

Mining communities and their descriptions on attributed graphs: a survey

Martin Atzmueller · Stephan Günnemann ·
Albrecht Zimmermann

Abstract Finding communities that are not only relatively densely connected in a graph but that also show similar characteristics based on attribute information has drawn strong attention in the last years. There exists already a remarkable body of work that attempts to find communities in vertex-attributed graphs that are relatively homogeneous with respect to attribute values. Yet, it is scattered through different research fields and most of those publications fail to make the connection. In this paper, we identify important characteristics of the different approaches and place them into three broad categories: those that select *descriptive attributes*, related to clustering approaches, those that enumerate *attribute-value combinations*, related to pattern mining techniques, and those that identify conditional attribute weights, allowing for pre-processing. We point out that the large majority of these techniques treat the same problem in terms of attribute representation, and are therefore interchangeable to a certain degree. In addition, different authors have found very similar algorithmic solutions to their respective problem.

1 Introduction

Graphs are a powerful mechanism to represent data. Applications range from social networks, over gene analysis, to smart sensor systems. Due to the ubiquitous nature of graphs, analyzing them is a highly active research field with clustering/community detection being one of the most important and frequently applied tasks. While classical graph clustering approaches have considered merely structural information, in recent years attributed graph clustering has gained strong at-

Martin Atzmueller
Tilburg University, Cognitive Science and Artificial Intelligence, Tilburg, The Netherlands.
Present: Osnabrück University, Institute of Computer Science, Osnabrück, Germany.
E-mail: martin.atzmueller@uni-osnabrueck.de

Stephan Günnemann
Technical University of Munich, Department of Informatics & Institute for Advanced Study,
Munich, Germany. E-mail: guennemann@in.tum.de

Albrecht Zimmermann
Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France.
E-mail: albrecht.zimmermann@unicaen.fr

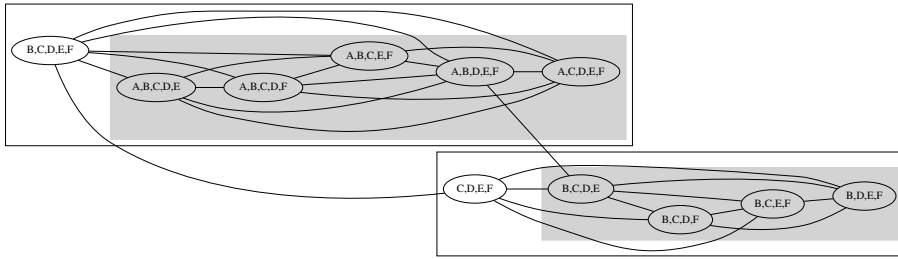


Fig. 1 Attributed graph with natural communities (indicated by rectangular box) and describable communities (grey background)

attention: it integrates additional attribute data about individual instances into the clustering task, to enhance its result. In a social network, e.g., the attributes describing each user’s characteristics might be combined with the underlying friendship network to form an attributed graph.¹ Figure 1 shows an example, in which each vertex is, for the sake of presentation, labeled with a set of items.

In the last few years, a number of clustering approaches for attributed graphs have been introduced. The discussion in the related work sections of publications on the topic tends to focus on whether different methods allow finding overlapping communities or not, or considers the technical methodology of the approaches (e.g., distance-based, model-based, random walk-based, etc.). In this survey, we choose a different way of looking at this issue based on the following observation: There are essentially two ways of exploiting attribute values. 1) to improve community detection by leveraging attribute value similarities, and 2) to derive a concrete *description* of discovered communities. The latter one enables us to better understand the structure of the detected communities, i.e. in order to answer the question why this set of vertices is a reasonable community. This is particularly relevant not only relating to *interpretability* but also given the recently renewed focus on *explainable* results of data analysis processes.

To return to the example shown in Figure 1, the rectangular boxes show two communities — two groups of vertices that are strongly connected to each other but that have few connections among each other. It can also be seen, however, that neither of those communities can be described by only one single set of items. The communities marked within grey rectangles, on the other hand, are still strongly connected within, weakly connected to each other, and describable with the items “A” (top-left community) and “B” (bottom-right community), respectively. Clearly, those are not the only two describable communities. It is easy to see that we can further find subcommunities which are described by more complex descriptions, e.g., considering the set of items “A, B” focussing on the four central vertices of the top-left community. Notably, the set of items “B, C” describes *two* communities, one on the left of the upper rectangle, and one in the center of the lower one, showing that descriptions are not necessarily unique to communities. Another example, referring to two communities described “C, D” is given in Figure 2.

¹ It is worth mentioning that the term ‘attributed graph’ is not the only term used by the scientific community. Other terms include, e.g., ‘graphs with feature vectors’, ‘labeled graphs’, or ‘annotated graph’, where also ‘network’ is often substituted for ‘graph’.

The approaches we discuss in this work do this at different levels of explicitness. There are approaches that identify for each community the *attribute-value combinations* that describe the community, returning ready-made descriptions. Additionally, there are approaches that explicitly identify *attributes* for which all vertices in a community have the same or similar values, without, however, also explicitly returning those values themselves. Finally, there are methods that derive *indicators* for the importance that attributes have for different communities but that would require post-processing of those indicators to enumerate the attributes. We discuss all three of those approaches. *In summary, the main focus of this survey are methods that explicitly treat attributes and therefore (can) derive descriptive communities.*

Indeed, this is in marked contrast to the survey by Bothorel et al. [2015], which discusses works that exploit the attribute information in graphs for improving clustering results, i.e. improving community *detection* — by calculating distances or by augmenting density information. That is, the primary goal of these methods is to improve clustering performance by using multiple data sources — not to find *descriptive* communities. This also means that attributes are not treated explicitly but the information that is contained in them is mixed with the information inherent in the network. This takes the form of, for instance, defining quality measures that also take attribute similarity into account. Because the returned results of those approaches are only the communities, and no information about the contribution of the attributes is included, even post-processing might then not result in community descriptions. Therefore, these are outside the scope of our survey.

We start our discussion by introducing fundamental definitions in Section 2, followed by a concrete description of our selection methodology in Section 3. We then continue with an in-depth survey and categorization of description-oriented approaches in Section 4. Next, we briefly touch the aspect of evaluation and graph generation for attributed graphs in Section 5 before Section 6 concludes the survey with a summary and an outlook on further promising research directions.

2 Definitions

In the following, we outline and summarize fundamental definitions on graphs and communities.

Definition 1 (Graph) A *graph* is a tuple $G = \langle V, E \rangle$, where V is a set of *vertices* and E a set of *edges* $E \subseteq V \times V$. We refer to the number of edges a vertex $v \in V$ is incident to as the vertex’ *degree*, $\deg(v) = |\{(u, v) \in E \mid u \in V\}|$.

Definition 2 (Attributed Graph) An *attributed* graph is a graph G in which each $v \in V$ is associated to a vector of attribute values $\mathbf{x} = (x_1, \dots, x_d)$, and each edge $e \in E$ to a vector $\mathbf{y} = (y_1, \dots, y_t)$. We use $a_i(v)$ to refer to the i th attribute value of a vertex v , and $a_i(e)$ for the edge e respectively. We denote with A_V the set of vertex attributes, $d = |A_V|$, and with A_E the set of edge attributes, $t = |A_E|$. If $|A_V| > 0, |A_E| = 0$, we refer to G as *vertex-attributed*; similarly if $|A_V| = 0, |A_E| > 0$, we call it *edge-attributed*. If $|A_V| = 0, |A_E| = 0$, then we refer to a plain graph.

Note, that this definition subsumes the widely used labeled graph definition, in which each vertex has a label, and each edge a label or a weight, as a special case.

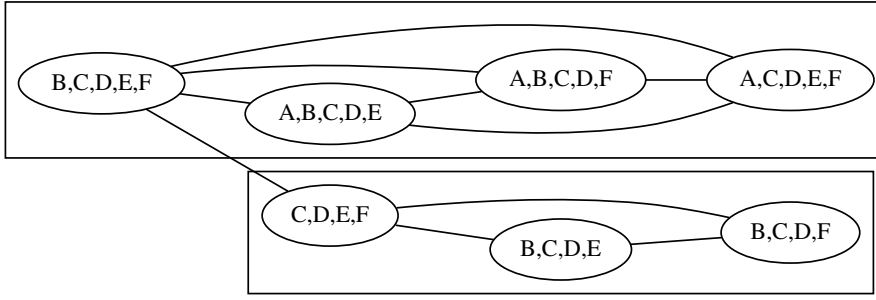


Fig. 2 Projection of vertices of the graph shown in Figure 1 labeled with “C,D”

Definition 3 (Projected Graph) Given a set of vertex attributes A_V , an attributed graph G , a description $p = \{A_1 \sqcap val_1, \dots, A_d \sqcap val_d\}$ with $A_i \in A_V$, $val_j \in dom(A_j)$, and $\sqcap \in \{<, \leq, =, \geq, >\}$, a projected graph G_p is defined as follows: the subgraph $G_p = \langle V_p, E_p \rangle$, $V_p = \{v_j \in V \mid a_i(v_j) \sqcap val_i\}$, $E_p = \{(u, v) \in E \mid u \in V_p, v \in V_p\}$, is referred to as the projected graph according to description p .

