Statistical Selection of Congruent Subspaces for Mining Attributed Graphs

Patricia Iglesias Sánchez [•] Emmanuel Müller [•○] Fabian Laforet [•] Fabian Keller [•] Klemens Böhm [•]

[•] *Karlsruhe Institute of Technology (KIT), Germany* [○] *University of Antwerp, Belgium*

{*patricia.iglesias, emmanuel.mueller, fabian.laforet, fabian.keller, klemens.boehm*}*@kit.edu*

***Abstract*—Current mining algorithms for attributed graphs exploit dependencies between attribute information and edge structure, referred to as homophily. However, techniques fail if this assumption does not hold for the full attribute space. In multivariate spaces, some attributes have high dependency with the graph structure while others do not show any dependency. Hence, it is important to select congruent subspaces (i.e., subsets of the node attributes) showing dependencies with the graph structure.**

In this work, we propose a method for the statistical selection of such congruent subspaces. More specifically, we define a measure which assesses the degree of congruence between a set of attributes and the entire graph. We use it as the core of a statistical test, which congruent subspaces must pass. To illustrate its applicability to common graph mining tasks and in order to evaluate our selection scheme, we apply it to community outlier detection. Our selection of congruent subspaces enhances outlier detection by measuring outlierness scores in selected subspaces only. Experiments on attributed graphs show that our approach outperforms traditional full space approaches and gives way to better outlier detection.

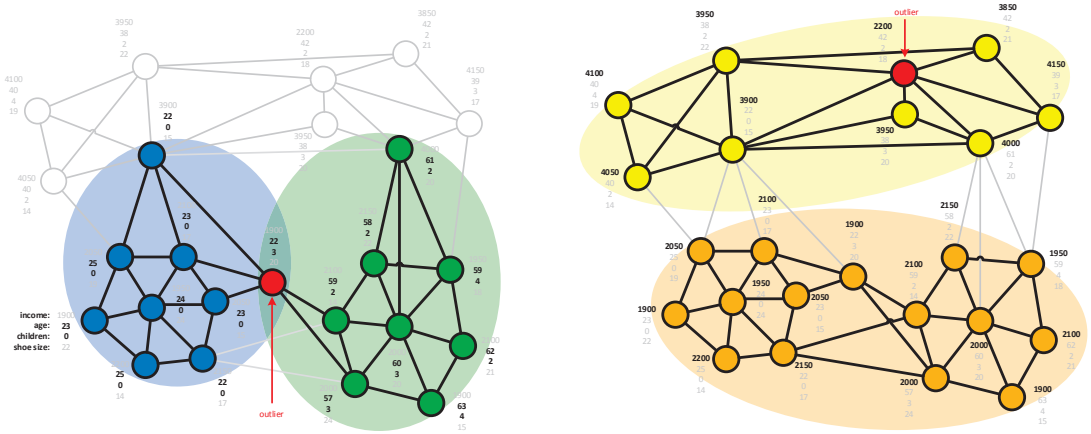
***Keywords*—attributed graphs, homophily, subspace selection**

I. Introduction

Attributed graphs are widely used for the representation of social networks, gene and protein interactions, communication networks, or product co-purchase in web stores. Each object is represented by its relationships to other objects (edge structure) and its individual properties (node attributes). For instance, social networks store *friendship relations* as edges and *age*, *income*, and other properties as attributes. Relationships and properties seem to be dependent on each other. Several publications [23], [27], [8], [10] have shown that exploiting existing dependencies is beneficial, e.g., for cluster and outlier detection. However, techniques proposed in these articles highly rely on this dependency assumption. In particular, *community outlier mining* [10] is able to detect an outlier node if connected nodes have similar values in all attributes. Such assumptions are known as homophily [16] and are widely used. However, looking at multivariate spaces, one can observe that not all given attributes have high dependencies with the graph structure. For example, social properties such as *income* or *age* have strong dependencies with the graph structure of social networks [16]. In contrast, properties such as *gender* are rather independent from it. Consequently, recent graph mining algorithms degenerate for multivariate attribute spaces that lack dependency with the graph structure in some of the attributes. This calls for a general pre-processing step that selects subspaces, i.e., subsets of the attributes, showing dependencies with the graph.

Let us illustrate this with a toy example in Figure 1. It features a social network with *friendship relation* as edges and node attributes (*income*, *age*, *number of children*, and *shoe size*). Given a dependency between a set of attributes (e.g., *age* and *number of children*) and the edge structure (cf. Figure 1(a)), we observe communities of young persons without children, old people with several children, and a deviating outlier. Considering another subspace (cf. Figure 1(b)), we observe a different community/outlier structure w.r.t. income. In general, we observe a dependency between high edge counts within a community and similar attribute values on different selections of attributes. However, the homophily assumption is not fulfilled for the full attribute space. Thus, the detection of either communities or outlier nodes is hindered considering all four attributes.

We call subsets of attributes showing a dependency with the graph structure *congruent subspaces*. A core challenge in selecting these subspaces lies in the modeling of dependence between graph structure and attribute values. Further, one has to ensure that congruent subspaces are selected only if there is sufficient evidence on this dependence. We propose the method *ConSub* for the statistical selection of *congruent subspaces*. More specifically, we address all those problems as follows: First, we propose a novel measure for the degree of congruence between a set of node attributes and a graph by means of edge counts and attribute values. We compare edge counts in subgraphs constrained by attribute value ranges. These constrained subgraphs are randomly chosen in a Monte Carlo processing and are used as a source of indication for dependencies. Our congruence measure exploits these dependencies between random subgraphs and their attribute subspaces. We select attribute subsets featuring those dependencies in multivariate attribute spaces. This selection can serve as general pre-processing step for algorithms that rely on the homophily assumption on attributed graphs. That is, our method ensures within a community similar values in all selected attributes of a congruent subspace. The distances between nodes (considering only attribute values from the selected subspace) closely resemble the graph structure. This allows us to establish a neighborhood of a node by simply comparing distances between connected nodes, which is useful for different mining tasks on attributed graphs [23], [27], [8], [10]. Regarding outlier mining, one can identify community outliers merely based on these subspace neighborhoods having



(a) subspace {age, children}

(b) subspace {income}

Fig. 1. Toy example for congruent subspaces in a social network

identified congruent subspaces. In our paper, we focus on such community outliers as an exemplary graph mining task and propose an agglomerative clustering approach to search node neighborhoods in which we compute the outlierness of each node. This is a significant improvement over techniques that do not consider subspaces and fall prey to the lack of homophily in the full attribute space. To the best of our knowledge *ConSub* is the first pre-processing technique that can ensure homophily in a subset of attributes w.r.t. the graph structure.

