisting in the removal of the network: structural information is stored into a similarity (or a distance) function between nodes. After defining this function, classic distance-based clustering methods can be applied. As an example, Combe *et al.* (2012) define a distance between nodes which is

[3] Self-organizing maps have been proposed as a learning approach that is robust to noise and can map high dimensional data into low dimensionality spaces, e.g. text.

Table 4. Similarity or distance functions combining structural and compositional dimensions

reference	**similarity or distance**
(Combe *et al.*, 2012)	$\alpha \cdot d_T(i,j) + (1-\alpha)\, d_S(i,j)$
(Villa-Vialaneix *et al.*, 2013)	$\alpha_0 K_0(i,j) + \sum_d \alpha_d K_d\left(c_i^d, c_j^d\right)$
(Dang & Viennet, 2012)	$\alpha \cdot G_{i,j} + (1-\alpha) \cdot simA(i,j)$

given by

$$d_{TS}(i,j) = \alpha \cdot d_T(i,j) + (1-\alpha)\, d_S(i,j), \tag{4}$$

where $d_T(i,j)$ and $d_S(i,j)$ are the attribute and structural similarity, respectively, between nodes i and j and $0 \leq \alpha \leq 1$ is a weighting factor. The authors leave the choice of the clustering method open. Another similar distance function by Dang & Viennet (2012), as listed in Table 4, is used to build a k-nearest neighbor graph in order to find clusters using the Louvain method (Blondel *et al.*, 2008).

The main feature of these approaches is that nodes which are structurally far from each other in the social graph can result to be close in case of similar attribute values. As a consequence, and depending on the distance-based clustering method, clusters may contain disconnected portions of the graph. Hanisch *et al.* (2002) experiment with a similar approach on biological networks and gene expression data. After the computation of the combined distance, they apply hierarchical clustering and a statistical measure to define the cutting point of the dendrogram.

While Villa-Vialaneix *et al.* (2013) share a similar purpose using a weighting parameter to balance their components, they rely on *kernels* to map the original (multi-space) data into an (implicit and unique) Euclidean space where SOMs can be used. In this case authors define a multi-kernel similarity function to combine composition and structure as indicated in Table 4. $K_0(i,j)$ indicates the kernel measuring structural similarity, c_i^d is the dth label of node i and α_d are weighting factors.

This approach also exploits the visual potential of SOMs which can be represented as bi-dimensional grids. In such grids, each cell represents a group of nodes, and the size of the cells is proportional to the number of observations associated with it. In this way the authors are able to represent the size of the communities, the distribution of topics and the links on the same 2-dimensional representation.

Dang & Viennet (2012) propose an extension of the Louvain method with a modification of modularity to include the similarity of the attributes in the community discovery process. This is given by

$$Q = \sum_{C \in \mathbf{C}} \sum_{i,j \in C} \left(\alpha \cdot S(i,j) + (1-\alpha) \cdot simA(i,j)\right), \tag{5}$$

where $\mathbf{C}$ indicates the set of graph partitions, $S(i,j)$ represents the strength between two nodes (computed as in the original definition of modularity), $simA(i,j)$ is a similarity

function based on attributes i and j and can be adapted according to how the attributes are represented. $0 \leq \alpha \leq 1$ is a weighting factor.

In general, for parametric methods an important question is how to choose α. According to the authors of these methods clusters are stable against small changes in the parameter. Dang & Viennet (2012) also propose a way to estimate α, and kernel-based approaches support automated parameter tuning (Villa-Vialaneix *et al.*, 2013). Depending on application, analysts may also set α to emphasize attribute homophily or connectivity. However, more case studies and future independent analyses will be welcome.

3.4 Walk-based approaches

A *random walk* on a possibly infinite network is a stochastic process where a walker goes from node to node by choosing a target neighbor at random at each step (Noh & Rieger, 2004). In the clustering context walk models are used to estimate vertex distances on attributed graphs. In accordance with this distance, k-means-like approaches attract *close* nodes around the predefined k centroids in order to aggregate the members of the communities.

Zhou *et al.* (2009) define a random walk process on graphs like the one in Figure 9(b). The result is that the more attribute values two vertices share, the more paths via the common attribute nodes exist. In this way random walks can be used to measure vertex proximity through both the structural links and the compositional links.

In the Connected k Centers method proposed by Ge *et al.* (2008) the walk strategy is a simple breadth-first search (BFS) defined for graphs like the one in Figure 9(a), where the feature vector is also used to determine the next visited node. This method implements the k-means algorithm using walks to compute distances: first, it picks k random nodes as cluster centers, second, all the nodes are assigned to one of the k clusters by traversing the graph using BFS; third the centroids of the clusters are recalculated. The second and third steps are repeated until there are no further changes in the clusters' centroids.

3.5 Methods based on statistical inference

Statistical inference is the process of drawing properties of datasets from a set of observations in a model and then inferring predictions about a larger population represented by the sample. In this section, and according to the classification provided by Fortunato (2010), we focus on two types of methods: the ones using generative models, as an intermediary step or in a pure manner to mix attributes and links in a unified model, and the ones using stochastic block models.

Many studies focus on the task of clustering networks of documents. Here, every document can be seen as a node characterized by a complex attribute defined by the words contained in the document. For example, Li *et al.* (2008) propose a clustering method to find communities in a large-scale document corpus exploiting both the document content (the words), and their references/citations. They use statistical inference as an intermediate step to find hidden topics to further manipulate the documents. The general principle is to find community cores and then include their members. The detection of cores identifies the documents that are frequently co-referenced and may play the role of community seeds.

A second phase merges the initial cores according to their topic similarity in order to improve the core consistency. The authors use here the well-known text-mining method called Latent Dirichlet Allocation (LDA) to find topics. LDA is a generative topic model so that unobserved or latent topics have probabilities to generate various observed words. A Bayesian inference finds the best fit of the model to the observations through likelihood maximization. Finally, the third step is to affiliate the remaining documents to the clusters. This affiliation propagation process may lead to misclassified documents and a final step removes false hits.

LDA is also used by Liu *et al.* (2009) and Balasubramanyan & Cohen (2011) but as a central approach and in an extended manner to identify latent groups. The Topic-Link LDA model defined by Liu *et al.* (2009) is a generative model considering topics, membership of authors and link formation between pairs of documents exhibiting both topic similarity and community closeness. The inference is designed to regularize the topic information when inferring the hidden communities and vice versa. The authors maximize likelihood using an expectation-maximization algorithm and demonstrate their unified model on three different tasks: topic modeling, community detection and link prediction in blogs and CiteSeer datasets. For the community detection task, we would highlight here an interesting remark. Their approach offers a meaningful investigation of how content similarity and community similarity contribute to the formation of links. They are able to reveal that author membership has a much stronger effect on link formation between blog posts in political domains than technical papers. They also show that the topic dimension plays a more important role than the community similarity in blog citing. Balasubramanyan & Cohen (2011) also address the problem of link modeling and combine two popular methods: block modeling and LDA.

Xu *et al.* (2012) propose a community detection model that is transformed into a statistical inference problem. Authors start by defining a generative Bayesian model that produces a sample of all the possible combinations of a graph, defined by its adjacency matrix $\mathbf{X}$, a matrix of features $\mathbf{Y}$ and a vector $\mathbf{Z}$ containing the assignation of each node to one out of k groups, i.e., a partition of the graph. This model produces a conjoint probability $p(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$. The idea is thus to find a partition $\mathbf{Z}^*$ such that $\mathbf{Z}^* = \arg_{\mathbf{Z}} \max p(\mathbf{Z} \mid \mathbf{X}, \mathbf{Y})$.

These techniques are very attractive to mix both attributes and topology into the same model, but unfortunately the optimization process to estimate the parameters of the likelihood is often costly. In addition, they do not rely on the definition of any distance, and the choice of the *a priori* distributions in the statistical models requires a non-trivial expertise.