The graph shown in Figure 2 depicts the result of projecting the graph shown in Figure 1 on the description “C,D”. The projection acts as a filter on the vertices, and creates two communities that can both be described by a single set of items, which we also call an itemset.

Definition 4 (Graph partition) A *partition* of a graph G is a set of sets of vertices $P_G = \{C_1, \dots, C_k\}$, with $C_i \cap C_j = \emptyset$, and $\bigcup_i C_i = C$; the individual C_i are also referred to as *clusters* or *communities*. The *external (internal) degree* of a vertex v refers to the number of edges connecting it to vertices in other (the same) communities:

- $deg_{ext}(v) = |\{(u, v) \in E \mid v \in C_i, u \in C_j, i \neq j\}|$,
- $deg_{int}(v) = |\{(u, v) \in E \mid v \in C_i, u \in C_i\}|$.

This definition is equivalent to the standard community detection definition, in which it is assumed that vertices can belong to a single community only, and that the graph is partitioned w.r.t. vertices, not w.r.t. edges. A consequence of the latter is that edges can have end points belonging to different communities, a characteristic that is exploited in calculating the quality of communities. When the assumption of strict vertex membership is relaxed, we refer to *overlapping* communities.

Overall, how to define communities is a rather complex topic, on which no consensus has been reached yet in the literature. We do not discuss all possible aspects but refer the interested reader to Fortunato [2010]. An often enforced requirement is connectedness.

Definition 5 (Path) Given a graph G , a *path* of length $p \in \mathbb{N}$ between two vertices v, u is a list of edges $\langle (v_1, v_2), (v_2, v_3), \dots, (v_p, v_{p+1}) \rangle$ for distinct vertices $v_i, v_i \in V$, i.e. $v_i \neq v_j, i \neq j$, with $v = v_1, u = v_{p+1}$.

Definition 6 (Connectedness/Reachability) A community $C \subseteq V$ is considered *connected* if and only if there is a path between any two vertices $v, u \in C$. The *n-reachability* of a community derives from the existence of a path of maximally length n between any two vertices in the community.

Yet given that reachability requirements could be satisfied by chains of vertices, stronger connectivity requirements are often imposed, such as that vertices need to form a *k-core* [Seidman, 1983].

Definition 7 (k-Core) A community C is referred to as a *k-Core* if and only if $deg_{int}(v) \geq k$ for every $v \in C$, i.e. each vertex is adjacent to at least k vertices of the community, and the community is *maximal*, i.e. one cannot add additional vertices without violating that property.

A sufficient criterion for communities, finally, is that they are not only connected but have more internal connections than external ones, focusing on the *density*. This is in general related to the notion of *density*, e.g., [Charikar, 2000, Diestel, 2006], where we focus on *edge density* differentiating between edges *internal/external* to a given community.

Definition 8 ((Edge) Density) Given a community V_i , its *intra-community density* is the ratio of existing internal edges to the maximum possible number of internal edges:

$$\delta_{int}(C_i) = \frac{|\{(u, v) \mid u, v \in C_i\}|}{|C_i|(|C_i| - 1)/2} = \frac{\sum_{v \in C_i} deg_{int}(v)}{|C_i|(|C_i| - 1)}.$$

Its *inter-community density* is the ratio of existing edges external edges to possible external edges:

$$\delta_{ext}(C_i) = \frac{|\{(u, v) \mid u \in C_i, v \in C_j, i \neq j\}|}{|C_i||V \setminus C_i|} = \frac{\sum_{v \in C_i} deg_{ext}(v)}{|C_i||V \setminus C_i|}.$$

Such criteria can be absolute, using the definition above with a threshold, but also relative. There are too many different measures for relative density to list them here, which is why we only mention the widely-known *modularity*.

Definition 9 (Modularity) The *modularity* [Newman, 2004, Newman and Girvan, 2004] of a graph clustering with k communities $C_1, \dots, C_k \subseteq V$ focuses on the number of edges *within* a community and compares that with the *expected* such number given a null-model (i.e., a corresponding random graph where the vertex degrees of G are preserved). It is given by

$$Modularity(C_1, \dots, C_k) = \frac{1}{2m} \sum_{i=1}^k \sum_{u, v \in C_i} A_{u, v} - \frac{deg(u) deg(v)}{2m},$$

where $A_{u, v}$ is the entry of the adjacency matrix referring to vertices u and v , and m is the number of edges of the whole graph.

Modularity has been used as optimization criterion driving a number of different classical community detection algorithms, i.e. ones not taking attribute information into account.

When it comes to communities in attributed graphs, finally, structural density is not enough but vertices should also agree with respect to attributes, which can be assessed using a *cohesion* function [Moser et al., 2009], for instance.

Definition 10 (Cohesion function) A *cohesion function* is a function

$$f : P(V) \times P(A_V) \times \mathbb{R} \mapsto \{\text{true}, \text{false}\}$$

This function is required to satisfy both a maximality characteristic, i.e. for any set of vertices V' and set of attributes A'_V , the latter contains *all* attributes for which V' is cohesive,

$$\begin{aligned} (f(V', A'_V, \theta_s) = \text{true} \wedge \nexists A''_V \supset A'_V : f(V', A''_V, \theta_s) = \text{true}) \Rightarrow \\ (f(V', A^*_V, \theta_s) = \text{true} \Rightarrow A^*_V \subseteq A'_V), \end{aligned}$$

and an anti-monotonicity characteristic, i.e. given a set of vertices and a set of attributes that are cohesive, any subsets of those attributes/vertices stay cohesive:

$$f(V', A'_V, \theta_s) = \text{true} \Rightarrow f(V'', A''_V, \theta_s) = \text{true}, \forall V'' \subseteq V', A''_V \subseteq A'_V$$

Moser et al. [2009] also provide a concrete example of such a definition:

$$f(V', A'_V, \theta_s) = \forall A_i \in A'_V : |\max_{v \in V'} a_i(v) - \min_{v \in V'} a_i(v)| \leq \theta_s$$

As an illustration, consider Figure 4: assuming $\theta_s = 0.2$, A'_V for the upper shaded community would be $\{A, B\}$, and for the lower shaded one $\{A, B, C\}$.

3 Scope and Overview: Algorithm Selection and Categorization

The numerous techniques that are capable of putting a concrete description on discovered communities rely on the following mechanisms: (a) descriptions drive community detection — they are explicitly enumerated and restrict the vertices that can be used to form communities, (b) communities drive description formation — only those attribute values appearing for vertices in a community can be used, or (c) vertex and attribute membership probabilities for communities are optimized together. The first two approaches are not necessarily exclusive: as we will see later, some methods iterate between the two.

Hence, there are different options for *constructing* a description. Following the local pattern mining view [Hand, 2002, Morik, 2002, Morik et al., 2005], we focus on *attributes*, and *attribute values*; and a description combines these in a suitable way, e. g., by a conjunction, disjunction, or combination thereof. Also, please note that such descriptions (patterns) induce local structures that can be regarded as the result themselves, or can be integrated into a global approach that partitions the complete (graph) data space.

In this paper, we intend to explore these issues in detail, drawing explicit connections between the different methods, in the same spirit as has been done in [Novak et al., 2009] for supervised rule induction. A comparison between [Pool

et al., 2014] and [Galbrun et al., 2014], for example, has been reported in the latter, showing that the description language and discriminative learning of the former leads to rather different results. However, the remaining techniques that we consider (see below), notwithstanding their similarities, have not been compared against each other before.

3.1 Algorithmic Selection Criteria

Our selection methodology is based on different aspects of description-oriented approaches, focusing on ideas from community detection and local pattern mining. For the latter, we first need to consider what makes up a local pattern. For that, we take some ideas and definitions from local pattern detection [Hand, 2002, Morik, 2002, Morik et al., 2005] which we also illustrate with an example below: According to Hand [2002] a local pattern can be regarded as a data vector exhibiting an anomalously high local density of data points compared to a background model. A local pattern has two important characteristics [Hand, 2002, Klösgen, 2002] — exemplified by the gray boxes in Figure 3: (1) Local patterns cover small parts of the data space. (2) Local patterns deviate from the distribution of the population of which they are part. This deviation is usually measured by interestingness measures that contrast their behavior with that of the entire data or of other patterns. As a simple illustration, consider Figure 3: item “A” occurs in five vertices, item “B” in only four but the set of items “A,B” in four vertices out of 11. The expected frequency of that set of items is $(5 \cdot 4)/11 = 1.81$, so its observed frequency deviates clearly from the background distribution. The expected distribution of “C, D”, on the other hand, is $(9 \cdot 9)/11 = 7.36$ and its observed distribution 7, it can therefore be regarded as not local.