Overall, our contributions with *ConSub* are as follows:

- We introduce the problem of congruent subspaces for graph mining techniques relying on the homophily assumption.
- We propose the first approach for the statistical selection of congruent subspaces.
- Our selection scheme allows a clear enhancement of community outlier mining compared to traditional full space methods.

II. Related Work

There is a variety of outlier mining models in multivariate data or anomaly detection for graph data [1]. However, there is only limited work on graph mining with multivariate node attributes and only few papers on outlier mining on attributed graphs. We distinguish our method from three mining paradigms on attributed graphs: (1) *full space approaches* assuming a dependency between all attributes and the entire graph, (2) *specialized subspace techniques* using specific subspace selection mechanisms internally in their algorithms, and (3) *general feature selection methods* that can be used as pre-processing step to any graph mining algorithm.

Full space approaches: Regarding graph clustering, some techniques exploit the correlation between all attributes and the graph structure in order to improve the clustering result [23], [27]. As an outlier mining algorithm, [8] searches for irregular subgraphs with numeric node attributes. In contrast, community outlier detection [10] focuses on outlier nodes that deviate from a community of similar nodes. The major drawback of all of these approaches is their assumption that all node attributes are dependent w.r.t. the entire graph. However, the quality of outlier detection is highly affected by irrelevant attributes as we will demonstrate in our experiments. Our method excludes such attributes, and, in contrast to traditional full space processing, allows for robust analysis of multivariate attribute spaces.

Specialized subspace techniques: Recent methods have observed the lack of dependency in the full attribute space and have proposed individual subspace selection schemes for specific subgraphs. A first technique for mining frequent subgraphs selects subsets of attributes [24]. A graph partitioning algorithm only incorporates binary attribute vectors in its search for strongly connected subgraphs [4]. Other approaches search for overlapping clusters in subsets of categorical attributes [15], clusters in subspaces of numeric node attributes [11], or outliers that can be derived from subspace clusters in attributed graphs [18], [19]. The selection schemes of all these techniques are based on individual subgraphs from the graph structure [4], [15], [11]. In order to retrieve a congruent subspace from their results, each node of the graph has to belong to a cluster result in the same subspace. However, these techniques do not aim to ensure this (e.g., specific cluster definitions such cliques enforce to exclude a large number of nodes from the graph). Thus, they can be considered specific solutions to the problem of subspace selection, but they lack generality and are not designed as pre-processing step for other graph mining models.

General feature selection methods: For individually analyzing the dependency of an attribute, the *assortative mixing coefficient* has been proposed in order to measure the correlation between a single attribute and the graph structure [22]. Nevertheless, this coefficient is not able to measure if a correlated subset of attributes also depends on the graph structure. In contrast to this assessment of individual attributes, feature selection is a general pre-processing step for supervised methods, and has been extended recently to unsupervised feature selection on attributed graphs [25]. However, the main focus of these techniques is the improvement of traditional feature selection on vector data by incorporating additional information given by object relationships in a graph structure. Hence, they do not intend to select the attributes that show high dependencies with the graph structure. They only utilize graphs as additional information source. In contrast to feature selection methods, we focus on the mutual dependency of the attribute values and the graph structure. Furthermore, we select multiple attribute sets that show dependence with the graph structure. In our

experiments, we will compare our subspace selection scheme as a pre-processing step to community outlier mining with main competitors from unsupervised feature selection [25] and full space outlier detection [10].

III. Problem Overview

We model an attributed graph by its graph structure $G = (V, E)$ and its attribute information A as follows:
(1) Each object o is a graph vertex $o \in V$ and connected by edges $(o, p) \in E$ to other nodes $p \in V \setminus \{o\}$ in the graph structure. We assume edges to be undirected and unweighted.
(2) Each object o is described by a vector $\vec{o} = (x_1, ..., x_d)$ where the attributes are named $A = \{A_1, \ldots, A_d\}$.

A. Selection Scheme

Existing algorithms exploit the dependencies between both graph G and attributes A for knowledge discovery. In particular, they exploit the assumption of homophily [16] that connected nodes tend to have similar characteristics. This effect is also known as *assortative mixing* [22]. However, this assumption may not be fulfilled for the full space of attributes A. Some of the attributes in A do not depend on the underlying graph structure, or they can even show an opposite trend known as *disassortative mixing* [22].
For example, *community outlier mining* needs to capture a group of similar objects w.r.t. the graph structure and the attribute values [10]. Thus, all the attribute values, used for outlier detection, and the graph structure have to be correlated. This occurs only if the network is assortative w.r.t. all given attributes. In case a network shows *disassortive mixing*, the search of similar objects w.r.t. both attribute values and the graph structure is hindered. However, simply measuring assortativity of a single attribute with the graph structure as proposed in [22] is not enough.We expect outliers or clusters to hide in combinations of attribute values, as has already been shown for multivariate vector data [1], [6]. Thus, one has to consider the dependency of multiple attributes among each other and with the graph structure. We call such attribute sets *congruent subspaces*.
We aim at an automatic selection of subspaces $S \subseteq A$ that show significant dependence with the graph structure. In this case, connected nodes show similar values in these subsets of the attributes. Informally, we define a *congruent subspace* as $S \subseteq A$ where the attribute values are consistent with the graph structure.