3.6 Subspace-based methods

Some of the clustering approaches reviewed so far share the belief that a carelessly usage of all the available attributes may lead to poor clusterings. This is the case, e.g., in the work by Villa-Vialaneix *et al.* (2013). We have already recalled the phenomenon called *curse of dimensionality* in Section 3.2: when the number of attributes is large the difference in the distance between two random pairs of data points (actors, in this case) tends to zero. This phenomenon motivates the development of clustering approaches focused on the identification of the discriminative attributes to produce well separated clusters. This general approach is known as *subspace clustering*, and has been also applied to the case

of node-attributed graphs. Subspace clustering methods are designed to select the 'best' subsets of dimensions. They search the projections of the data in different dimensions and identify clusters that are relevant *locally* to some of these subspaces.

Subspace clustering is interesting because it may reveal groups that would not be detected considering the entire set of attributes. However finding relevant projections is computationally hard. The final choice of which groups to keep is also costly and requires an optimization step combining the best size, density, entropy, dimensionality and any other relevant quality function (see Section 4.1). Moreover, as each cluster is relevant in its own subspace, this has the effect of producing overlapping clusters and requires additional efforts to control the redundancy ratio between them.

One semi-automated approach to identify relevant subsets of attributes has been presented by Cruz *et al.* (2011b), where the authors propose a framework helping human analysts to manually select their preferred compositional perspective. The choice of the subset of attributes is given explicitly as an input to an automatic clustering process.

Differently, Günnemann *et al.* (2013) propose a completely automated method to efficiently combine subspace and subgraph clusters. In particular, they use their former GAMer method to extract an exhaustive list of candidate clusters, but apply a different final selection of the clusters to be returned to the user. The GAMer method greedily selects the clusters that locally optimize a quality measure. Here, they propose a solution based on global optimization, maximizing the sum of the clusters' qualities under redundancy constraints. The overall complexity of this definition of clustering is #P-hard[4]. Therefore, the authors propose a heuristic that, for example, produces a clustering of the whole DBLP database[5] in about 7 hours with commonly available hardware. They also show that the quality remains comparable to the greedy solution computed by GAMer in terms of F1 value and density.

The time complexity of subspace clustering approaches is notoriously high, but the discovery of dense subgraphs in selected subspaces can be valuable. However, the high number of required input parameters (minimum cluster size, dimensionality, density, redundancy) can have a negative impact on the practical usability of these methods. Finally, as we will see in Section 4.1, the evaluation of attributed graph clusters in general is still under study, and maybe more for overlapping ones where no ground truth exists.

3.7 Other methods

Other works directly extend well-known and efficient graph-based methods. Cruz *et al.* (2011a) extend the Louvain method (Blondel *et al.*, 2008) introducing a local minimization of the *entropy* generated by the attributes between the modularity optimization and the community aggregation steps. Dang & Viennet (2012) also extend the Louvain method in a similar way, by optimizing at each iteration the linear combination of the classical modularity and a new modularity based on the attributes.

[4] This is the complexity of some hard counting problems, and implies that an exact solution to this problem cannot be currently computed in acceptable time

[5] 133 097 nodes; 631 384 edges; 2 695 attribute dimensions. Available at: http://dblp.uni-trier.de

Akoglu *et al.* (2012) propose a method to identify cohesive groups in attributed graphs composed of n nodes each described by a feature vector. In this case, the attributes are binary, i.e., a node either has or not certain attributes. The algorithm uses the adjacency matrix $\mathbf{A}_{n\times n}$ of the graph and a matrix $\mathbf{F}_{n\times f}$ representing the assignation of features for each vector. The main underlying idea is to find k groups of nodes using the structural information and l groups using the feature information. The cost function is based on the encoding of the matrices $\mathbf{A}$ and $\mathbf{F}$ as well as the configuration of the clusters, where the encoding uses the approach proposed by Rissanen (1983).

Barbieri *et al.* (2013) present an approach using the notion of information cascades, and in particular the idea that an information cascade is more likely to occur within a community rather than between communities. Thus, they use a given set of information cascades to build a probabilistic model named Community-Cascade Network (CCN). To learn the parameters of the model authors use an expectation-maximization approach, which however has been reported to be computational expensive.

Ruan *et al.* (2013) also propose a content- and structure-based community detection algorithm called CODICIL. The algorithm starts by creating an edge set with the structure and a graph generated from the similarity of the nodes, i.e., the final edge set will contain the original structure plus edges derived from obtaining the top k most similar neighbors for each node. This similarity is calculated using the cosine distance between the TF-IDF vector from the content of each node. Then, this new graph is sampled to select certain relevant edges and, at last, this sampled graph is clustered using a classic graph clustering technique.

Finally, some approaches focus on the discovery of significant patterns, such as association rules or regular structures in graphs. Significant examples are the works by Moser *et al.* (2009), Silva *et al.* (2010), Atzmueller & Mitzlaff (2011) and Pool *et al.* (2014), focusing on mining descriptive community patterns and allowing the analysts to understand the structure of frequent subgraphs around topics which may be useful in scenarios like fraud detection or counter-terrorism. Differently from graph partitioning methods, frequent patterns can overlap and do not necessarily cover the entire dataset.

4 Practical aspects

4.1 Evaluation

Comparing the quality of two clusterings is a fundamental capability. It can be used to choose among alternative algorithms, inside a single algorithm as a stopping condition or as a guide to choose the next step in a so-called *greedy* approach, making an assignment that maximizes the quality improvement. However, evaluating clustering algorithms is an open problem, even when graphs without attributesor even tabular data are involved. This has been clearly discussed in recent surveys by Schaeffer (2007) and Fortunato (2010) where the identified problems not only concern the ambiguous and personal definition of *good cluster*, but also the need for results that are easier to interpret and use, benchmark datasets and quality functions to explain why a clustering is regarded as good or not.

While evaluating graph clustering is a hard and open problem even when no attributes are present, several measures to evaluate graph clusterings have been proposed, and some

have been extensively applied. Therefore, without claiming that these measures represent the final or only solution to the problem, in this section we start from them as an existing way of evaluating graph clustering and focus on what we need to add when we deal with attributed graphs.

The main additional aspect to consider when attributed graphs are involved is the co-existence of multiple objective functions. Having a description of the data that includes both structural and compositional aspects, we may have sets of nodes that are very similar according to their attributes but disconnected from each other. Similarly, we may have well connected sets of nodes with rather heterogeneous compositional attributes. Both cases can be considered good clusters depending on the user requirements and while we would certainly prefer to identify sets of nodes making a good cluster with respect to all these aspects, we must accept the co-existence of multiple evaluation functions — or a *multi-objective evaluation function*.

In the rest of this section, we introduce relevant evaluation measures for different aspects involved in defining good attributed graph clusters. In order to demonstrate their differences, we apply these measures to a toy graph.

4.1.1 Structural measures

Evaluating the quality of a clustering of a simple graph without node or edge attributes is a complex problem in itself. In this section, we will consider two different scenarios: evaluation with and without ground truth.

External evaluation measures. When ground truth is available, the problem is reduced to computing similarity between two clusterings. Since we confront the found structures to externally provided class information, we call such similarity measures *external evaluation measures*. These measures can be divided into two main groups: based on pair counting and based on information theory. We will briefly discuss the most typical representatives to give the readers an idea rather than a complete overview of the methods.

Given two partitions $\mathbf{C}_u = \{C_{u1}, C_{u2}, \ldots, C_{um}\}$ and $\mathbf{C}_v = \{C_{v1}, C_{v2}, \ldots, C_{vr}\}$ of a set of nodes, the pair-counting-based measures show the proportion of agreement between both partitions. These measures have two requirements: (1) the partitions are disjoint, i.e., $\bigcap_{C_i \in \mathbf{C}} C_i = \emptyset$, and (2) all elements have the same weight in the clustering process.