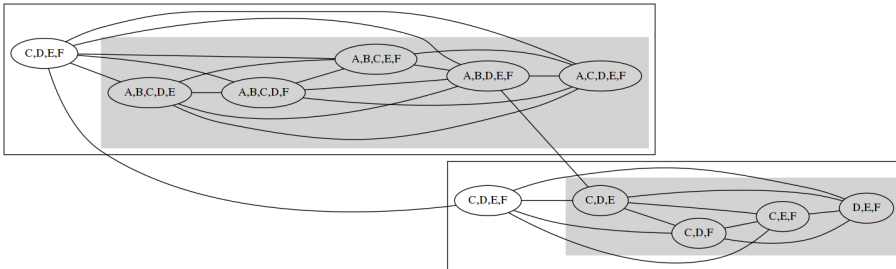


Fig. 3 Attributed graph with a community describable by a local, discriminative description (top), and one describable by a non-local, non-discriminative one (bottom)

Therefore, in an unsupervised view on local pattern detection, no information but the data itself is given to find out what patterns may be present in the database. In contrast, a supervised view exploits some information about a concept of interest, or some target distribution in order to identify interesting patterns. Then, a local pattern can be regarded as a subgroup, for example, covering a set of instances that contrasts the global model, cf. Morik [2002]. If we consider the

edge distribution to be the target distribution, “A,B” is also a local pattern from a supervised perspective since the described community is denser than expected.

Thus, our main focus in this survey is on techniques that have two important aspects in common: (1) Each algorithm identifies a subset of attribute dimensions, i.e. attributes or attribute-values, that are relevant for the detected communities. (2) These subsets can be mapped to individual communities and their respective induced subgraphs (according to the idea of a local pattern).

While communities (i.e. set of vertices) are local structures more or less by definition, different categories how to handle the attributes have been proposed. We are specifically interested in local methods where the focus is on subsets of attribute dimensions that are locally relevant. Also, we focus on methods that create concise attributive descriptions, in contrast to those approaches for which the derived descriptions often only take the form of certain values appearing in the *majority* of vertices in a community, instead of *all* of them.

Based on these intuitions we can identify three possible categories of algorithms/methods, allowing different potential for interpretation/description:

1. *Description via (explicit) attribute selection*: Considering Figure 4, such a method would select $\{A, B\}$ for the upper shaded community because their values are rather close for all vertices, as well as $\{A, B, C\}$ for the lower one.
2. *Description via (explicit) attribute-value selection*: Considering Figure 4, such a method could find the description $A \geq 0.75 \wedge B = 0.75$ for the upper shaded community, for instance, and $B \leq 0.3 \wedge C \leq 0.65 \wedge C \geq 0.5$ for the lower one.
3. *Description via implicit attribute selection/attribute weighting*, i.e. *post-processing* algorithmic output w.r.t. attributes: For Figure 4, such a method could for instance derive the following weights:

Attribute	Upper communities		Lower communities	
	complete	shaded	complete	shaded
A	1.18	5	∞	∞
B	2	∞	3.33	6.67
C	1.11	1.25	2.5	6.67
D	1.11	1.43	1.11	1.11
E	1.11	1.11	1.11	1.43
F	1.11	1.43	1.43	1.43

If we apply a threshold of 3 to select relevant attributes, the non-shaded upper community cannot be described at all, the shaded one by attributes “A” and “B”, the lower non-shaded one with “A” and “B”, and the shaded one with “A”, “B”, and “C”.

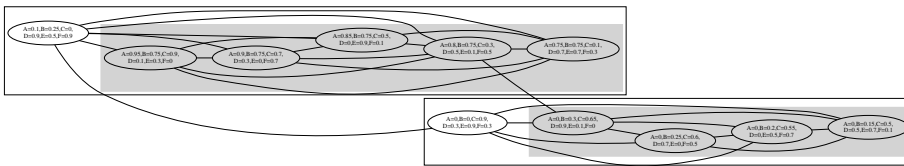


Fig. 4 Projection of vertices of the graph shown in Figure 4 labeled with “C,D”

Reference	Attribute Selection	Attr.-Value Selection	Post-processing
Atzmueller and Mitzlaff [2010, 2011], Atzmueller et al. [2016] Atzmueller et al. [2018, 2019]		✓	
Boden et al. [2012, 2013]	✓	✓	
Du et al. [2017]	✓		
Galbrun et al. [2014]		✓	
Günemann et al. [2010, 2011, 2012, 2013a,b,c]	✓		
Kalofolias et al. [2019]		✓	
Moser et al. [2009]	✓		
Pool et al. [2014]		(✓)	
Sánchez et al. [2013]	✓		
Silva et al. [2012]		✓	
Soldano and Santini [2014], Soldano et al. [2015, 2017]		✓	
Balasubramanyan and Cohen [2011]			✓
Baldominos et al. [2017]			✓
Li et al. [2010]			✓
Martínez-Seis [2017]			✓
McCallum et al. [2006]			✓
Newman and Clauset [2016]			✓
Revelle et al. [2015]			✓
Steinhaeuser and Chavla [2008]			✓
Smith et al. [2016]			✓
Wang et al. [2016]			✓
Xu et al. [2012]			✓
Yang et al. [2013]			✓

Table 1 Categorization of algorithms (references in alphabetical order) based on the selection criteria. The top part shows the selected methods based on the inclusion criteria (1 and 2), while the bottom includes the algorithms which will be discussed in less detail in Section 4.3.

As we will outline in Section 4.3, the third option differs from the first two in that attributes are not explicitly selected. Instead, all methods discussed in that section derive some kind of indicator for attributes that could be post-processed to create description.

Table 1 provides an overview of all considered techniques according to this selection methodology in the order as discussed above, i.e. (1) attribute selection, (2) attribute-value selection, and (3) postprocessing. Here, the methods in the upper part (and the respective algorithms) will be discussed in detail in Sections 4.1 and 4.2, and the ones in the lower part in Section 4.3, including explicit and more implicit methods and options for postprocessing.

3.2 Algorithmic Categorization

Considering the above selection of approaches based on their type of description, we provide in the following sections a more detailed categorization of the first two groups according to different criteria. While this section presents an overview on the given criteria, the next section summarizes and categorizes the techniques in more detail.

The first three subcategories concern the informativeness of the descriptions:

1. Does the technique select explicit attribute values as part of the description? All techniques surveyed in detail select a subset, or subspace, of attributes that are specific to the given communities. Not all of them also select the attribute *values* that describe the community. While those can usually be extracted in a post-processing step, given the community and the relevant attribute subspace, selecting values allows to present the user with communities and their actual descriptions directly.
2. Can found communities overlap? The ability to mine overlapping communities gives additional flexibility and therefore a higher chance to find high-quality results. On the other hand, this can lead to redundancy among communities and reduce interpretability.
3. Does the technique identify *local* patterns as descriptions, according to the criteria given in Section 3.1?
 - In addition, we assess whether found descriptions are *discriminative*, i.e. whether they are found by contrasting different communities, or, in other words, whether knowing any of the descriptions allows one to recover a particular community. Notably, a non-local description will not be discriminative but a local one will not automatically help to discriminate between communities.

To illustrate this sub-characteristic, we can again consider Figure 3. “A,B”, is discriminative in that this description occurs in all vertices of the upper highlighted community and only there. “C,D”, on the other hand, while correctly describing the lower highlighted community, also occurs in other vertices.

Additional categories concern the applicability of the techniques:

4. In which language are descriptions enumerated? Most commonly, description languages are sets of attributes, or conjunctions of attribute-value pairs, but more expressive languages are also possible.
5. Does the technique work on discrete attribute values, continuous ones, or both?
6. Are attributes considered on vertices, edges, or both?
7. Does the technique consider a single graph, or does it allow for multi-layer graphs/multiplex networks?

Finally, techniques can traverse the search space either *heuristically* or in an *exact* manner, trading off execution speed against qualitative guarantees. A summary of the approaches and their corresponding characteristics is presented in Table 2.