Definition 1: Congruent Subspaces
Given graph $G = (V, E)$ and subspace $S \subseteq A$,

$$S \text{ is congruent with } G :\Leftrightarrow$$

*$\forall\ V' \subseteq V$ with high **mutual similarity** between the attribute values in subspace S:*
*Subgraph $G' = (V', E')$ with $E' = \{(o, p) \in E \mid o, p \in V'\}$ has **significantly more edges** than **expected** if edges were distributed at random. The set of all selected subspaces is then:*

$$CS = \{S \subseteq A \mid S \text{ is congruent with } G\}$$

A subgraph showing more edges than expected in subspace S is the result of a positive correlation between attribute values and the graph structure: The graph structure is congruent with subspace S. Having less edges than expected also shows a dependency with the graph structure. However, this negative correlation indicates that the graph structure is opposite to the attribute values in subspace S. Thus, the attribute values in S are not congruent with the graph structure.
Given Definition 1, three main questions remain: (1) how to define mutual similarity of objects within a subgraph (V', E') in subspace S, (2) how to perform the statistical significance test on the observed edge count $|E'|$, and (3) how to assess the number of expected edges based on a predefined null model. We address all of these questions for the selection of one congruent subspace in Section IV and describe the algorithm for the selection of CS in Section V.

B. Community Outlier Mining

Community outliers appear in a combined consideration of the graph structure and the attribute values. Exceptional nodes are highly deviating in some of their attribute values from the community they belong to [10]. In general, communities can be found by considering the graph structure and the distribution of the attribute values in the database as in the original publication [10]. Communities can be detected if attribute values show a certain degree of congruence w.r.t. the graph structure. In this case, a community outlier can be detected as an irregularity deviating from such a group of similar nodes.
However, outlier mining fails if the full attribute space A does not follow the assumption that all the attributes are congruent with the graph structure. In case of CODA [10], we observe a huge amount of false positives and false negatives. In particular, multivariate data poses a major problem for CODA. As mentioned in the original publication [10], CODA can only deal with dimensions that are correlated with the graph (i.e., in our notion: A is congruent with the graph). However, not all given dimensions are congruent on the graph structure in real world networks [16], [22]. Furthermore, different subsets of attributes correspond to different community/outlier structures (cf. Figure 1). Thus, outliers hidden in different congruent subspaces are missed if one only considers the full dimensional space or a single projection of the attribute space. Therefore, we introduce the notion of *subspace community outliers*.

Definition 2: Subspace Community Outlier
Given a congruent subspace S and a neighborhood $N \subseteq V$, we define an outlier as:

$$\textit{a node } o \in N \textit{ that shows a } \textbf{\textit{high deviation}} \textit{ in } S$$

i.e., it is highly deviating from the local neighborhood in the attribute values of S.

In the following, we assume $score(o, S)$ to be a function which quantifies the outlier degree of an object in a subspace S. We measure deviation by an aggregate of scores in all congruent subspaces:

Definition 3: Subspace Outlier Score

$$score(o) = \frac{\sum_{S \in CS} score(o, S)}{|CS|}$$

We deem the selection of congruent subspaces CS to be the key feature of Definition 3. It is the major difference to traditional outlier scores using the entire set of attributes

$score(o, A)$. Please note that other aggregation functions or even ensemble techniques might be of interest as well [20], [2]. However, this is research orthogonal to our current work and will not be addressed in this paper. Here, we focus on the selection of CS as described in the following sections and give more details on the instantiation of $score(o, S)$ in Section VI.

IV. Subspace Selection

In order to assess the congruence of a subspace $S \subseteq A$ with the graph structure, we consider several random subgraphs constrained by attribute ranges in subspace S. In more detail, we select random intervals of attributes $A_j \in S$ in a Monte Carlo processing. For each interval, we consider the subgraph formed by the nodes that have attribute values within these intervals. Thus, we ensure similar attribute values within the subgraphs as it is a requirement for congruent subspaces (cf. Definition 1). We determine the number of edges in these subgraphs and compare them to the number of edges expected. Observing more edges than expected highlights the dependence between the selected attribute region and the induced subgraph. In this case, we deem the edge structure and the node attributes congruent on the subspace. In Section IV-A, we introduce the *ConSub* measure for the assessment of congruence. We describe the estimation of the number of expected edges in Section IV-B and propose a statistical test for the comparison of observed and expected edge counts in Section IV-C.

A. Congruence Assessment

In *ConSub*, we consider intervals of the attribute values for the retrieval of subgraphs where nodes have similar values. These attribute regions $[low_j, high_j]\ \forall A_j \in S$ restrict the graph structure to subgraphs (cf. Definition 4). For an overall assessment of the dependencies between subspaces and the graph structure we consider several of these subgraphs that are constrained by different attribute regions.

Given a subspace S, we define a *constraint subgraph*:

Definition 4: Constraint Subgraph $G_{C,S}$
Given a set of constraints C consisting of all the pairs $(I_j, A_j) \in C$ formed by each dimension $A_j \in S$ and an interval $I_j = [low_j, high_j]$, we define a constrained subgraph $G_{C,S} = (V_{C,S}, E_{C,S})$ as

$$V_{C,S} = \{o \in V \mid \vec{o} = (x_1, \ldots, x_d) \wedge \forall A_j \in S : x_j \in I_j\}$$

and

$$E_{C,S} = \{(o, p) \in E \mid o \in V_{C,S} \wedge p \in V_{C,S}\}$$

Continuing our running example from Figure 1, we consider the constraint subgraph $G_{\{([3000,5000],income)\},\{income\}}$. Since this subgraph is congruent with the given constraints, we observe an unexpectedly high number of edges $|E_{\{([3000,5000],income)\},\{income\}}|$.

For our assessment of congruence, we compare this observed edge count with the expected number of edges. We compute the expected number of edges based on a null model assuming no congruency between graph structure and attribute values. In particular, we use a model that preserves the degree distribution, as we will describe in Section IV-B.

Finally, the deviation of observed and expected edge counts is measured based on the constraint subgraph. However, this assessment of a single constraint subgraph does not provide sufficient evidence for the congruence of the entire graph on subspace S. In order to get a sufficient number of samples to determine the congruence of a subspace, we propose a Monte Carlo processing. In iteration m, we select a constraint subgraph $G^m_{C,S}$ by randomly generating a set of constraints C in subspace S. Then, the respective samples are used to compute the observed and expected edge count, and are passed to a *deviation* function.