The Rand index (RI) is one of the first approaches for comparing two partitions (Rand, 1971). It can be considered as an alternative to accuracy because it expresses the number of pairs of nodes that were placed within the same group in both partitions divided by the number of all node pairs. This comparison leads to a similarity function $c(\mathbf{C}_u, \mathbf{C}_v)$ between partitions that is expressed as

$$c(\mathbf{C}_u, \mathbf{C}_v) = \sum_{i<j}^{n} \frac{\gamma_{ij}}{\binom{n}{2}}, \tag{6}$$

where

$$\gamma_{ij} = \begin{cases} 1 & \text{if } \exists k, k' : x_i, x_j \in C_{uk} \wedge x_i, x_j \in C_{vk'} \\ 1 & \text{if } \exists k, k' : x_i, x_j \notin C_{uk} \wedge x_i, x_j \notin C_{vk'} \\ 0 & \text{otherwise.} \end{cases}$$

The agreements between partitions $\mathbf{C}_u$ and $\mathbf{C}_v$ can be summarized using a contingency matrix as presented in Figure 11. In this matrix, n_{ij} is the number of agreements while $n_{i\cdot}$ is the number of elements of the ith group from the $\mathbf{C}_u$ partition and $n_{\cdot j}$ is the number of elements in the jth group in the $\mathbf{C}_v$ partition.

		$\mathbf{C}_v$				
	Class	v_1	v_2	$\dots$	v_r	$\sum_{i\cdot}$
	u_1	n_{11}	n_{12}	$\dots$	n_{1r}	$n_{1\cdot}$
	u_2	n_{21}	n_{22}	$\dots$	n_{2r}	$n_{2\cdot}$
$\mathbf{C}_u$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	u_m	n_{m1}	n_{m2}	$\dots$	n_{mr}	$n_{m\cdot}$
	$\sum_{\cdot j}$	$n_{\cdot 1}$	$n_{\cdot 2}$	$\dots$	$n_{\cdot r}$	n

Fig. 11. A contingency matrix representing the agreements n_{ij} between two partitions

Using a contingency matrix similar to the one presented in Figure 11, Equation 6 can be re-expressed as

$$c\left(\mathbf{C}_u,\mathbf{C}_v\right)=\frac{\binom{n}{2}-\left[1/2\left(\sum_i\left(\sum_j n_{ij}\right)^2+\sum_j\left(\sum_i n_{ij}\right)^2\right)-\sum\sum n_{ij}^2\right]}{\binom{n}{2}}. \tag{7}$$

Note that $c\left(\mathbf{C}_u,\mathbf{C}_v\right)\in[0,1]$, i.e., it is 0 when the partitions are dissimilar and 1 when the partitions are identical. Later Hubert & Arabie (1985) introduced the adjusted Rand index (ARI) which is a version of the Rand index corrected for chance. The ARI is given by

$$ARI\left(\mathbf{C}_u,\mathbf{C}_v\right)=\frac{\sum_{i=1}^r\sum_{j=1}^s\binom{n_{ij}}{2}-\left[\sum_{i=1}^r\binom{n_{i\cdot}}{2}\sum_{j=1}^s\binom{n_{\cdot j}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_{i=1}^r\binom{n_{i\cdot}}{2}+\sum_{j=1}^s\binom{n_{\cdot j}}{2}\right]-\left[\sum_{i=1}^r\binom{n_{i\cdot}}{2}\sum_{j=1}^s\binom{n_{\cdot j}}{2}\right]/\binom{n}{2}}, \tag{8}$$

where $n_{i\cdot}$, $n_{\cdot j}$ and n_{ij} are values taken from the contingency matrix in Figure 11.

Another common measure is the Jaccard index which is given by the ratio of the node pairs that were clustered together in both partitions and the node pairs clustered together in at least one partition (Jaccard, 1901).

The second group of external evaluation measures uses mutual information (MI) between partitions, i.e., the information both partitions share. These measures are based on entropy and joint entropy of the partitions. Using the same contingency matrix presented in Figure 11, the MI index is given by

$$MI\left(\mathbf{C}_u,\mathbf{C}_v\right)=\sum_{i=1}^m\sum_{j=1}^r\frac{n_{ij}}{n}\log\frac{n_{ij}/n}{n_{i\cdot}n_{\cdot j}/n^2}. \tag{9}$$

This measure can be normalized by the joint entropy of the partitions ensuring that the MI lies within the interval $[-1,1]$ or $[0,1]$. Variations of this measure with different normalizing factors or adjustments with correction for chance are presented in detail by Danon *et al.* (2005) and Vinh *et al.* (2010).

Internal evaluation measures. Without ground truth, determining the quality of a clustering is based on its intrinsic characteristics. We refer to such measures as *internal evaluation measures*. According to Ben-David & Ackerman (2008), "a clustering quality measure is a function that maps pairs of the form $(dataset, clustering)$ to some ordered set

(say, the set of non-negative real numbers), so that these values reflect how good or cogent that clustering is." Some general properties for good quality measures have been proposed, such as scale invariance, monotonicity and richness (Ben-David & Ackerman, 2008; van Laarhoven & Marchiori, 2013), but in practice the problem depends on the purpose of the analysis.

To assess quality, Gaertler (2005) uses two functions, $f(\mathbf{C})$ and $g(\mathbf{C})$, to measure, respectively, the density and the sparsity of the clustering. These functions are combined as follows

$$index(\mathbf{C}) = \frac{f(\mathbf{C}) + g(\mathbf{C})}{N(G)}, \tag{10}$$

where $N(G)$ is a normalization function for the index defined as $\max\{f+g\}$ over all clusterings (Brandes *et al.*, 2008). Using the general index defined in Equation 10, three different quality indices can be derived: coverage, conductance, and performance.

Coverage $\gamma(\mathbf{C})$ is a measure of the ratio of the intra-cluster weights to the total amount of edge weights:

$$\gamma(\mathbf{C}) = \frac{\omega(E(\mathbf{C}))}{\omega(E)}, \tag{11}$$

where $E(\mathbf{C})$ is the set of intra-cluster edges and $\omega(\cdot)$ is the sum of the weights of a set of edges. According to the general definition in Equation 10, $f = \omega(E(\mathbf{C}))$ and $g = 0$.

Conductance $\varphi(G)$ is a measure based on the observation that if a cluster is well connected, then a large number of edges have to be removed in order to bisect it. Thus, conductance $\varphi(G)$ of a graph G is the minimum conductance value over all cuts of G (Brandes *et al.*, 2008) — that is, the lowest possible value of the total weight of all edges between the clusters of a partition $\mathbf{C}$. Along with the *graph conductance*, two other measures exist: intra-cluster conductance $\alpha(\mathbf{C})$ and inter-cluster conductance $\delta(\mathbf{C})$. *Intra-cluster conductance* is the minimum conductance value over all induced subgraphs $G(C_i)$ while the inter-cluster conductance is the maximum conductance over all induced cuts $(C_i, \overline{C_i})$. Thus, given a cut $\mathbf{C} = (C, \overline{C})$, according to Brandes *et al.* (2008), the conductances $\varphi(C)$ and $\varphi(G)$ can be defined as follows:

$$\varphi(C) = \begin{cases} 1, & C \in \{\emptyset, V\} \\ 0, & C \notin \{\emptyset, V\} \wedge \overline{\omega(\mathbf{C})} = 0 \\ \frac{\overline{\omega(\mathbf{C})}}{\min\left(a(C), a(\overline{C})\right)}, & \text{otherwise} \end{cases} \tag{12}$$

$$\varphi(G) = \min_{C \subseteq V} \varphi(C), \tag{13}$$

where $a(C)$ is the sum of the weight over all edges adjacent to C. It is expressed as

$$a(C) = 2 \sum_{e \in E(C)} \omega(e) + \sum_{f \in E(C, \overline{C})} \omega(f).$$

The intra-cluster conductance of a partition $\mathbf{C}$ is defined as

$$\alpha(\mathbf{C}) = \min_{i \in \{1,\ldots,k\}} \varphi(G(C_i)), \tag{14}$$

while the inter-cluster conductance of a partition **C** as

$$\delta(\mathbf{C}) = \begin{cases} 1, & if\, \mathbf{C} = \{V\} \\ 1 - \max_{i \in \{1,\ldots,k\}} \varphi(C_i), & \text{otherwise.} \end{cases} \tag{15}$$

In order to express the preceding indices in the form of the general framework from Equation 10, we set $g = 0$ for intra-cluster conductance, $f = 0$ for inter-cluster conductance and $N = f + g = 1$ for both cases.