Reference(s)	Algorithm	Explicit AV	Overlapping	Locality	Heuristic	Attributed V/E	Language	Discrete/Continuous
Atzmueller et al. [2016], Atzmueller and Mitzlaff [2011, 2010]	COMODO	✓	✓	✓ _{disc}	×	both	conjunctive	both
Atzmueller et al. [2018, 2019]	MinerLSD	✓	✓	(✓)	×	vertices	conjunctive	discrete
Boden et al. [2013, 2012]	MIMAG	✓	✓	✓	×	edges, multi-layer vertices	conjunctive	continuous
Du et al. [2017], Sun et al. [2018]	OSCom	×	✓	×	×	vertices	conjunctive, disjunctive, or cardinality-based matching	both
Galbrun et al. [2014]	LDense/Spectral/Pivot	✓	✓	✓	(✓)	vertices	conjunctive	discrete
Günemann et al. [2013c, 2010]	GAMER	✓	✓	✓	×	vertices	conjunctive	continuous
Günemann et al. [2013a]	EDCAR	✓	✓	✓	✓	vertices	conjunctive	continuous
Günemann et al. [2012, 2011]	DB-CSC	×	✓	✓	×	vertices	sets of attributes	continuous
Günemann et al. [2013b]	SSCG	×	×	×	✓	vertices	sets of attributes	continuous
Kalofolias et al. [2019]	RoSi	✓	✓	✓ _{disc}	×	vertices	conjunctive	both
Moser et al. [2009]	CoPaM	×	✓	×	×	vertices	conjunctive	both
Pool et al. [2014]	DCM	✓	✓	✓ _{disc}	✓	vertices	disjunction of conjunctions	both
Sánchez et al. [2013]	ConSub	×	✓	✓	×	vertices	conjunctive	continuous
Silva et al. [2012]	SCPM	✓	✓	×	×	vertices	conjunctive	discrete
Soldano and Santini [2014], Soldano et al. [2015, 2017]	ParameterLC / MinerLC	✓	✓	×	×	vertices	conjunctive	discrete

Table 2 Detailed algorithmic categorization of the algorithms discussed in Sections 4.1 and 4.2

4 Survey on Relevant Algorithms

In the following three subsections, we describe the selected techniques in more detail. We focus mainly on the first four characteristics (1.-4.) since the applicability criteria (4.-8.) do not lend themselves to much interpretation, and add some information about the traversal strategy. The order of the discussed techniques will be chronological, allowing the reader to follow the methodological developments.

4.1 Attribute Selection

We subdivide techniques according to whether they select attribute values or not, and the first class of techniques identifies attribute subspaces that are relevant for particular communities but not the values of those attributes, which could however be derived in a post-processing step.

CoPaM Moser et al. [2009] propose to mine so-called cohesive patterns. A cohesive pattern is a tuple of a set attributes D and a subgraph $G = (V, E)$ that fulfills three criteria: (1) D satisfies a cohesion function, (2) G is dense, and (3) G is connected.

To find patterns, the approach first removes all non-cohesive edges, i.e. edges for which the vertices violate the cohesion function. The resulting connected components are processed independently in an APRIORI-like manner, joining edges until they violate the cohesive pattern constraint, i.e. the approach is community-driven. Attribute subspaces are identified via a *maximal* attribute subspace for a subgraph that is still cohesive. This implies several characteristics of the found patterns: as mentioned above, while attribute subspaces are explicitly selected, attribute values are not. Since edges are joined, it is possible that communities (vertex-)overlap. Furthermore, because attribute subspaces are chosen without recurrence to the full graph or even communities resulting from the same connected community, it is neither guaranteed that they are local, nor that they are discriminative.

GAMER Günnemann et al. [2010, 2013c] enhance the above principle by taking the possible redundancy of subgraphs into account. While CoPaM reports all maximal dense subgraphs — which might overlap to a high extent — the works by Günnemann et al. [2010, 2013c] focus on finding a *set* of non-redundant dense subgraphs with maximal interestingness. Here, interestingness can be any function taking the density, size, or number of attributes of the subgraph into account. Furthermore, the attributes of the community need to satisfy a cohesion function. To find clusters, a set enumeration tree operating on the set of vertices is exploited. The tree is traversed in a best-first approaches leading to an exact, non-heuristic solution.

EDCAR Günnemann et al. [2013a] use the same modeling approach as the work above. In contrast, however, they exploit a heuristic search principle, thus leading to much better scalability. More precisely, the set enumeration tree is explored via the GRASP (Greedy Randomized Adaptive Search) principle.

DB-CSC Deviating from the above scenario that the attribute values of a cluster are bounded by a specific interval, Günnemann et al. [2012, 2011] propose a density-based cluster definition. More precisely, in the selected attribute subspace the cluster needs to follow the well-known DBSCAN [Ester et al., 1996] clustering definition; while in the graph space an extension of k-cores has been proposed. Again, a *set* of non-redundant clusters is generated. Since DBSCAN allows to find arbitrarily shaped clusters, no specific attribute values selection is provided per cluster. For finding the clusters, an apriori-like search principle combined with fixed-point iteration is exploited: starting with 1-dimensional clusters, higher dimensional clusters are iteratively constructed. Within each subspace, clusters can be detected via a fixpoint iteration. The subspaces are neither contrasted to the overall distribution, nor to other communities.

SSCG Günnemann et al. [2013b] extend the principle of spectral clustering to find subspace clusters in attributed graphs. Following the idea of subspace clustering, each cluster is associated with an individual set of relevant attributes. The selected attributes subsequently determine the similarity/weight of two adjacent vertices; that is, the affinity matrix used in spectral clustering is no longer static but depends on the selected subspaces. Overall, since neither the subspaces nor the clusters are known, both aspect are learned in a joint fashion by minimizing the so-called normalized subspace cut — an extension of the normalized cut. The approach does not identify local patterns but optimizes a global model. Since solving this optimization problem is NP-hard, the authors propose an approximative alternating optimization scheme.

ConSub Sánchez et al. [2013] use a Monte Carlo process to generate interval constraints on vertex attributes, which are used to create projected subgraphs. If the number of edges in the subgraph is higher than expected, a *congruent* subspace and corresponding subgraph has been found. To derive larger attribute subspaces, the authors propose a bottom-up, APRIORI-like approach, similar to Günnemann et al. [2010]. The authors view their approach rather as dimensionality reduction to make community (outlier) detection more effective.

There are two common threads to the techniques described so far: 1) descriptions drive community discovery, and 2) vertex attributes' values' *similarity* are considered, either via explicit thresholding or via clustering.

OSCom Starting from ego-networks, Du et al. [2017], Sun et al. [2018] apply a metric-based greedy strategy for detecting a set of subnetworks based on the respective attributed neighborhood, i.e. the common attributes. After that, subcommunities are extracted, forming an overall supergraph. Finally, global semantic communities are identified on this supergraph.

MIMAG Orthogonal to the above works that mostly consider vertex-attributed graphs, Boden et al. [2013, 2012] focus on edge-attributed graphs. Similar to the work of Günnemann et al. [2010] they build on extensions of quasi-cliques (i.e. $\delta_{int} \geq 0.5$), now taking multiple graph layers into account and finding descriptions operating on the edge attributes. They propose a joint set enumeration tree to efficiently generate the communities in an informed best first search.

4.2 Attribute-Value Selection

The second class of techniques identifies both attributes and their relevant values directly. This obviates the need for a post-processing step of the discovered patterns to discover the appropriate values for the description. These techniques in many cases also use descriptions to drive community detection directly in order to establish a mapping between attribute dimensions and induced subgraph. The presented techniques below are somewhat younger than those proposed in the preceding section, and not surprisingly, there are clear connections to existing (local) pattern mining approaches.

SCPM Silva et al. [2012] binarize attributes, allowing them to treat attribute-value combinations as items, and apply frequent itemset mining to find promising candidates. By projecting the graph on the itemset, certain vertices will be removed, and the remaining connected components can be checked for the satisfaction of a minimum density constraint. By calculating upper bounds on the structural correlation of itemsets, the pruning capabilities of the approach are enhanced. Clearly, overlap is entirely possible for the communities found by this approach. In addition, while frequent patterns have been considered the first instance of local patterns in the literature, there is in fact no locality as such — a frequent pattern can be so frequent that it applies to different sections of the network. The literature on frequent patterns includes quite many examples of interestingness measures that relate the frequency of a pattern to background models [Vreeken and Tatti, 2014].

ParaminerLC / MinerLC Soldano and Santini [2014] take this approach towards the logical conclusion in terms of frequent itemset mining, mining *closed frequent itemsets* as candidate descriptions. As in Silva et al. [2012], the graph is projected and connected components identified. A difference to the older technique is the use of the Galois operator on the *candidate community*, refining both community and description further. Both enumeration options, descriptions driving community discovery and communities driving description enumeration, are therefore interleaved.