Definition 5: Congruence Measure
Given M Monte Carlo runs where $G^m_{C,S}$ is the constraint subgraph in iteration m:

$$congruence(S) \equiv \frac{1}{M} \sum_{m=1}^{M} deviation(|E^m_{C,S}|, E_{exp}(G^m_{C,S}))$$

where $|E^m_{C,S}|$ is the observed edge count in $G^m_{C,S}$ and $E_{exp}(G^m_{C,S})$ is the expected edge count.

The observed edge count is the number of edges of the subgraph $G^m_{C,S}$, and the expected number of edges is estimated under the assumption that there is no congruence between the constraint subgraph $G^m_{C,S}$ and the attribute values of S. The observed and the expected edge count are passed to a *deviation* function which is explained in Section IV-C. In our case we perform a statistical test in order to measure the significance of the observed deviation. The overall congruence of a subspace S is then computed as the average of the deviation of all constraint graphs analyzed.

B. Expected Edge Count Estimation

Definition 1 requires that a constraint subgraph has significantly more edges than expected if edges were distributed at random. Hence, we face the problem of estimating the expected edge count. Null models are commonly used as the basis for such expected edge counts. They are structural instantiations of a graph where edges are wired at random [9], [21]. In our approach, we want to use this estimation for testing if there are significantly more edges than expected. However, it is essential that this estimation is as concrete as possible. We only have to reject the null model in the case of congruent subspaces. Thus, we propose a null model with the following characteristics:
(1) By definition, the null model supposes attributes values and edge structure to be independent, i.e., the attribute distribution does not have any impact on the edge connections.
(2) We exploit information of the whole graph structure considering its structural characteristics. Previous work has shown that communities may differ in their degree distribution. Thus, preserving the degree distribution is an important requirement for an accurate estimation of the expected number of edges [21]. Therefore, we employ a null model that preserves the degree distribution of the given graph.
(3) If a lower dimensional subspace $S'_1 = S \backslash \{A_j\}$ contains a large number of observed edges, the expected edge count in the higher dimensional subspace S should be high as well. Thus, we consider lower-dimensional projections of S to compute the expected number of edges in $G_{C,S}$. We adapt the estimation accordingly (i.e., we increase the expectation if we observe

a high number edges in a lower-dimensional projection of S). To achieve this, we estimate the expected edge count in a constraint subgraph $G_{C,S}$ based on a *relaxed subgraph* $G_{C\setminus\{(I_j,A_j)\},S\setminus\{A_j\}}$ where A_j is a randomly selected attribute.

Let us first define the degree function of the edges of such a relaxed subgraph. Our null model preserves this degree distribution.

Definition 6: Preserved Degree Function
Given a constraint subgraph $G_{C,S}$ and a randomly selected attribute A_j, the preserved degree function of a node $o \in V'$ is:

$$deg(G_{C,S}, A_j, o) = |\{(o,p) \in E \mid p \in V'\}|$$

where $V' = V_{C\setminus\{(I_j,A_j)\},S\setminus\{A_j\}}$ is the set of nodes belonging to the relaxed subgraph.

Given the set of nodes $V_{C,S}$ of the constraint subgraph, we estimate the edge count by the summation of the expected number of edges that exist between nodes in $V_{C,S}$. In order to calculate the expected edge count of a single node o we apply the hypergeometric distribution. Each vertex draws $deg(G_{C,S}, A_j, o)$ edges to other nodes without a constraint on A_j in the constraint subgraph $G_{C,S}$. Thus, each edge creates a connection to one object in $V' \setminus \{o\}$. The *population size* of the hypergeometric distribution is given by the sum of the degrees: $\sum_{p\in V'\setminus\{o\}} deg(G_{C,S}, A_j, p)$. Since we are interested in the expected edge count in $V_{C,S}$, the sum of the conditional degrees in $V_{C,S} \setminus \{o\}$ describes the *number of success states in the population*. Overall we obtain the following edge estimator by summing up the mean values of each hypergeometric distribution for each node in $G_{C,S}$.

Definition 7: Expected Edge Count
Given a constraint graph $G_{C,S} = (V_{C,S}, E_{C,S})$, the expected edge count w.r.t. attribute A_j is computed as:

$$E_{exp}(G_{C,S}) = \tfrac{1}{2} \sum_{o\in V_{C,S}} deg(G_{C,S}, A_j, o) \cdot \frac{\sum_{p\in V_{C,S}\setminus\{o\}} deg(G_{C,S},A_j,p)}{\sum_{p\in V'\setminus\{o\}} deg(G_{C,S},A_j,p)}$$

It is possible to use other edge count estimators for the instantiation of $ConSub$, but they have to satisfy the assumption that attributes and graph structure are independent. In contrast to existing estimators, such as [21] used for the modularity calculation, we exclude self-loops that are meaningless in the context of analyzing congruence. Furthermore, we have managed to bring down the computing effort for the estimation of the expected edge count from quadratic to linear time: The overall *population size* and the *number of success states* of the hypergeometric distribution have to be calculated only once in advance with linear effort for all vertices $o \in V_{C,S}$. The expected number of edges is estimated by iterating over all nodes of the constraint subgraph.

C. Statistical Test

In order to find congruent subspaces with significantly more edges than expected (cf. Definition 1), we propose to use a statistical test. To this end, we instantiate the $deviation(|E_{C,S}|, E_{exp}(G_{C,S}))$ function in Definition 5 as follows. With homophily being the main goal of our selection, only subspaces with significantly more observed edges than the expected ones should pass our selection criterion. Note that the expected number of edges has to be computed based on a null model guaranteeing the independence between attribute values and edge structure, as explained in Section IV-B. So, we can use a statistical test in order to compare the discrepancies between the number of edges observed and the expected one if both resources are independent. We model the null and the alternative hypothesis for our statistical test as:

$$H_0 : |E_{C,S}| = E_{exp}(G_{C,S})$$
$$H_1 : |E_{C,S}| > E_{exp}(G_{C,S})$$

The null hypothesis represents the case where the number of edges observed is equal to the expected one that assumes that attribute values and graph structure are independent. As our null model ensures this independence in its count estimation, we can conclude from the null hypothesis that the subspace is not congruent with the graph structure. The number of observed edges would be as expected if the attributes values were independent from the graph structure. On the other hand, having a larger number of edges observed than expected shows that the subspace is congruent. We use the alternative hypothesis of a one-tailed test for ensuring the condition of congruent subspace. For $ConSub$, we use the Wilcoxon signed-rank test [26], a parameter free test without any assumption on the data distribution. This is one possible instantiation of the statistical test in our framework, but we are not restricted to it.