Performance defines the quality of a partition based on the "correctness" of the classification of a node pair. The density function f counts the number of edges within all clusters while the sparsity function g counts the "nonexistent edges" between clusters (Gaertler, 2005), that is, the number of *not* connected pairs of nodes among all clusters. The definitions are

$$\begin{aligned} f(\mathbf{C}) &= \sum_{i=1}^{k} |E(C_i)| \\ g(\mathbf{C}) &= \sum_{u,v \in V}^{k} [(u,v) \notin E]\, I_{i,j}(u,v), \end{aligned} \tag{16}$$

where the function I is defined as:

$$I_{i,j}(u,v) = \begin{cases} 1, & u \in C_i \wedge v \in C_j, i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

Finally, performance as presented by Brandes *et al.* (2008) is

$$\text{perf}(\mathbf{C}) = \frac{f(\mathbf{C}) + g(\mathbf{C})}{\frac{1}{2} n(n-1)}, \tag{18}$$

where n is the number of nodes of the graph.

A comparison of clustering algorithms and measures has been provided by Leskovec *et al.* (2010), and more details concerning the limitations of these measures can be found in the works by Gaertler (2005) and Brandes *et al.* (2008).

Other candidates for a quality measure are *density* and *modularity* (Newman & Girvan, 2004; Fortunato, 2010), and they can also be directly optimized instead of just being used as evaluation functions. We will not add additional details about modularity, that has already been described earlier in the article.

In Figure 12, some sample measures are illustrated on two alternative partitions of the same graph.

4.1.2 Edge-attributed graph clustering

Only a few works have proposed evaluation measures for multiple graphs. The measure introduced by Mucha *et al.* (2010) takes into account both the pairs of nodes and the pairs of graphs — this approach has already been described in Section 2.2.

A different approach is given by Boden *et al.* (2012). In the spirit of subspace clustering, a set of "interesting" non-redundant clusters is sought. Candidate multidimensional clusters are considered to be all the node sets that are densely connected in every respective dimension (in all single layers that are contained in the cluster). From these, the result

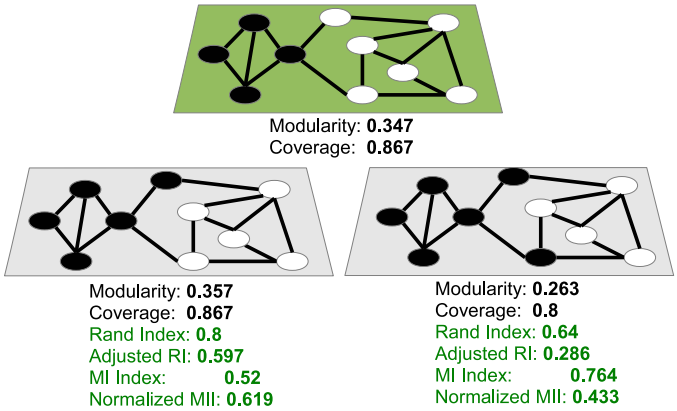


Fig. 12. Two graph partitions (lower graphs) and ground truth (upper graph): the values of some internal (modularity, coverage) and external evaluation measures (computed according to the ground truth) are shown

is selected by maximizing the quality sum $\sum_C Q(C)$ of all clusters while keeping the set of clusters non-redundant. Redundancy is computed as an overlap of edges of two clusters. The quality function $Q(C)$ is meant to be specified by users since it is application-dependent. Nevertheless, the authors provide a default quality function which multiplies average density of the layers, size and dimensionality. Additionally, a minimum cluster size is set to 8 nodes and a minimum of 2 dimensions is required for each cluster. This evaluation measure is bound to a specific cluster model. Moreover, it is limited to finding multi-dimensional clusters that are clustered in all the single layers at the same time (this results from the condition on the candidate clusters).

The problem of measuring distances between clusterings of graphs with weighted edges of multiple types is also tackled by Rocklin & Pinar (2011).

4.1.3 Node-attributed graph clustering

Node-attributed graph clustering approaches like the ones by Zhou *et al.* (2009), Cruz *et al.* (2014) and Dang & Viennet (2012) use a combination of two measures: density δ for the structural part and entropy $\mathscr{H}$ for the attributes. Given a graph $G(V,E)$ and a partition $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ of G, density is defined as:

$$\delta(\mathbf{C}) = \frac{1}{|E|} \sum_{C_i \in \mathbf{C}} |E(C_i)|, \tag{19}$$

where $E(C_i)$ is the set of edges that start and finish in the ith community. That is, density represents the proportion of edges that lie within the communities and a higher density corresponds to a better clustering.

The term *entropy*, used in several different contexts to measure the degree of disorder of a complex system, indicates the heterogeneity of the elements inside a cluster according to

their attribute values. It is given by

$$\mathscr{H}(\mathbf{C}) = \frac{1}{|V|} \sum_{C_i \in \mathbf{C}} H(C_i), \tag{20}$$

where $H(C_i)$ is the entropy of the ith community and is calculated as

$$H(C_i) = -\sum_{j=1}^{r} p_{ij} \ln p_{ij} + (1 - p_{ij}) \ln (1 - p_{ij}),$$

where r is the number of attributes and p_{ij} is the proportion of elements in the community C_i with the same value on the attribute j. The objective of the clustering is to reduce the entropy which is equivalent to increasing the homogeneity of the partition.

Another validation technique is presented by Li *et al.* (2008). In this work, documents are classified into ACM's 17 major computer science categories. This is a fuzzy classification that allows each document to belong to several categories. Thus, each document d_i is assigned to a (17-dimensional) topic vector z_i and then the documents are clustered into K groups. Each group C_j is further assigned to a topic vector Z_j.

The paper defines a measure called $\overline{PCS}$ as

$$\overline{PCS} = \frac{PCS_j}{K}, \tag{21}$$

where K is the number of communities, PCS_j is

$$PCS_j = \frac{\sum_{k:d_{jk} \in C_j} \eta(d_{jk})}{n_j},$$

where n_j is the size of the community j and

$$\eta(d_{jk}) = \begin{cases} 1 & \text{if } z_{jk} = Z_j \\ 0 & \text{otherwise} . \end{cases}$$

Thus for each cluster C_j, the measure computes the proportion of elements $d_k \in C_j$ such that $z_k = Z_j$, i.e., how many documents within the community have a topic vector that is equal to the community's topic vector.

In some cases it is possible to define the number and labels of the groups by hand as presented by Ge *et al.* (2008) where authors compare the obtained partition with the expected one by counting the number of elements classified correctly by an algorithm. This approach is acceptable for small networks but becomes prohibitive for large networks with high dimensional feature spaces.

When ground truth is available, it is possible to use validation methods such as Rand index or mutual information index. In this line, Combe *et al.* (2012) define a framework for comparing the resulting partition with the ground truth. They use a contingency matrix (similar to the one presented in Figure 11) created from the ground truth and a partition found by the tested algorithm. Then they calculate the proportion of nodes that were well grouped according to the ground truth.

Yang *et al.* (2009) use two validation approaches that are based on ground truth: the normalized mutual information (NMI), briefly described in Section 4.1.1, and the pairwise F measure (PWF). The PWF measure is given by the relation between pairwise *precision*

and *recall*. This relation is

$$PWF = \frac{\left(1+\beta^2\right) precision \times recall}{(\beta \times precision) + recall}, \tag{22}$$

where $\beta > 0$ is a parameter used to favor either precision or recall. It is common to leave $\beta = 1$. To calculate precision and recall, the following expressions are used

$$\begin{aligned} precision &= \frac{|S \cap T|}{|S|} \\ recall &= \frac{|S \cap T|}{|T|}, \end{aligned}$$

where S is the set of node pairs that are assigned to the same community and T is the set of node pairs that have the same label.

4.1.4 A multi-objective evaluation approach

In the previous sections we introduced several evaluation measures and we have seen that, in general, finding a good clustering of an attributed graph requires optimization of at least two objective functions. Therefore, there will always be a trade-off between compositional and structural dimensions. For node-attributed graphs, the objectives are the structural quality of the clusters (intra-cluster vs. inter-cluster edges) and the intra-cluster homogeneity of the node attributes. For edge-attributed graphs, the situation is more complicated since it is less obvious how to define a good clustering. According to Boden *et al.* (2012), cluster candidates are well clustered in all of their dimensions, but this assumption could prevent the discovery of potentially useful clusters.