A follow-up work [Soldano et al., 2015] turns the approach into an iterative one, treating found communities as networks in which sub-communities should be found. The similarity to the preceding approach means that Soldano *et al.* also inherit the limitations, such as the lack of true locality, while they also apply a different definition of local (abstract) patterns; essentially, they add the idea of graph abstractions which lead to further constrained subnetworks where communities are identified, as described above. This is implemented in the MinerLC algorithm (as an adaptation of the ParaminerLC algorithm) for undirected but also regarding directed networks [Soldano et al., 2017] and further graph abstractions. If a (strong) constraining graph abstraction constraint is applied (e. g., a k -core [Seidman, 1983] constraint, where $k > 1$), then MinerLC basically focuses on those (locally) induced (constrained) subgraphs, thus advancing on purely frequent pattern based approaches for community detection on attributed networks. There are further extensions, e. g., regarding two-mode attributed networks [Soldano et al., 2019] with according constraints as well.

DCM Instead of starting from the description side, as the approaches discussed above, Pool et al. [2014] start with communities (as groups of vertices). The space of possible communities is larger than that of (conjunctive) descriptions, which means that they have to use a heuristic approach to find high-scoring ones, as is usual in community detection. Concretely, the approach starts from basic community candidates and greedily adds/removes vertices to improve a community score. Once those candidates stabilize, a pattern mining approach is used to find discriminative conjunctive patterns that predict vertices' community membership. For each community, corresponding patterns are combined into a *disjunction*. This gives DCM a much richer description language than other methods discussed in this section. Vertices matching the description are included in the community (and non-matching ones removed). Since this will result in changes to the communities, the process is iterated until the community structure remains unchanged.

To ensure interpretability and control redundancy, the top- k communities are selected in a post-processing step, scored by a measure trading off community quality and description complexity, and controlled by a redundancy threshold on the Jaccard-similarity between communities vertices. The use of the discriminative pattern miner results in local patterns, and the redundancy threshold can be used to control community overlap — typically some overlap will be accepted.

Spectral, LDense, Pivot Galbrun et al. [2014] also consider the problem of finding a *set* of at most k communities in a labeled graph, the cumulative densities of which are maximal. Vertices are described by labels, i.e. by words. Since a *bag-of-words* shows the same characteristics as an *itemset*, the two problem settings are interchangeable. After translating their problem into the *generalized maximum coverage problem* and showing guarantees for a greedy algorithm that always adds the community having highest residual density, they propose three different techniques for finding the best community. To control redundancy, edges already included in communities are removed between iterations but can be re-added in later iterations to improve the formed communities.

- One of the three techniques, SPECTRAL, begins with calculating a similarity matrix between attribute-values, using Jaccard over vertices having the respective attribute values as similarity measure. Using the Laplacian of this matrix, attribute values are ordered according to the fiedler vector, and continuous intervals in this ordering considered to identify candidates for communities. The set of communities found by this approach can be vertex-overlapping but edge-overlap is explicitly excluded. Descriptions are not compared to those of other communities or the background graph.
- Next, LDENSE, greedily — i.e. heuristically — adds labels such that the corresponding vertex set has the highest density, until the description becomes too specific and matches no vertices anymore. Among the vertex sets formed during this search process, the densest (and its description) is included into the solution set.
- The third approach, PIVOT, heuristically forms communities, and after formation greedily constructs the description best matching it. As in [Pool et al., 2014], vertices are then added and removed according to whether they match the description or not.

COMODO Atzmueller et al. [2016], propose a technique that explicitly aims at identifying local patterns. Inspired by subgroup discovery methods [Atzmueller, 2015], their approach exhaustively enumerates conjunctions of attribute-value tests, and calculates (standard) community quality measures such as the modularity [Newman and Girvan, 2004], the segregation index [Freeman, 1978], or the inverted average out degree fraction [Yang and Leskovec, 2012] on the corresponding communities, using upper bounds/optimistic estimates of these measures to aid in pruning. Comodo returns the top-k community patterns, with an optional redundancy check using a minimal improvement filter, e.g., Bayardo et al. [2000]. The use of the community quality measure implies discriminative descriptions, such that a description covering several communities (components) receives a low score.

Atzmueller [2016] applies the algorithm also to more complex community quality functions for anomaly detection on labeled edges. The measures used to score descriptions compare community densities to that of the entire graph, satisfying the locality property. [Atzmueller et al., 2016, Atzmueller, 2016] build on [Atzmueller and Mitzlaff, 2010, 2011], which used fewer measures. These papers precede the other works discussed in this section.

MinerLSD Atzmueller et al. [2018, 2019] combine central ideas of the *COMODO* [Atzmueller et al., 2016] and *MinerLC* [Soldano et al., 2015, 2017] algorithms for explicitly mining closed local patterns into the *MinerLSD* algorithm. It focuses both on local pattern mining, applying the standard local modularity metric [Newman, 2004, Atzmueller et al., 2016] as possible for *COMODO*. In addition, *MinerLSD* can utilize graph abstractions which reduce graphs to k-core subgraphs [Soldano et al., 2015] for enabling further graph (interestingness) constraints. Then, local patterns are identified in a similar way as for *MinerLC*, while the applied community measure (local modularity) also favors discriminative descriptions as for *COMODO*. In particular, in order to prevent the typical pattern explosion in pattern mining, *MinerLSD* employs closed patterns. Then, the top-k patterns or those above a certain local modularity threshold are returned.

RoSi Kalofolias et al. [2019] apply the same approach as *Comodo* — treat community detection as subgroup discovery, let description enumeration drive discovery, use optimistic estimates — but propose a different, k-core based measure to discover more robust communities.

With the exception of *DCM*, all techniques in this section have very much in common with each other. *SCPM*, *ParaminerLC*, *Spectral*, *LDense*, and *Pivot* all use an itemset representation. While the conjunctions of attribute-value combinations used by *Comodo*, *MinerLSD* and *RoSi* would give them more flexibility in the case of numerical attributes, for discrete attributes these can be translated into items, as *SCPM* shows.

Most of the methods also let descriptions drive community discovery, although *DCM* and *ParaminerLC* interleave the two processes to a certain degree, and *Pivot* also starts from communities.

Reference(s)	Algorithm	Overlapping	Technique	Attributed V/E	Discrete/ Continuous
Balasubramanian and Cohen [2011]	Block-LDA	×	Block model + LDA	V	discrete
Baldominos et al. [2017]	-	×	Louvain	V	discrete
Li et al. [2010]	TTR-LDA-Community model	×	Topic model	V	discrete
Martínez-Seis [2017]	RELNA	✓	Generative model	V	discrete
McCallum et al. [2006]	GT Model	✓	Topic Model	E	discrete
Newman and Clauset [2016]	-	×	Stochastic block model	V	discrete
Qin et al. [2018]	ASCD	✓	NMF	V	discrete
Revelle et al. [2015]	SENG	✓	Topic model	V	continuous
Smith et al. [2016]	B/T-CME	×	Random walk	V	continuous
Steinhaeuser and Chavla [2008]	-	×	Clustering	V	continuous
Wang et al. [2016]	SCI	✓	NMF	V	discrete
Xu et al. [2012]	BAGC	×	Clustering	V	discrete
Yang et al. [2013]	CESNA	✓	Generative model	V	discrete

Table 3 Algorithmic categorization of the algorithms discussed in Sections 4.3

4.3 Attribute-Guided Graph Mining (Post-processing possibilities)

There are a number of techniques that employ option three in Section 3.1, i.e. that utilize descriptive information for mining attributed graphs but do not explicitly select attributes or attribute values directly. The post-processing necessary would therefore be more extensive than in the case of the methods described in Section 4.1. Strictly speaking, most of these methods fall into the class of algorithms only exploiting attribute information to improve community detection that are described by Bothorel et al. [2015].

They differ from those methods that integrate attribute information via combined similarity functions or by introducing virtual vertices, however. In the former case, inverting the function to derive attribute importance is far from obvious, and in the latter there may be parts of a community that depend not at all on attribute vertices and others that fall apart if one removed these vertices.

Whereas the methods described in Section 4.2 explicitly enumerate both attributes and their values, and those in Section 4.1 at least return the attributes that need to be processed, the techniques in this section calculate the relative importance of attributes and this information has to be post-processed to derive descriptions. We therefore discuss most of these techniques in less detail, giving more attention to those that demonstrated this kind of post-processing.

We summarize the different methods in Table 3, indicating whether overlapping communities can be found, the used algorithmic technique, whether the method considers vertex or edge attributes, and whether attributes are discrete or continuous.

4.3.1 Explicit post-processing

We start with methods including explicit post-processing options.