In order to have results that are significant, we cannot apply the test on a single constraint subgraph $G_{C,S}$. We need to get several samples. We use the randomly selected attribute $A_j \in S$, which is used to create the relaxed subgraph $G_{C\setminus\{(I_j,A_j)\},S\setminus\{A_j\}}$, in order to ensure to be sensitive in the whole attribute range. The attribute A_j randomly divided into k intervals. Using the number of edges observed E^i_{obs} and the expected one E^i_{exp} in the respective intervals $1 \leq i \leq k$, the test variable of the Wilcoxon signed-rank test is computed as:

$$W = \left| \sum_{i=1}^{k} \left[sgn\left(E^i_{obs} - E^i_{exp}\right) \cdot rank\left(|E^i_{obs} - E^i_{exp}|\right) \right] \right|$$

where $sgn(\circ)$ is the signum function, and $rank(\circ)$ denotes the rank using an ascending order of all $|E_{obs} - E_{exp}|$. Strong discrepancy between the observed and the expected edge counts yields large values of W. Thus, a subspace is congruent with the graph, given a significance level α, if the null hypothesis is rejected. We compute the *p-value* that can be obtained from the parameters k and α according to [26]. We use this as the instantiation of $deviation(|E_{C,S}|, E_{exp}(G_{C,S}))$ in each Monte Carlo iteration (cf. Definition 5). Thus, the congruence measure is the average *p-value* for all the Monte Carlo iterations.

V. Computation

The selection of congruent subspaces $S \subseteq A$ is computationally expensive due to the exponential number of subspaces. Overall, one must analyze $2^{|A|}$ subspaces to have the optimal selection of congruent subspaces. Therefore, we propose to address this issue with a standard procedure for subspace search on vector data [7], [20], [12]. This is a heuristic based on the well-known Apriori processing paradigm. Given d-dimensional congruent subspaces $\{S_1, S_2, \ldots\}$, we derive the $(d+1)$-dimensional candidate subspaces with a bottom up

procedure similarly to the Apriori algorithm [3]. However, we only consider those subspaces that are congruent with a significance level α for the generation of higher dimensional subspaces.

Due to the Monte Carlo approach and the statistical measure of congruence, monotonicity does not hold. Hence, our search does not guarantee to find all congruent subspaces. Nevertheless, all the selected subspaces are congruent with a significance level α. So outlier mining approaches relying on the homophily assumption achieve substantial improvements with this selection. In Section VII, experiments will not only demonstrate this, but also the runtime efficiency w.r.t. the dimensionality of our heuristic. In the following, we describe Algorithm 1 in more detail.

Algorithm: For each subspace in the candidate set given as parameter of the algorithm, we perform M Monte Carlo iterations. In each Monte Carlo iteration, a relaxed subgraph is created according to $(|S| - 1)$ random constraints. The remaining attribute A_j is split in k intervals, and this leads to k constraint subgraphs to consider. These constraints are randomly generated based on the adaptive selection of intervals [12]. The deviations between the observed and the expected edge count in these subgraphs are assessed using our statistical test and are aggregated to the congruence measure. After the execution of all Monte Carlo iterations, the congruence value is tested against the given significance level. In case of significance, the candidate is added to the set of congruent subspaces. After all candidates have been analyzed, the $|S|$-dimensional congruent subspaces are used in order to create the new $(|S| + 1)$-dimensional candidates. We initialize the candidate set with each attribute as a one-dimensional candidate subspace.

Complexity: First, we discuss the complexity of analyzing one subspace. Then we explain the worst case scenario w.r.t. the number of subspaces. Our proposed algorithm for the selection of one subspace has a linear cost with the number of nodes and edges. In particular, the complexity of analyzing one subspace is $O(M \cdot (|S| \cdot |V| + |E| + k \cdot \log(k)))$. It needs M Monte Carlo runs for each subspace with dimensionality $|S|$. For each, we have to access the entire graph $(|S| \cdot |V|)$ times in order to select the constrained node sets. This is because the chosen constraints do not guarantee that it contains nodes in all subspaces. In the worst case we also have to iterate $|E|$ times over each edge in order to determine the observed edge count. Performing the Wilcoxon signed-rank test has an effort of $(k \cdot \log(k))$ since the results have to be ordered.

Regarding the number of subspaces, an exponential number of them might be congruent according to the characteristics of the attributed graph. However, in practice, most of the subspaces are excluded very early in the Apriori candidate generation as shown in our experiments with real world data. Thus, our algorithm for subspace selection has low runtimes even for large graphs. The number of selected subspaces depends on the selected significance level α. In Section VII, we study the impact of this parameter on the results and the runtimes.

Algorithm 1 SubspaceSelection

Input: G, M, α, k, Candidate Set $Cand$
Output: Congruent Subspace Set CS

```
1: for all S ∈ Cand do
2:   for i = 1 → M do
3:     choose a random A_j ∈ S
4:     create a random relaxed subgraph G_{C\{(I_j,A_j)},S\{A_j}}
5:     split A_j in a set of k random intervals I_j
6:     for all constraint pairs (A_j, I_j) ∈ C do
7:       determine obs. and exp. (cf. Def. 7) edge
         count for the current constraint subgraph G_{C,S}
8:     end for
9:     calculate test variable W
10:    deviation = p-value corresponding to W
11:    update congruence(S) (cf. Def. 5)
12:  end for
13:  if congruence(S) ≤ α then
14:    CS = CS ∪ {S}
15:  end if
16: end for
17: create new candidates Cand* using CS
18: return CS ∪ SubspaceSelection(G, M, α, k, Cand*)
```

VI. Community Outlier Detection

Given the selection of congruent subspaces in Section IV, we can already enhance the quality of existing community outlier detection models such as CODA [10] or of other graph mining tasks [23], [27], [8] that rely on the homophily assumption. We will show the improvement of CODA in our experiments. However, in addition to this use of *ConSub* as pre-processing, we want to exploit further properties of congruent subspaces for a better community outlier model. Our model yields an improvement over CODA due to (1) its distance-based neighborhood definition that does not assume a specific data distribution, (2) a hierarchical neighborhood computation, and (3) a ranking of outliers overcoming binary outlier detection. However, we point out that our distance-based outlier model (*DistOut*) is just one out of many that are conceivable on top of congruent subspace selection.