Another possible evaluation perspective consists in no longer checking if a clustering is good as a whole, but whether any specific interesting clusters are found. In general, in order to evaluate a specific cluster in an attributed graph, one can take into consideration its structural quality, homogeneity of node attributes, size, dimensionality and novelty. We can thus see these variables as different dimensions of a search space where each multidimensional point is a cluster. Good clusters can be selected based on custom settings of weights of the dimensions, or unweighted approaches like the Pareto front can be used to find all clusters that are potentially better than others according to any combination of these basic evaluation functions.

For structural quality and node homogeneity, any measure from Sections 4.1.1 and 4.1.3 may be selected. To assess novelty, we suggest to use one of the proposed measures of overlap, such as Jaccard index. The value of the maximum overlap can be returned as novelty. In this way, emerging clusters of minimal dimensionality are favored, preventing information overload.

4.2 Applicability

Approaches preprocessing edge- or node-attributed graphs by reducing them to graphs without attributes normally keep the same asymptotic complexity of the clustering algorithm used after preprocessing. The exact complexity of the preprocessing phase depends on the data structure and the specific *flattening* algorithm, but it is normally achievable in close-to-linear time on the size of the graph. As an example, edge-attributed flattening

as described in Definition 3 and using a tree-based main memory indexing structure takes $\mathcal{O}(m \log m)$, where m is the number of edges, that is, the average number of edges per edge type times the number of edge types.

As such, while not taking full advantage of the information represented by the different edge types, these methods can be applied to very large graphs using any of the existing efficient clustering algorithms reported e.g. by Coscia *et al.* (2011), they are simple to implement and (with some variation in the flattening algorithm) can also be applied to directed and weighted graphs. However, in the case of weighted edge-attributed graphs, domain knowledge is necessary to decide how to merge weights on different edge-types. The conceptual problem of merging weights with different semantics as described by Magnani & Rossi (2013b), e.g., the number of exchanged messages on an email layer and the duration of friendship on a social media network, emphasizes the deficiencies of single-layer approaches.

Similarly, for node-attributed graphs where the node attributes are flattened into edge weights before applying a community detection algorithm, the time complexity of the preprocessing step depends on the number of attributes and on the method used to compute how similar the nodes are. In case of *matching similarity*, for each edge, the number of common attributes between the end nodes is computed which takes $\mathcal{O}(mf)$, where f is the attribute space size and where in general $f \ll m$. In high dimensional spaces we expect that each node is described by a sparse vector and that allows for efficient methods such as growing self-organizing maps. These methods, when coupled with efficient graph clustering, exhibit a near linear complexity. The other methods still take advantage of the sparse nature of the graph and thus, having less than quadratic complexity, are able to address large datasets.

With more integrated methods, such as subspace approaches or the one proposed by Ruan *et al.* (2013), the clustering process can reach a high complexity — quadratic and more. But in general, for linear combination or walk-based methods, the complexity depends on the algorithm used for clustering the features, e.g., SOM or k-means among others, and whether the approach is global or local. The resulting process can still be practically used for reasonably large graphs, and graphs with hundred thousand nodes have been successfully processed in the reviewed works on subspace clustering.

On the other hand, most of the community detection algorithms require the choice of parameters that control the output of the algorithm, for example the number of clusters k, the weight to emphasize the connectivity α or weighting variables for linear combination approaches, the number of iterations, statistical distributions for model-based methods, redundancy or heuristics in NP-hard subspace approaches; this sometimes requires major assumptions and domain knowledge about the data, which reduces their applicability. Only a few methods among the ones reported in this work are parameter-free, including the ones by Neville *et al.* (2003), Cruz *et al.* (2011a) and Akoglu *et al.* (2012).

Regarding the directionality of the edges, most of the methods described in this article rely on the application of existing approaches when the structural part of the graph must be analyzed, in which case any existing algorithm for directed graphs can be used. This evidently applies to the single-layer and weight-modification approaches, and is also the case for subspace methods, even if these last approaches may require some adaptation when specific algorithms have been hardcoded inside them. Methods based on extended

modularity cannot be used without modifications on directed graphs, because they are based on the original definition of modularity which assumes undirected edges. However, they can be extended in the same way as it has been done by Nicosia *et al.* (2009) for non-attributed graphs. With respect to node-attributed graphs, approaches based on linear combination can be straightforwardly used with directed edges as they are based on the computation of graph distances, that can be obtained on directed graphs as well. Similarly, walk-based approaches are naturally well suited to directed graphs.

As a final consideration, the works we have mentioned so far are all based on the general idea of clustering several dimensions at the same time; e.g., relationships, affiliation, competencies, socio-demographic features, among others. However, the information stored respectively in the attributes and in the edges may be uncorrelated and will not necessarily reinforce the same clusters. In practice, trying to merge several dimensions may result in failing to find any well separated clusters even when clusters exist under a single dimension. An alternative approach is to run dedicated and specialized clustering steps for each dimension (structure, edge attributes, etc.), and then integrate the resulting partitions *a posteriori* only if this leads to better clusters. Cruz & Bothorel (2013) propose to manipulate the partitions with a contingency matrix where structural groups are in rows and compositional ones are in columns. The integration of the partitions relies on predefined strategies. Even if matrix manipulation may not seem user-friendly, this original proposal is interesting from another perspective: according to their objectives, the analysts can try different combinations without re-computing the basic partitions and thus potentially save computational costs.

5 Open problems and discussion

Attributed graph clustering is an active research area, and as such it presents a number of open problems. In addition, being it an extension and combination of well established areas (graph clustering and multi-dimensional relational clustering), open problems can be classified into two main categories: 1) those already present when single graphs are considered (and the easier to identify) and 2) those specifically related to the combination of structure and attributes.

An example of the first category pertains to partitioning and overlapping algorithms. While the majority of graph clustering methods partition nodes into disjoint sets, many authors have pointed out that in real contexts individuals often belong to multiple communities. Even without considering attributes, this has motivated the development of several methods, such as the well-known clique percolation method by Palla *et al.* (2005), or the ones by Nicosia *et al.* (2009) and Wang *et al.* (2011) where extended versions of modularity are used to evaluate overlapping clusters. In their recent paper, Xie *et al.* (2013) review the state-of-the-art in overlapping community detection algorithms, quality measures, and benchmarks for non-attributed graphs. They provide a framework to evaluate the performances of both the community-level and node-level detection, and conclude that this research field is still work in progress, as more than 70% of the overlaps still remain uncovered. Other problems include how to measure the significance of overlapping nodes and how to interpret the resulting communities (Xie *et al.*, 2013). Recently, Yang *et al.* (2013) have used node attributed graphs for detecting overlapping communities, stating

that the resulting communities can be interpreted more easily by analyzing the attributes of the nodes belonging to each community. However, quality and interpretation issues are still open questions.

In general, like for other kinds of approaches, the presence of attributes introduces more parameters to be considered and requires the consideration of multiple aspects at the same time. However, in our opinion, when edge attributes are present, the dispute between partitioning and overlapping approaches should be reconsidered. In fact, overlapping is usually determined by participation in different networks: as an example, the same individual can be in her working team community, in her family community, in the community of her team mates at the fencing club, etc. This example suggests that if we can split our social network into a set of specialized networks (or, saying it in another way, if we can cluster our relationships into different classes), then we may find that some specialized networks only involve partitions. However, this consideration should not be understood as a statement against overlapping methods.

An example of the second category of open problems is the exponential explosion in the number of attribute value combinations to be considered during the clustering process. While this is a well-known problem in relational data mining, it is unknown in the domain of single graph clustering, and it is one of the main aspects reviewed in this article. In Section 2, we hypothesized that clusters can emerge when a specific combination of graphs is considered, and disappear when more graphs are added to the model. In Section 3, we discussed the notions of point of view and subspace clustering to counteract the fact that considering all the node attributes may lead to the *curse of dimensionality* problem. Furthermore, beyond the *quantitative* selection of a good subset of original data (which can be stated as a feature selection problem), scientists will have to take into account *qualitative* considerations: how to define the analysis context in order to decide how good a clustering is? How to make this context understandable to an analyst without domain knowledge and usable by a domain expert without deep analytical skills? How to conceive efficient techniques to present multiple results in real-time?