GT model We begin with the work of McCallum et al. [2006], which has several interesting characteristics: 1) this is, to the best of our knowledge, the first such work, 2) they consider attributes on edges, not vertices, and 3) they explicitly post-process their results to retrieve the most relevant attributes. Concretely, they consider edges to be labeled with words, equivalent to items, and employ a topic model taking both labels and group membership into account. They extract the five to eight (depending on experimental setting) most relevant words from the topic model.

Block-LDA Balasubramanian and Cohen [2011] combine block models with LDA to estimate both community membership and conditional topic distributions. By Gibbs sampling fifteen terms per community, they recover the most relevant terms.

CESNA Yang et al. [2013] use a model in which each vertex has community affiliation probabilities. Those affiliation probabilities predict both edges between vertices and attribute values, and the formation process consists of estimating those affiliations in such a way that they align with the edges and attributes observed in the data. Vertices are annotated with words or phrases, by exploiting the estimated conditional attribute weights, the authors extract the top attributes per community.

SENC Revelle et al. [2015] use a topic model for finding relevant topics for communities, as well as vertex membership probabilities, and employ an EM algorithm to optimize the two. They assume that vertices are described by words but differ from other work in using *term weights* (*TFIDF*), i.e. switching from an itemset-like setting to one of numerical values. In the experimental evaluation, they present the top-40 terms per community according to learned conditional probabilities.

SCI Wang et al. [2016] employ non-negative matrix factorization (NMF). Vertices are described by bags-of-words, or itemsets, and the objective function combines topology and attribute similarity, using a trade-off parameter. As a result of their formulation, one of the derived matrices encodes the relationships between communities and attributes, which they exploit to extract the top-10 words.

ASCD The work of Qin et al. [2018] differs only to a small degree from *SCI*, mainly due to a focus on the fact that topology and shared attribute values can *disagree*, requiring the ability to fine-tune the trade-off between the two. They also extract the top-10 words.

4.3.2 Post-processing left to the user

Finally, we focus on approaches which do not include explicit post-processing, but leave that to the user for potentially extracting descriptions from the discovered communities. Steinhäuser and Chavla [2008] annotate edges with the similarity between vertices' attribute values, and group them into communities by thresholding those similarity values. By post-processing those communities, one could identify those attributes for which vertices are similar, as well as their values but the descriptions could be rather general. Li et al. [2010] first form communities using the Girvan-Newman method [Girvan and Newman, 2002], and then identify relevant topics using Latent Dirichlet Allocation. Community detection is not informed by descriptions, and communities are not adjusted afterwards, however, meaning that descriptions could be unreliable or non-existent. Xu et al. [2012] propose building an MAP model over vertex attribute values to cluster vertices. While one could use model values to identify the most relevant attributes for each cluster, this is not an output of the approach. Smith et al. [2016] use a random-walk based method for identifying communities, and derive weights for attribute values based on their frequency in the network and the visitation frequency of the random walker. Those walks could be used to identify the description corresponding to a community in post-processing. Newman and Clauset [2016] use a Bayesian modeling technique based on stochastic block models for estimating community allocations including structural and attributive information, however no description is targeted. Baldominos et al. [2017] find *stereotypes* from communities detected using a modularity-optimizing algorithm by weighting labels according to the proportion of vertices in the community that support them. Conversely, Martínez-Seis [2017] use homophilic principles for obtaining a ranking of the attributes and then only apply those for community detection.

5 Evaluation and attributed graph generation

A glaring issue for finding descriptions of communities is evaluation. It is already a difficult challenge in the case of classical community detection because the ground truth is often not known, and evaluating whether the description of a community is appropriate is arguably even harder. There are some benchmark graph generators for creating plain networks, e.g., [Lancichinetti et al., 2008, Baldesi et al., 2018, Bojchevski et al., 2018], however, these do not take attribute information into account when creating the respective graphs.

Attributed graph generators aim to generate graphs following natural properties, e.g., power-law behaviour of the degree distribution. Existing works cover extensions of preferential attachment models [Zheleva et al., 2009, Lee et al., 2015], stochastic block models [Newman and Clauset, 2016], or sampling approaches [Robles-Granda et al., 2016]. Kaytoue et al. [2017] generate attributed graphs incorporating connected components with attributive structure. Furthermore, Seratosa [2018] presents a methodology for generating pairs of attributed graph with a bounded graph edit distance, focussing on graph matching problems. Note that all these approaches for generating attributed graphs usually do not explicitly model communities. That is, the 'true' community structure will not be known for the generated graphs. In contrast, Largeron et al. [2015, 2017] introduce a graph generator for attributed graphs that is able to incorporate community structure in the generated attributed graphs. However, the work does not propose how to find these communities.

It is worth noting that all attributed graph clustering models based on probabilistic generative models (e.g. Kim and Leskovec [2011], Yang et al. [2013], Xu et al. [2014]) could in principle also be used for generating data; usually, however, they are used for inference only.

6 Conclusion

Even though community detection in attributed graphs, and more concretely detecting communities and their descriptions together, is still a relatively young research direction, progress has been quick and a variety of mature techniques exist already. Surveying those approaches, we have identified three main families, one which employs subspace clustering ideas, i.e. identifying those attribute-subspaces for which communities occur in the graph/network, a second one that adapts ideas developed in local pattern mining, and a third one that identifies the conditional importance of attributes in certain communities.

For the first class of methods, this allows exploiting the rich set of clustering techniques developed over several decades of research to address the similarity question in the attribute space, giving those approaches both high flexibility and good running times. Accordingly, multiple established clustering notions such as degree-based clustering, spectral clustering, or density-based clustering have been transferred to the attributed graph domain.

The second class of techniques has undergone the same progression as previous forms of pattern mining: starting from frequent patterns, via condensed representations, to exhaustive techniques employing sophisticated upper bounds to find the best patterns according to established quality criteria. As was to be expected, this

progression happened much faster than for the original pattern mining settings, which also means that the field has completely caught up to the state of the art. Any future developments in pattern mining could be ported to the communities-plus-descriptions-setting without problems, giving rise to new powerful methods.

The third class draws liberally from anything that allows to assess attribute importance, whether via clustering, learning, probabilistic modeling or others. This gives that class the highest flexibility when it comes to integrating recent advances, and makes it the largest of the three algorithmic classes we considered. Yet matching attributes or their combinations to communities will in most cases only be approximate, as opposed to the more concrete descriptions of the other two classes.

One of the benefits of surveying the state of the art lies in seeing what potential research directions remain underexplored.

1. Existing approaches have arguably picked the low-hanging fruit in focusing on itemset or attribute-value annotations. Yet vertices could as well be described by sequences, graphs (molecules), or logical formulas. Given that clustering and pattern mining techniques for such complex data representations exist, there is no reason that one could not port existing work to the — admittedly challenging — problem setting of finding communities that have more complex descriptions than attribute-value combinations.
2. The real-life data captured in graphs is often not static but changes over time, whether in collaboration networks or networks modeling human mobility, e. g., Giannotti et al. [2016]. Dynamic attributed graphs have been studied [Desmier et al., 2014, Boulicaut et al., 2016], as has community detection in dynamic graphs [Mucha et al., 2010, Nguyen et al., 2011, Xie et al., 2013] but the two have yet to be married.
3. While we pointed out a few approaches that address edge-attributed graphs, i.e. characterizations of the *relationships* among entities, the vast majority of existing work focuses on vertex-attributed graphs, i.e. characterizations of the entities *themselves*.
4. Richer network representations like multi-layer and multiplex networks [Mucha et al., 2010] provide a rich set of analysis options concerning the network structure which can be exploited in community detection.
5. Finally, while each of the previously mentioned future directions should be expected to be challenging, we fully expect that at some point they will be combined, if only because the problem setting offers a very rich descriptive model of the world. Finding communities in dynamic multiplex networks that can be described (and therefore understood) by complex descriptions on vertices and edges is the foreseeable endpoint of a development of which we have only sketched the beginnings in this work.

It is not entirely clear, however, whether developments in this direction can be expected anytime soon. Research on graph and network analysis has exceedingly focussed on *embedding* techniques in recent years, even if it is not clear that such techniques represent clear improvements [Mara et al., 2020]. It is therefore entirely possible that we will see the same development as in deep learning-based machine learning: opaque models are learned, and symbolic methods added afterwards to make those models interpretable, instead of deriving interpretable results directly.