Distance-Based Neighborhood: For community outlier detection we need to define the neighborhood of a node. This means that we have to find the set of nodes with the highest similarity between them and this node. In the neighborhood search, we also have to consider both the graph structure and the attribute values. Congruent subspaces solve the main part of this problem as they ensure that nodes with similar attribute values are connected by the graph structure. We can exploit this mutual similarity of connected nodes resulting from Definition 1 by considering the distances between a set of nodes for the neighborhood search. So, we do not assume a fixed distribution of the data (e.g., Gaussian distribution as in CODA [10]).

Overall, our idea for community outlier detection is to find the neighborhood showing the highest similarity and to compute the score of each node w.r.t. its neighborhood. We call the neighborhoods consisting of nodes that are similar w.r.t. the attribute values in a congruent subspace and highly connected with each other, homogeneous neighborhoods. Given a homogeneous neighborhood and the congruent subspace, we can measure the local deviation of each object from its neighborhood in the congruent subspace. A community outlier [10] appears when it has a high local deviation. However, *ConSub* selects a set of subspaces (cf. Definition 1) and each

object may belong to different homogeneous neighborhoods depending on the congruent subspace (cf. Figure 1). Thus, an outlier score (cf. Definition 3) has to compute the deviation of a node w.r.t. its neighborhood in each *congruent subspace*. In the following, we first describe the distance measures used to compute the similarity between nodes. Finally, we explain the criteria for assessing the homogeneity of a neighborhood and present the outlier score.

Distance Measures: To search for the hierarchical neighborhood, we use a bottom-up agglomerative clustering approach: First, each node forms its own cluster and is merged to larger clusters during the process. The agglomerative step merges clusters with the highest similarity w.r.t. both the graph structure and the attribute values. We need thereby to compare the similarity of two neighborhoods N_1 and N_2 for the merging process. Therefore, we first define new distance measures without any assumption of the data distribution for clusters in the joint space of attribute values and edge structure. The similarity measure considers the edges between them, given by the set of inter-cluster edges:

$$E_{inter}(N_1, N_2) = \{(o,p) \in E \mid o \in N_1 \wedge p \in N_2\}$$

We compute the average distance of two connected nodes by these inter-cluster edges.

Definition 8: Cluster Distance in the Joint Space
Given a non-empty set of inter-cluster edges $E_{inter}(N_1, N_2)$ and a congruent subspace S, the edge distance between N_1 and N_2 is as follows:

$$dist(N_1, N_2) = \begin{cases} avg_D(N_1, N_2) & \text{, if } |E_{inter}(N_1, N_2)| \neq 0 \\ 1 & \text{otherwise.} \end{cases}$$

$$avg_D(N_1, N_2) = \frac{\sum_{(o,p) \in E_{inter}(N_1,N_2)} dist_S(\vec{o}, \vec{p})}{|E_{inter}(N_1, N_2)|}$$

where $dist_S(\vec{o}, \vec{p}) \in [0,1]$ is any normalized distance function on the attribute projection of $\vec{o}$ and $\vec{p}$ on the subspace S.

Neighborhoods with the lowest distance are merged in each step. Small distances (e.g. $dist(N_1, N_2) \approx 0$) indicate high similarity between two clusters w.r.t. similar attribute values and closeness in the graph due to the available inter-cluster edges.

Outlier Score: Each node has to be evaluated w.r.t. the neighborhood it belongs to and which shows the most homogenous behavior. To overcome the binary decision proposed in [10], we finally compute the outlier score w.r.t. its homogeneous neighborhood. We define the homogeneity of a neighborhood N as follows:

Definition 9: Neighborhood Homogeneity
Given a congruent subspace S, the homogeneity of a neighborhood N is

$$hom_S(N) = \frac{interdist_S(N) - intradist_S(N)}{\max\{interdist_S(N), intradist_S(N)\}}$$

where $intradist_S(N)$ is the average distance $dist_S(\vec{o}, \vec{q})$ between connected nodes $o, q \in N$ of the neighborhood in subspace S. $interdist_S(N)$ is the average distance $dist_S(\vec{o}, \vec{p})$ of nodes $o \in N$ to nodes $p \notin N$ outside the neighborhood.

We compute the outlier score of a node o when the merging process of its current neighborhood N_1 with another neighborhood N_2 does not increase the homogeneity, i.e., $hom_S(N_1) > hom_S(N_1 \cup N_2)$. This agglomerative process to compute the score allows to analyze the outlier property of nodes belonging to multiple neighborhoods as shown in Figure 1(a).
Given a homogeneous neighborhood and the congruent subspace, we can measure the local deviation of each object from its neighborhood in the congruent subspace. We measure the deviation of an object o w.r.t. the homogeneous neighborhood it belongs to and formalize the local attribute deviation as follows:

Definition 10: DistOut Score
Given a congruent subspace S, an object o, the neighborhood N it belongs to and the edge set $E_N = \{(u,v) \subseteq E \mid u, v \in N\}$, we define the outlier score as:

$$score(o, S) = \frac{\frac{1}{|\{(o,p) \in E_N\}|} \cdot \sum_{(o,p) \in E_N} dist_S(\vec{o}, \vec{p})}{\frac{1}{|E_N|} \cdot \sum_{(u,v) \in E_N} dist_S(\vec{u}, \vec{v})}$$

Following [5], we compare the average distance of a node to its direct neighbors with the average distance of all the connected nodes in the neighborhood in order to quantify the deviation of the object.