The main problem related to the existence of multiple points of view which is peculiar of graphs with edge labels (and more in general multiple interconnected graphs) is the existence of a large number of views, where every view corresponds to a specific combination of values on the edges. Despite some promising attempts to address this problem, inspired by the field of sub-space clustering, in the authors' opinion this aspect deserves a lot more research to be able to apply clustering algorithms to real on-line social networks. Given the intrinsic computational complexity of the problem, a possible direction involves the consideration of domain knowledge to focus the cluster discovery process on promising combinations of dimensions.

Initial work in this direction by Cruz *et al.* (2013) has defined control facilities to combine existing precomputed partitions. The objective is to offer tools to compare different approaches and visualize the results in a way that allows user feedback. The success of UCINET and — more recently — visual analytics software like Gephi and NodeXL is a sign that analysts are requesting such easy-to-apply tools. This requires advances focusing on usability, simplicity, efficiency and scalability, evaluation facilities such as comparison of methods, selection of relevant attributes and/or modeling. In fact, these

research directions are as meaningful in an attributed-graph context as they are for non-attributed graphs.

Understandably, early works on attributed graph clustering have focused on finding static communities, which is a preliminary and necessary step to study their evolution. Here researchers can partially reuse the same approaches used to find evolving communities on simple graphs, in particular the comparison of nodes clustered at different timestamps to identify evolutionary steps like *create*, *merge* and *split*. However, in the case of attributed graphs the evolution does not only regard the networks. The existence of multiple interconnected graphs and communities spanning some of them may also require a revision of the concept of evolutionary step.

A related problem that has generated a whole research sub-field in the realm of simple graphs is the study of network creation models. What are the forces leading to a specific network model exposing a modular structure? Rephrasing this question in the context of attributed graphs, how can we explain not only how some people have become densely interconnected, i.e., a cluster, but also why their attributes follow a specific value distribution and how these connections have developed in the different graph layers or edge types?

All the aspects mentioned so far highlight different levels of increasing complexity that we have to face when we consider attributes: the number of views to evaluate, the number of parameters to consider, e.g., in the evaluation functions, and the number of configurations of the system, e.g., the additional degrees of freedom in its evolution. A straightforward conclusion is that in the case of attributed graphs the applicability of forthcoming results may be strictly dependent on algorithmic advances, in particular regarding computational models like streaming, distributed, budget-based, approximate and incremental approaches enabling big data analysis.

Bibliography

Aggarwal, Charu, & Wang, Haixun. (2010). *Managing and Mining Graph Data*. Springer US.

Akoglu, Leman, Tong, Hanghang, Meeder, Brendan, & Faloutsos, Christos. (2012). Pics: Parameter-free identification of cohesive subgroups in large attributed graphs. *Pages 439–450 of: Sdm*. SIAM / Omnipress.

Atzmueller, Martin, & Mitzlaff, Folke. (2011). Efficient descriptive community mining. *Proc. 24th intl. flairs conference*. AAAI Press.

Balasubramanyan, Ramnath, & Cohen, William W. (2011). *Block-lda: Jointly modeling entity-annotated text and entity-entity links*. Chap. 39, pages 450–461.

Barabási, Albert-László, & Albert, Réka. (1999). Emergence of scaling in random networks. *Science*, **286**(5439), 509–512.

Barbieri, Nicola, Bonchi, Francesco, & Manco, Giuseppe. (2013). Cascade-based community detection. *Pages 33–42 of: Proceedings of the sixth acm international conference on web search and data mining*. WSDM '13. New York, NY, USA: ACM.

Ben-David, Shai, & Ackerman, Margareta. (2008). Measures of clustering quality: A working set of axioms for clustering. *Pages 121–128 of: Advances in neural information processing systems*.

Berlingerio, Michele, Coscia, Michele, & Giannotti, Fosca. (2011a). Finding and Characterizing Communities in Multidimensional Networks. *Pages 490–494 of: 2011 international conference on advances in social networks analysis and mining*. IEEE.

Berlingerio, Michele, Coscia, Michele, Giannotti, Fosca, Monreale, Anna, & Pedreschi, Dino. (2011b). Foundations of Multidimensional Network Analysis. *Pages 485–489 of: 2011 international conference on advances in social networks analysis and mining*. IEEE.

Berlingerio, Michele, Pinelli, Fabio, & Calabrese, Francesco. (2013). ABACUS: Apriori-BAsed Community discovery in mUltidimensional networkS. *Data Mining and Knowledge Discovery*, Springer, **27**(3).

Blondel, Vincent D., Guillaume, Jean-Loup, Lambiotte, Renaud, & Lefebvre, Etienne. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: Theory and experiment*, **2008**(10), P10008 (12pp).

Boden, Brigitte, Günnemann, Stephan, Hoffmann, Holger, & Seidl, Thomas. (2012). Mining coherent subgraphs in multi-layer graphs with edge labels. *Page 1258 of: Proceedings of the 18th acm sigkdd international conference on knowledge discovery and data mining - kdd '12*. New York, USA: ACM Press.

Bonchi, Francesco, Gionis, Aristides, Gullo, Francesco, & Ukkonen, Antti. (2012). Chromatic correlation clustering. *Page 1321 of: Proceedings of the 18th acm sigkdd international conference on knowledge discovery and data mining - kdd '12*. New York, USA: ACM Press.

Borgatti, Stephen P, Mehra, Ajay, Brass, Daniel J, & Labianca, Giuseppe. (2009). Network analysis in the social sciences. *Science*, **323**(5916), 892–5.

Brandes, Ulrik, Gaertler, Marco, & Wagner, Dorothea. (2008). Engineering graph clustering: Models and experimental evaluation. *Journal of experimental algorithmics*, **12**, 1–26.

Brodka, Piotr, Stawiak, Pawel, & Kazienko, Przemyslaw. (2011). Shortest Path Discovery in the Multi-layered Social Network. *Pages 497–501 of: 2011 international conference on advances in social networks analysis and mining*. IEEE.

Cai, Deng, Shao, Zheng, He, Xiaofei, Yan, Xifeng, & Han, Jiawei. (2005). Mining hidden community in heterogeneous social networks. *Pages 58–65 of: Proceedings of the 3rd international workshop on link discovery - linkkdd '05*. New York, New York, USA: ACM Press.

Combe, David, Largeron, Christine, Egyed-Zsigmond, Előd, & Géry, Mathias. (2012). Combining relations and text in scientific network clustering. *Pages 1280–1285 of: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*.

Coscia, Michele, Giannotti, Fosca, & Pedreschi, Dino. (2011). A classification for community discovery methods in complex networks. *Statistical analysis and data mining*, **4**(5), 512–546.

Cruz, Juan David, & Bothorel, Cécile. (2013). Information integration for detecting communities in attributed graphs. *Computational aspects of social networks (cason)*.

Cruz, Juan David, Bothorel, Cécile, & Poulet, François. (2011a). Entropy based community detection in augmented social networks. *Pages 163–168 of: Computational aspects of social networks (2011)*.

Cruz, Juan David, Bothorel, Cécile, & Poulet, François. (2011b). Semantic clustering of social networks using points of view. *Coria: conférence en recherche d'information et applications 2011*.

Cruz, Juan David, Bothorel, Cécile, & Poulet, François. (2012). Détection et visualisation des communautés dans les réseaux sociaux. *Revue d'intelligence artificielle*, **26**(4), 369–392.

Cruz, Juan David, Bothorel, Cécile, & Poulet, François. (2013). Integrating heterogeneous information within a social network for detecting communities. *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*.

Cruz, Juan David, Bothorel, Cécile, & Poulet, François. (2014). Community detection and visualization in social networks: Integrating structural and semantic information. *Acm trans. intell. syst. technol.*, **5**(1), 11:1–11:26.

Dang, The Anh, & Viennet, Emmanuel. (2012). Community detection based on structural and attribute similarities. *Pages 7–14 of: International conference on digital society (icds)*. ISBN: 978-1-61208-176-2.