References

- Martin Atzmueller. Subgroup Discovery. *WIREs: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.
- Martin Atzmueller. Detecting community patterns capturing exceptional link trails. In Ravi Kumar, James Caverlee, and Hanghang Tong, editors, *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016, San Francisco, CA, USA, August 18-21, 2016*, pages 757–764. IEEE Computer Society, 2016.
- Martin Atzmueller and Folke Mitzlaff. Towards Mining Descriptive Community Patterns. In *Workshop on Mining Patterns and Subgroups*. Leiden, The Netherlands, 2010.
- Martin Atzmueller and Folke Mitzlaff. Efficient Descriptive Community Mining. In *Proc. 24th International FLAIRS Conference*, pages 459 – 464, Palo Alto, CA, USA, 2011. AAAI Press.
- Martin Atzmueller, Stephan Doerfel, and Folke Mitzlaff. Description-Oriented Community Detection using Exhaustive Subgroup Discovery. *Information Sciences*, 329:965–984, 2016.
- Martin Atzmueller, Henry Soldano, Guillaume Santini, and Dominique Bouthinon. MinerLSD: Efficient local pattern mining on attributed graphs. In Hanghang Tong, Zhenhui Jessie Li, Feida Zhu, and Jeffrey Yu, editors, *2018 IEEE International Conference on Data Mining Workshops, ICDM Workshops, Singapore, Singapore, November 17-20, 2018*, pages 219–228. IEEE, 2018.
- Martin Atzmueller, Henry Soldano, Guillaume Santini, and Dominique Bouthinon. MinerLSD: Efficient mining of local patterns on attributed networks. *Applied Network Science*, 4(1):43:1–43:33, 2019.
- Ramnath Balasubramanyan and William W. Cohen. Block-LDA: Jointly modeling entity-annotated text and entity-entity links. In *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, April 28-30, 2011, Mesa, Arizona, USA*, pages 450–461. SIAM / Omnipress, 2011.
- Luca Baldesi, Carter T. Butts, and Athina Markopoulou. Spectral graph forge: Graph generation targeting modularity. In *2018 IEEE Conference on Computer Communications, INFOCOM 2018, Honolulu, HI, USA, April 16-19, 2018*, pages 1727–1735. IEEE, 2018.
- Alejandro Baldominos, Javier Calle, and Dolores Cuadra. Beyond Social Graphs: Mining Patterns Underlying Social Interactions. *Pattern Analysis and Applications*, 20(1):269–285, 2017.
- Roberto Bayardo, Rakesh Agrawal, and Dimitrios Gunopulos. Constraint-Based Rule Mining in Large, Dense Databases. *Data Mining and Knowledge Discovery*, 4:217–240, 2000.
- Brigitte Boden, Stephan Günnemann, Holger Hoffmann, and Thomas Seidl. Mining coherent subgraphs in multi-layer graphs with edge labels. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 1258–1266, 2012.
- Brigitte Boden, Stephan Günnemann, Holger Hoffmann, and Thomas Seidl. Rmics: a robust approach for mining coherent subgraphs in edge-labeled multi-layer graphs. In *Conference on Scientific and Statistical Database Management, SSDBM '13, Baltimore, MD, USA, July 29 - 31, 2013*, pages 23:1–23:12, 2013.

- Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann. NetGAN: Generating Graphs via Random Walks. In Jennifer Dy and Andreas Krause, editors, *Proc. International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 609–618, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.
- Cécile Bothorel, Juan David Cruz, Matteo Magnani, and Barbora Micenková. Clustering attributed graphs: Models, measures and methods. *Netw. Sci.*, 3(3): 408–444, 2015.
- Jean-François Boulicaut, Marc Plantevit, and Céline Robardet. Local pattern detection in attributed graphs. In *Solving Large Scale Learning Tasks. Challenges and Algorithms*, pages 168–183. Springer, 2016.
- Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 84–95. Springer, 2000.
- Elise Desmier, Marc Plantevit, Céline Robardet, and Jean-François Boulicaut. Granularity of co-evolution patterns in dynamic attributed graphs. In Hendrik Blockeel, Matthijs van Leeuwen, and Veronica Vinciotti, editors, *Advances in Intelligent Data Analysis XIII - 13th International Symposium, IDA 2014, Leuven, Belgium, October 30 - November 1, 2014. Proceedings*, volume 8819 of *Lecture Notes in Computer Science*, pages 84–95. Springer, 2014.
- Reinhard Diestel. *Graph Theory*. Springer, Berlin, 2006.
- Hongxia Du, Heli Sun, Jianbin Huang, Zhongbin Sun, Liang He, and Hong Cheng. Mining cohesive clusters with interpretations in labeled graphs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 774–785. Springer, 2017.
- Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pages 226–231. AAAI Press, 1996.
- Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.
- L.C. Freeman. Segregation In Social Networks. *Sociological Methods & Research*, 6(4):411, 1978.
- Esther Galbrun, Aristides Gionis, and Nikolaj Tatti. Overlapping community detection in labeled graphs. *Data Min. Knowl. Discov.*, 28(5-6):1586–1610, 2014.
- Fosca Giannotti, Lorenzo Gabrielli, Dino Pedreschi, and Salvatore Rinzivillo. Understanding human mobility with big data. In Stefan Michaelis, Nico Piatkowski, and Marco Stolpe, editors, *Solving Large Scale Learning Tasks*, volume 9580 of *LNCS*, pages 208–220. Springer, 2016.
- M. Girvan and M. E. J. Newman. Community Structure in Social and Biological Networks. *PNAS*, 99(12):7821–7826, June 2002.
- Stephan Günnemann, Ines Färber, Brigitte Boden, and Thomas Seidl. Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. In Geoffrey I. Webb, Bing Liu, Chengqi Zhang, Dimitrios Gunopulos, and Xindong Wu, editors, *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, pages 845–850. IEEE Computer Society, 2010.

- Stephan Günnemann, Brigitte Boden, and Thomas Seidl. DB-CSC: A density-based approach for subspace clustering in graphs with feature vectors. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I*, pages 565–580, 2011.
- Stephan Günnemann, Brigitte Boden, and Thomas Seidl. Finding density-based subspace clusters in graphs with feature vectors. *Data Min. Knowl. Discov.*, 25(2):243–269, 2012.
- Stephan Günnemann, Brigitte Boden, Ines Färber, and Thomas Seidl. Efficient mining of combined subspace and subgraph clusters in graphs with feature vectors. In *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part I*, pages 261–275, 2013a.
- Stephan Günnemann, Ines Färber, Sebastian Raubach, and Thomas Seidl. Spectral subspace clustering for graphs with feature vectors. In Hui Xiong, George Karypis, Bhavani M. Thuraisingham, Diane J. Cook, and Xindong Wu, editors, *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, pages 231–240. IEEE Computer Society, 2013b.
- Stephan Günnemann, Ines Färber, Brigitte Boden, and Thomas Seidl. GAMer: A Synthesis of Subspace Clustering and Dense Subgraph Mining. In *Knowledge and Information Systems (KAIS)*. Springer, 2013c.
- David J Hand. Pattern Detection and Discovery. In David J Hand, Niall M Adams, and Richard J Bolton, editors, *Pattern Detection and Discovery*, volume 2447 of *LNCS*, pages 1–12. Springer, 2002.
- Janis Kalofolias, Mario Boley, and Jilles Vreeken. Discovering robustly connected subgraphs with simple descriptions. In Jianyong Wang, Kyuseok Shim, and Xindong Wu, editors, *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, pages 1150–1155. IEEE, 2019.
- Mehdi Kaytoue, Marc Plantevit, Albrecht Zimmermann, Ahmed Anes Bendimerad, and Céline Robardet. Exceptional contextual subgraph mining. *Mach. Learn.*, 106(8):1171–1211, 2017.
- Myunghwan Kim and Jure Leskovec. Modeling Social Networks with Node Attributes Using The Multiplicative Attribute Graph Model. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI’11*, pages 400–409, Arlington, Virginia, United States, 2011. AUAI Press.
- Willi Klösgen. *Handbook of Data Mining and Knowledge Discovery*, chapter 5.2: Subgroup Patterns. Oxford University Press, New York, 2002.
- Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark Graphs for Testing Community Detection Algorithms. *Physical review E*, 78(4):046110, 2008.
- Christine Largeron, Pierre-Nicolas Mougél, Reihaneh Rabbany, and Osmar R. Zaïane. Generating attributed networks with communities. *PloS one*, 10(4):e0122777, 2015.
- Christine Largeron, Pierre-Nicolas Mougél, Oualid Benyahia, and OR Zaïane. DANCer: Dynamic Attributed Networks With Community Structure Generation. *Knowledge and Information Systems*, 53(1):109–151, 2017.
- Jay Lee, Manzil Zaheer, Stephan Günnemann, and Alexander J. Smola. Preferential attachment in graphs with affinities. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015*,