VII. Experiments

We evaluate quality, runtime, and parameterization of our approach on synthetic and real world datasets. We facilitate comparability and repeatability of our experiments for future research in this area by providing datasets and parameter settings on our website[1]. In our experiments we focus on the comparison of different subspace selection schemes: (1) no selection using the full attribute set A (*FullSpace*), (2) unsupervised feature selection (*LUFS*) [25], and (3) our congruent subspace selection (*ConSub*). For each of these pre-processing methods we apply community outlier detection (*CODA*) [10]. To ensure comparability in all respects, we have used identical settings for the outlier mining step (*CODA*). With this first setup we evaluate the quality of subspace selection.
Second, we also show results of *ConSub* with our new

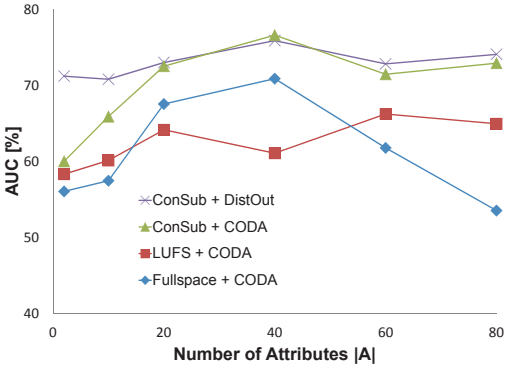


Fig. 2. Quality as a function of the number of attributes $|A|$ (*ConSub* settings: $k = 10$, $M = 150$ and $\alpha = 5\%$)

[1] http://www.ipd.kit.edu/~muellere/consub/

distance-based outlier model *DistOut* (cf. Definition 10), showing the full potential of our method. This setup demonstrates the benefits of congruent subspaces for an enhanced outlier model that exploits congruent subspaces for a distance-based outlier definition. For quality assessment we use the *area under the ROC curve* (AUC). For each position in the ranking, we compute the ratio of precision/recall and compute AUC as commonly used for the evaluation of outlier rankings [1].

A. Synthetic Data

We generate synthetic datasets of different size $|V|$, $|E|$ and dimensionality $|A|$. The generated graphs follow a power law distribution in order to reproduce the properties observed in real networks [13]. Attribute information is divided into relevant and irrelevant attributes (each 50% of $|A|$). For irrelevant attributes, nodes are assigned values from a uniform random distribution. Each relevant attribute can be part of several congruent subspaces. An attribute is merged with another subset of attributes to form a higher dimensional subspace with a probability of 20%. In these congruent subspaces, we assign nodes belonging to a community similar attribute values following a Gaussian distribution, and thus, fulfilling the assumption made by *CODA*. To ensure that there are community outliers, we randomly select 10% of cluster nodes and manipulate some of their attribute values in the congruent subspaces.

Quality: We evaluate the quality of our approach contingent on the number of attributes $|A|$. We depict average AUC values in Figure 2. We use the average results on three datasets, to reduce random effects in synthetic data generation. Comparing *FullSpace*, *LUFS* and *ConSub* with *CODA* as outlier mining, we clearly see an enhancement of community outlier mining by congruent subspaces obtained from *ConSub*. *CODA* shows many false positives and false negatives in both full space and for the features selected by *LUFS*. In particular, *LUFS* fails as a pre-processing for community outlier detection as it does not ensure congruence and does not allow different communities/outlier structures depending on different subspaces. Overall, our distance-based outlier detection *ConSub + DistOut* shows quality similar to *ConSub + CODA*. However, it is by far more efficient than *CODA* as shown in the following.

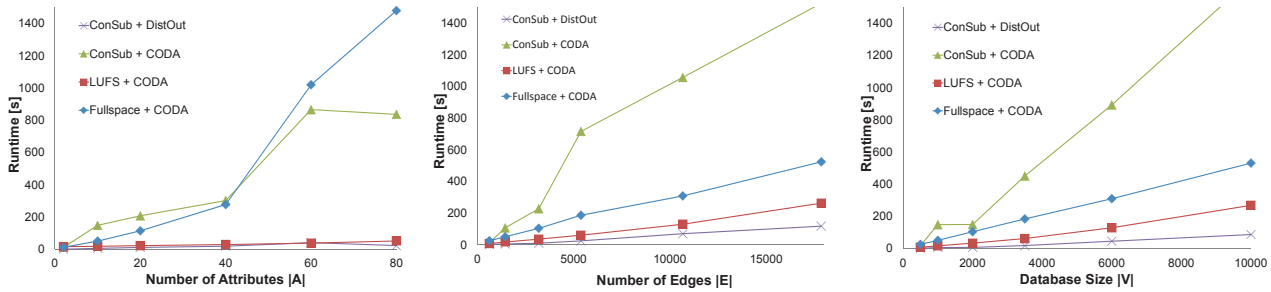
Runtime: Figure 3(a) shows the runtimes with increasing number of attributes. *ConSub + DistOut* and *LUFS + CODA* show best scalability w.r.t. $|A|$. However, *LUFS* selects a single subspace only, while *ConSub* outputs multiple subspaces. With *ConSub + DistOut*, we can analyze more subspaces within the same amount of time, and thus, reach better detection quality of community outliers that are hidden in different community structures determined by the underlying congruent subspace. Overall, *CODA* does not scale with the number of attributes $|A|$. The reason is that the matrix operations for multivariate likelihood functions of the underlying Gaussian distribution are costly and these matrix operations are executed for each subspace. Additionally, *DistOut* does not require an iterative algorithm to find the optimal neighborhood of a node due to the careful selection of the congruent subspaces. As a consequence, *DistOut* shows faster runtimes overall in comparison with *CODA*. Regarding our second set of scalability experiments w.r.t. number of nodes $|V|$ and number of edges $|E|$, we show results in Figure 3(b) and Figure 3(c). Similar to previous results, *ConSub + DistOut* analyzes multiple subspaces in substantially less runtime than the original *CODA* algorithm or *CODA* enhanced with feature selection algorithm *LUFS*.