Danon, Leon, Díaz-Guilera, Albert, Duch, Jordi, & Arenas, Alex. (2005). Comparing community structrue identification. *Journal of statistical mechanics: Theory and experiment*, **2005**(09), P09008.

Fortunato, Santo. (2010). Community detection in graphs. *Physics reports*, **486**(3-5), 75–174.

Fortunato, Santo, & Barthélemy, Marc. (2007). Resolution limit in community detection. *Proceedings of the national academy of sciences of the united states of america*, **104**(1), 36–41.

Freeman, Linton C. (1996). Cliques, galois lattices, and the structure of human social groups. *Social networks*, **18**(3), 173 – 187.

Gaertler, Marco. (2005). *Network analysis: Methodological foundations*. Springer Berlin / Heidelberg. Chap. Clustering, pages 178 – 215.

Gao, Jianxi, Buldyrev, Sergey V., Stanley, H. Eugene, & Havlin, Shlomo. (2011). Networks formed from interdependent networks. *Nature physics*, **8**(1), 40–48.

Ge, Rong, Ester, Martin, Gao, Byron J., Hu, Zengjian, Bhattacharya, Binay, & Ben-Moshe, Boaz. (2008). Joint cluster analysis of attribute data and relationship data: The connected k-center problem, algorithms and applications. *Acm trans. knowl. discov. data*, **2**(2), 7:1–7:35.

Getoor, Lise, & Diehl, Christopher P. (2005). Link mining: a survey. *Sigkdd explor. newsl.*, **7**(2), 3–12.

Goffman, Erving. (1974). *Frame analysis: an essay on the organization of experience*. Harper colophon books ; CN 372. New York: Harper & Row.

Gong, Neil Zhenqiang, Talwalkar, Ameet, Mackey, Lester W., Huang, Ling, Shin, Eui Chul Richard, Stefanov, Emil, Shi, Elaine, & Song, Dawn. (2011). Jointly predicting links and inferring attributes using a social-attribute network (san). *Corr*, **abs/1112.3265**.

Günnemann, Stephan, Boden, Brigitte, Färber, Ines, & Seidl, Thomas. (2013). Efficient mining of combined subspace and subgraph clusters in graphs with feature vectors. *Pages 261–275 of: Advances in knowledge discovery and data mining*. Springer.

Han, Jiawei, Kamber, Micheline, & Pei, Jian. (2011). *Data Mining: Concepts and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann.

Hanisch, Daniel, Zien, Alexander, Zimmer, Ralf, & Lengauer, Thomas. (2002). Co-clustering of biological networks and gene expression data. *Bioinformatics*, **18**(suppl 1), S145–S154.

Hric, Darko, Darst, Richard K., & Fortunato, Santo. (2014). *Community detection in networks: structural clusters versus ground truth*. *Phys. Rev. E*, **9**, 062805

Hubert, Lawrence, & Arabie, Phipps. (1985). Comparing partitions. *Journal of classification*, **2**, 193–218. 10.1007/BF01908075.

Jaccard, Paul. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la société vaudoise des sciences naturelles*, **37**, 547–579.

Karger, David R. (1993). Global min-cuts in rnc, and other ramifications of a simple min-out algorithm. *Pages 21–30 of: Proceedings of the fourth annual acm-siam symposium on discrete algorithms*. SODA '93. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Kazienko, Przemysaw, Brodka, Piotr, Musial, Katarzyna, & Gaworecki, Jarosaw. (2010). Multi-Layered Social Network Creation Based on Bibliographic Data. *Pages 407–412 of: 2010 ieee second international conference on social computing*. IEEE.

Kim, Myunghwan, & Leskovec, Jure. (2012). Multiplicative attribute graph model of real-world networks. *Internet mathematics*, **8**(1-2), 113–160.

Kivelä, Mikko, Arenas, Alexandre, Barthelemy, Marc, Gleeson, James P., Moreno, Yamir, & Porter, Mason A. (2014). Multilayer Networks. *Journal of complex networks*, 1–69.

Kohonen, Teuvo. (1997). *Self-organizing maps*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Kossinets, Gueorgi, & Watts, Duncan J. (2006). Empirical analysis of an evolving social network. *Science*, **311**(5757), 88–90.

La Fond, Timothy, & Neville, Jennifer. (2010). Randomization tests for distinguishing social influence and homophily effects. *Pages 601–610 of: Proceedings of the 19th international conference on world wide web*. WWW '10. New York, NY, USA: ACM.

Lancichinetti, Andrea, & Fortunato, Santo. (2011). Limits of modularity maximization in community detection. *Physical review. e, statistical, nonlinear, and soft matter physics*, **84**(6 Pt 2).

Lazega, Emmanuel, & Pattison, Philippa E. (1999). Multiplexity, generalized exchange and cooperation in organizations: a case study. *Social networks*, **21**(1), 67–90.

Leskovec, Jure, Kleinberg, Jon, & Faloutsos, Christos. (2005). Graphs over time: densification laws, shrinking diameters and possible explanations. *Pages 177–187 of: Proceedings of the eleventh acm sigkdd international conference on knowledge discovery in data mining*. ACM.

Leskovec, Jure, Lang, Kevin J., & Mahoney, Michael. (2010). Empirical comparison of algorithms for network community detection. *Pages 631–640 of: Proceedings of the 19th international conference on world wide web*. WWW '10. New York, NY, USA: ACM.

Li, Huajing, Nie, Zaiqing, chien Lee, Wang, Giles, C. Lee, & rong Wen, Ji. (2008). Scalable community discovery on textual data with relations. *In proceedings of the acm conference on information and knowledge management (cikm 2008)*.

Li, WJ, & Yeung, DY. (2009). Relation regularized matrix factorization. *Pages 1126–1131 of: Ijcai-09*. IJCAI'09. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Liu, Yan, Niculescu-Mizil, Alexandru, & Gryc, Wojciech. (2009). Topic-link lda: Joint models of topic and author community. *Pages 665–672 of: Proceedings of the 26th annual international conference on machine learning*. ICML '09. New York, NY, USA: ACM.

Magnani, Matteo, & Rossi, Luca. (2011). The ML-model for multi layer network analysis. *Ieee international conference on advances in social network analysis and mining*. IEEE Computer Society, Los Alamitos.

Magnani, Matteo, & Rossi, Luca. (2013a). Formation of multiple networks. *Pages 257–264 of: Social computing, behavioral-cultural modeling and prediction*. Springer.

Magnani, Matteo, & Rossi, Luca. (2013b). Pareto Distance for Multilayer Network Analysis. *Pages 249–256 of: Social computing, behavioral-cultural modeling and prediction*. Springer.

Magnani, Matteo, Micenkova, Barbora, & Rossi, Luca. (2013). Combinatorial Analysis of Multiple Networks. *arxiv preprint*.

Minor, Michael J. (1983). New directions in multiplexity analysis. *Applied network analysis*, 223–244.

Moise, Gabriela, Zimek, Arthur, Kröger, Peer, Kriegel, Hans-Peter, & Sander, Jörg. (2009). Subspace and projected clustering: experimental evaluation and analysis. *Knowledge and information systems*, **21**(3), 299–326.

Moser, Flavia, Colak, Recep, Rafiey, Arash, & Ester, Martin. (2009). *Mining cohesive patterns from graphs with feature vectors*. Chap. 50, pages 593–604.

Mucha, Peter J, Richardson, Thomas, Macon, Kevin, Porter, Mason A, & Onnela, Jukka-Pekka. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *Science*, **328**(5980), 876–8.

Nan Du, Wang, Hao, & Faloutsos, Christos. (2010). Analysis of Large Multi-modal Social Networks: Patterns and a Generator. Balcázar, José Luis, Bonchi, Francesco, Gionis, Aristides, & Sebag, Michèle (eds), *Machine learning and knowledge discovery in databases*. Lecture Notes in Computer Science, vol. 6321. Berlin, Heidelberg: Springer Berlin Heidelberg.

Neville, Jennifer, Adler, Micah, & Jensen, David D. (2003). Clustering relational data using attribute and link information. *Proceedings of the workshop on text mining and link analysis, eighteenth international joint conference on artificial intelligence*.

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review e*, **69**(2), 026113+.