- San Diego, California, USA, May 9-12, 2015*, 2015.
- Daifeng Li, Bing He, Ying Ding, Jie Tang, Cassidy Sugimoto, Zheng Qin, Erjia Yan, Juanzi Li, and Tianxi Dong. Community-based Topic Modeling for Social Tagging. In *Proc. 19th ACM International Conference on Information and Knowledge Management, CIKM*, pages 1565–1568, New York, NY, USA, 2010. ACM.
- Alexandru Cristian Mara, Jefrey Lijffijt, and Tijn De Bie. Benchmarking network embedding models for link prediction: Are we making progress? In Geoffrey I. Webb, Zhongfei Zhang, Vincent S. Tseng, Graham Williams, Michalis Vlachos, and Longbing Cao, editors, *7th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2020, Sydney, Australia, October 6-9, 2020*, pages 138–147. IEEE, 2020.
- Bella Martínez-Seis. RELNA: Ranking Attributes in Social Networks to Detect Overlapping Communities Efficiently. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*, pages 1431–1435. IEEE, 2017.
- Andrew McCallum, Xuerui Wang, and Natasha Mohanty. Joint group and topic discovery from relations and text. In Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, Anna Goldenberg, Eric P. Xing, and Alice X. Zheng, editors, *Statistical Network Analysis: Models, Issues, and New Directions - ICML 2006 Workshop on Statistical Network Analysis, Pittsburgh, PA, USA, June 29, 2006, Revised Selected Papers*, volume 4503 of *Lecture Notes in Computer Science*, pages 28–44. Springer, 2006.
- Katharina Morik. Detecting Interesting Instances. In David J Hand, Niall M Adams, and Richard J Bolton, editors, *Pattern Detection and Discovery*, volume 2447 of *LNCS*, pages 13–23. Springer Berlin/Heidelberg, 2002.
- Katharina Morik, Jean-François Boulicaut, and Arno Siebes, editors. *Local Pattern Detection, International Seminar, Dagstuhl Castle, Germany, April 12-16, 2004, Revised Selected Papers*, volume 3539 of *LNCS*, 2005. Springer.
- Flavia Moser, Recep Colak, Arash Rafiey, and Martin Ester. Mining cohesive patterns from graphs with feature vectors. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2009, April 30 - May 2, 2009, Sparks, Nevada, USA*, pages 593–604. SIAM, 2009.
- Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.
- Mark E. Newman. Detecting community structure in networks. *The European physical journal B*, 38(2):321–330, 2004.
- Mark E. Newman and Aaron Clauset. Structure and Inference in Annotated Networks. *Nature Communications*, 7, 2016.
- Mark E. Newman and Michelle Girvan. Finding and Evaluating Community Structure in Networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 69(2):1–15, 2004.
- Nam P. Nguyen, Thang N. Dinh, Sindhura Tokala, and My T. Thai. Overlapping communities in dynamic networks: their detection and mobile applications. In Parmesh Ramanathan, Thyaga Nandagopal, and Brian Neil Levine, editors, *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking, MOBICOM 2011, Las Vegas, Nevada, USA, September 19-23, 2011*, pages 85–96. ACM, 2011.
- Petra Kralj Novak, Nada Lavrač, and Geoffrey I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup

- mining. *Journal of Machine Learning Research*, 10:377–403, 2009.
- Simon Pool, Francesco Bonchi, and Matthijs van Leeuwen. Description-driven community detection. *ACM Trans. Intell. Syst. Technol.*, 5(2):28:1–28:28, 2014.
- Meng Qin, Di Jin, Kai Lei, Bogdan Gabrys, and Katarzyna Musial-Gabrys. Adaptive community detection incorporating topology and content in social networks. *Knowl. Based Syst.*, 161:342–356, 2018.
- Matt Revelle, Carlotta Domeniconi, Mack Sweeney, and Aditya Johri. Finding community topics and membership in graphs. In Annalisa Appice, Pedro Pereira Rodrigues, Vítor Santos Costa, João Gama, Alípio Jorge, and Carlos Soares, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II*, volume 9285 of *Lecture Notes in Computer Science*, pages 625–640. Springer, 2015.
- Pablo Robles-Granda, Sebastián Moreno, and Jennifer Neville. Sampling of attributed networks from hierarchical generative models. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1155–1164, 2016.
- Patricia Iglesias Sánchez, Emmanuel Müller, Fabian Laforet, Fabian Keller, and Klemens Böhm. Statistical selection of congruent subspaces for mining attributed graphs. In Hui Xiong, George Karypis, Bhavani M. Thuraisingham, Diane J. Cook, and Xindong Wu, editors, *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, pages 647–656. IEEE Computer Society, 2013.
- Stephen B. Seidman. Network Structure and Minimum Degree. *Social Networks*, 5:269–287, 1983.
- Francesc Serratosa. A methodology to generate attributed graphs with a bounded graph edit distance for graph-matching testing. *Int. J. Pattern Recognit. Artif. Intell.*, 32(11):1850038:1–1850038:19, 2018.
- Arlei Silva, Wagner Meira Jr., and Mohammed J. Zaki. Mining attribute-structure correlated patterns in large attributed graphs. *Proc. VLDB Endow.*, 5(5):466–477, 2012.
- Laura M. Smith, Linhong Zhu, Kristina Lerman, and Allon G. Percus. Partitioning networks with node attributes by compressing information flow. *ACM Trans. Knowl. Discov. Data*, 11(2):15:1–15:26, 2016.
- Henry Soldano and Guillaume Santini. Graph abstraction for closed pattern mining in attributed networks. In Torsten Schaub, Gerhard Friedrich, and Barry O’Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 849–854. IOS Press, 2014.
- Henry Soldano, Guillaume Santini, and Dominique Bouthinon. Local rules associated to k-communities in an attributed graph. In Jian Pei, Fabrizio Silvestri, and Jie Tang, editors, *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015, Paris, France, August 25 - 28, 2015*, pages 1340–1347. ACM, 2015.
- Henry Soldano, Guillaume Santini, Dominique Bouthinon, and Emmanuel Lazega. Hub-authority cores and attributed directed network mining. In *29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2017, Boston, MA, USA, November 6-8, 2017*, pages 1120–1127. IEEE Computer Society, 2017.

- Henry Soldano, Guillaume Santini, Dominique Bouthinon, Sophie Bary, and Emmanuel Lazega. Bi-pattern mining of attributed networks. *Applied Network Science*, 4(1):37:1–37:26, 2019.
- Karsten Steinhaeuser and Nitesh V; Chavla. Community detection in a large real-world social network. In *Social computing, behavioral modeling, and prediction*, pages 168–175. Springer, 2008.
- Heli Sun, Hongxia Du, Jianbin Huang, Zhongbin Sun, Liang He, Xiaolin Jia, and Zhongmeng Zhao. Detecting semantic-based communities in node-attributed graphs. *Computational Intelligence*, 34(4):1199–1222, 2018.
- Jilles Vreeken and Nikolaj Tatti. Interesting Patterns. In *Frequent pattern mining*, pages 105–134. Springer, 2014.
- Xiao Wang, Di Jin, Xiaochun Cao, Liang Yang, and Weixiong Zhang. Semantic community identification in large attribute networks. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 265–271. AAAI Press, 2016.
- Jierui Xie, Mingming Chen, and Boleslaw K. Szymanski. LabelRankT: incremental community detection in dynamic networks via label propagation. In *Proceedings of the Workshop on Dynamic Networks Management and Mining, DyNetMM 2013, New York, New York, USA, June 22-27, 2013*, pages 25–32. ACM, 2013.
- Zhiqiang Xu, Yiping Ke, Yi Wang, Hong Cheng, and James Cheng. A model-based approach to attributed graph clustering. In K. Selçuk Candan, Yi Chen, Richard T. Snodgrass, Luis Gravano, and Ariel Fuxman, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 505–516. ACM, 2012.
- Zhiqiang Xu, Yiping Ke, Yi Wang, Hong Cheng, and James Cheng. GBAGC: A general bayesian framework for attributed graph clustering. *ACM Trans. Knowl. Discov. Data*, 9(1):5:1–5:43, 2014.
- Jaewon Yang and Jure Leskovec. Defining and Evaluating Network Communities Based on Ground-truth. In *Proc. ACM SIGKDD Workshop on Mining Data Semantics, MDS '12*, pages 3:1–3:8, New York, NY, USA, 2012. ACM.
- Jaewon Yang, Julian J. McAuley, and Jure Leskovec. Community detection in networks with node attributes. In Hui Xiong, George Karypis, Bhavani M. Thuraisingham, Diane J. Cook, and Xindong Wu, editors, *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, pages 1151–1156. IEEE Computer Society, 2013.
- Elena Zheleva, Hossam Sharara, and Lise Getoor. Co-evolution of social and affiliation networks. In John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki, editors, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 1007–1016. ACM, 2009.