Nicosia, Vincenzo, Mangioni, Giuseppe, Carchiolo, Vincenza, & Malgeri, Michele. (2009). Extending the definition of modularity to directed graphs with overlapping communities. *Journal of statistical mechanics: Theory and experiment*, **2009**(03), P03024.

Noh, Jae Dong, & Rieger, Heiko. (2004). Random walks on complex networks. *Physical review letters*, **92**(11), 118701.

Palla, Gergely, Derényi, Imre, Farkas, Illés, & Vicsek, Tamás. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, **435**(7043), 814–818.

Pei, Jian, Jiang, Daxin, & Zhang, Aidong. (2005). On mining cross-graph quasi-cliques. *Page 228 of: Proceeding of the eleventh acm sigkdd international conference on knowledge discovery in data mining - kdd '05*. New York, USA: ACM Press.

Pool, Simon, Bonchi, Francesco, & Leeuwen, Matthijs van. (2014). Description-driven community detection. *Acm trans. intell. syst. technol.*, **5**(2), 28:1–28:28.

Rand, William M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the american statistical association*, **66**(336), pp. 846–850.

Rissanen, Jorma. (1983). A universal prior for integers and estimation by minimum-description lenght. *Annals of statistics*, **11**(2), 416 – 431.

Rocklin, Matthew, & Pinar, Ali. (2011). On Clustering on Graphs with Multiple Edge Types. *arxiv preprint*, Sept.

Rossetti, Giulio, Berlingerio, Michele, & Giannotti, Fosca. (2011). Scalable Link Prediction on Multidimensional Networks. *Pages 979–986 of: 2011 ieee 11th international conference on data mining workshops*. IEEE.

Ruan, Yiye, Fuhry, David, & Parthasarathy, Srinivasan. (2013). Efficient community detection in large networks using content and links. *Pages 1089–1098 of: Proceedings of the 22nd international conference on world wide web*. WWW '13.

Schaeffer, Satu Elisa. (2007). Graph clustering. *Computer science review*, **1**, 27–64.

Shi, Jianbo, & Malik, Jitendra. (2000). Normalized cuts and image segmentation. *Pattern analysis and machine intelligence, ieee transactions on*, **22**(8), 888–905.

Silva, Arlei, Meira, Jr., Wagner, & Zaki, Mohammed J. (2010). Structural correlation pattern mining for large graphs. *Pages 119–126 of: Proceedings of the eighth workshop on mining and learning with graphs*. MLG '10. New York, NY, USA: ACM.

Skvoretz, John, & Agneessens, Filip. (2007). Reciprocity, multiplexity, and exchange: Measures. *Quality & quantity*, **41**(3), 341–357.

Stein, Benno, & Niggemann, Oliver. (1999). On the nature of structure and its identification. *Pages 122–134 of: Graph-theoretic concepts in computer science*. Springer.

Steinhaeuser, Karsten, & Chawla, NiteshV. (2008). Community detection in a large real-world social network. *Pages 168–175 of:* Liu, Huan, Salerno, JohnJ., & Young, MichaelJ. (eds), *Social computing, behavioral modeling, and prediction*. Springer US.

Sun, Yizhou, Han, Jiawei, Aggarwal, Charu C., & Chawla, Nitesh V. (2012). When will it happen? *Page 663 of: Proceedings of the fifth acm international conference on web search and data mining - wsdm '12*. New York, USA: ACM Press.

Tang, Lei, Wang, Xufei, & Liu, Huan. (2011). Community detection via heterogeneous interaction analysis. *Data mining and knowledge discovery*, **25**(1), 1–33.

Tong, Hanghang, Faloutsos, Christos, & Koren, Yehuda. (2007). Fast direction-aware proximity for graph mining. *Pages 747–756 of: Proceedings of the 13th acm sigkdd international conference on knowledge discovery and data mining*. KDD '07. New York, NY, USA: ACM.

van Laarhoven, Twan, & Marchiori, Elena. (2013). *An axiomatic study of objective functions for graph clustering*. Tech. rept. CoRR, abs/1308.3383.

Villa-Vialaneix, Nathalie, Olteanu, Madalina, & Cierco-Ayrolles, Christine. (2013). Carte auto-organisatrice pour graphes étiquetés. *Page Article numéro 4 of: Atelier Fouilles de Grands Graphes (FGG) - EGC'2013*.

Vinh, Nguyen Xuan, Epps, Julien, & Bailey, James. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. mach. learn. res.*, **11**(Dec.), 2837–2854.

Wang, Bing, Cao, Lang, Suzuki, Hideyuki, & Aihara, Kazuyuki. (2011). Epidemic spread in adaptive networks with multitype agents. *Journal of physics a: Mathematical and theoretical*, **44**(3), 035101.

Wang, Jianyong, Zhou, Zhiping, & Lizhu, Zeng;. (2006). CLAN: An Algorithm for Mining Closed Cliques from Large Dense Graph Databases. *Pages 73–73 of: 22nd international conference on data engineering (icde'06)*. IEEE.

Wasserman, Stanley, & Faust, Katherine. (1994). *Social Network Analysis: Methods and Applications*. Structural analysis in the social sciences, 8, vol. 8, no. 1. Cambridge University Press.

Xie, Jierui, Kelley, Stephen, & Szymanski, Boleslaw K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*, **45**(4), 43.

Xu, Zhiqiang, Ke, Yiping, Wang, Yi, Cheng, Hong, & Cheng, James. (2012). A model-based approach to attributed graph clustering. *Pages 505–516 of: Proceedings of the 2012 acm sigmod international conference on management of data*. SIGMOD '12. New York, NY, USA: ACM.

Yang, Jaewon, McAuley, Julian, & Leskovec, Jure. 2013 (Dec). Community detection in networks with node attributes. *Pages 1151–1156 of: Data mining (icdm), 2013 ieee 13th international conference on*.

Yang, Shuang-Hong, Long, Bo, Smola, Alex, Sadagopan, Narayanan, Zheng, Zhaohui, & Zha, Hongyuan. (2011). Like like alike: joint friendship and interest propagation in social networks. *Pages 537–546 of: Proceedings of the 20th international conference on world wide web, www*. ACM.

Yang, Tianbao, Jin, Rong, Chi, Yun, & Zhu, Shenghuo. (2009). Combining link and content for community detection: a discriminative approach. *Pages 927–936 of: Kdd '09: Proceedings of the 15th acm sigkdd international conference on knowledge discovery and data mining*. New York, NY, USA: ACM.

Yin, Zhijun, Gupta, Manish, Weninger, Tim, & Han, Jiawei. (2010a). Linkrec: a unified framework for link recommendation with user attributes and graph structure. *Pages 1211–1212 of: Proceedings of the 19th international conference on world wide web*. WWW '10. New York, NY, USA: ACM.

Yin, Zhijun, Gupta, M., Weninger, T., & Han, Jiawei. (2010b). A unified framework for link recommendation using random walks. *Pages 152–159 of: Advances in social networks analysis and mining (asonam), 2010 international conference on*.

Zhao, Peixiang, Li, Xiaolei, Xin, Dong, & Han, Jiawei. (2011). Graph cube: on warehousing and olap multidimensional networks. *Pages 853–864 of: Proceedings of the 2011 acm sigmod international conference on management of data*. ACM.

Zheleva, Elena, Sharara, Hossam, & Getoor, Lise. (2009). Co-evolution of social and affiliation networks. *15th acm sigkdd conference on knowledge discovery and data mining (kdd)*.

Zhiping Zeng, Jianyong Wang. (2006). Coherent closed quasi-clique discovery from large dense graph databases. *Proceedings of the 12th acm sigkdd international conference on knowledge discovery and data mining - kdd '06*

Zhou, Yang, Cheng, Hong, & Yu, Jeffrey Xu. (2009). Graph clustering based on structural/attribute similarities. *Proc. vldb endow.*, **2**(1), 718–729.

Zhou, Yang, Cheng, Hong, & Yu, Jeffrey Xu. (2010). Clustering large attributed graphs: An efficient incremental approach. *Pages 689–698 of: Data mining (icdm), 2010 ieee 10th international conference on*. IEEE.

View publication stats