Parameter Settings: The box plots in Figure 4 show an overview of quality results achieved on all synthetic datasets and highlights the robustness of our method w.r.t. parameterization. *ConSub* has three parameters: the significance level (α), the number of intervals (k) and the number of Monte Carlo iterations (M). We have evaluated the impact of each of these parameters on the quality and the runtime. We have run a variety of parameter settings on all synthetic datasets that have been used in previous experiments (cf. Figure 3). In total, we analyze each parameter setting on 36 datasets of different size and dimensionality. The influence of statistical fluctuations given by the number of Monte Carlo iterations M does not have a large impact on the quality if we run at least 150 iterations (cf. Figure 4(a)). However, more iterations result in higher runtimes (cf. Figure 4(d)) without a considerable increase in quality. We recommend to use $M = 150$ as a default value for this parameter, as used in all other experiments. The number of intervals k determines the sample size for the statistical test in each Monte Carlo iteration. An extremely low sample size induces a decrease in quality, but we observe a parametrization with good quality results for $k \geq 10$ as shown in Figure 4(b). Again, a larger sample size increases the runtime (cf. Figure 4(e)), but it does not increase quality substantially. We set this parameter to $k = 10$ as default value. The last parameter is the significance level α which controls the generation of higher dimensional subspaces as explained in Section V. High values $\alpha \geq 10\%$ induce considerably higher runtimes (cf. Figure 4(f)) as a large number of subspace candidates has to be processed. On the other hand, too restrictive values $\alpha \leq 1\%$ require considerably less time with a quality loss. In this case, the number of subspaces analyzed is too small. Thus, the choice of $\alpha = 5\%$ is a trade-off between quality and efficiency.

B. Real Data

We use four attributed graphs obtained from real world networks for the evaluation of our approach. We use the *Amazon* co-purchase network [14] consisting of product nodes with 28 attributes such as product prices, ratings, number of reviews, etc. Further, we use the *Enron* communication network with email transmission as edges between email addresses. Each node contains 20 attributes describing aggregated information about average content length, average number of recipients, or time range between two mails.

Evaluation based on given ground-truth: In order to present quality assessment we need some known outliers. Therefore, we derive three attributed graphs with known outliers and compare our detected outliers to this ground truth: *Disney* is a benchmark graph from a case study in [19], in which high school students have manually tagged unexpected Disney films in the Amazon co-purchase network. *Books* is a second graph out of the Amazon network, in which we use tagsprovided by Amazon users. In particular, we use the popular tag *amazonfail* as outlier ground-truth. This tag has been used for few years to let users express their disagreement with the sales ranks.We use products that were tagged by at least 20 users as outliers.

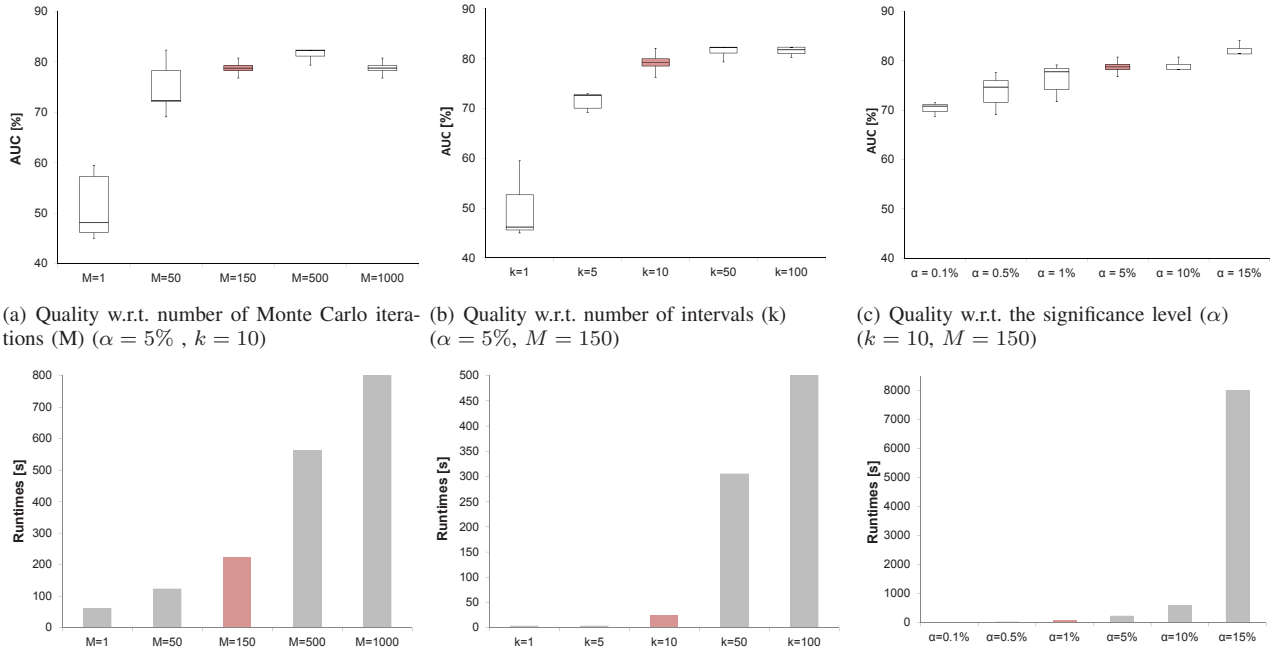


(a) Runtime w.r.t. number of attributes $|A|$, with fixed $|V| = 1000$

(b) Runtime w.r.t. number of edges $|E|$, with fixed $|A| = 10$

(c) Runtime w.r.t. number of nodes $|V|$, with fixed $|A| = 10$

Fig. 3. Runtime scalability w.r.t. $|A|$, $|E|$, and $|V|$ (*ConSub* settings: $k = 10$, $M = 150$ and $\alpha = 5\%$)



(a) Quality w.r.t. number of Monte Carlo iterations (M) ($\alpha = 5\%$, $k = 10$)

(b) Quality w.r.t. number of intervals (k) ($\alpha = 5\%$, $M = 150$)

(c) Quality w.r.t. the significance level (α) ($k = 10$, $M = 150$)

(d) Average runtime w.r.t. number of Monte Carlo iterations (M) ($\alpha = 5\%$, $k = 10$)

(e) Average runtime w.r.t. number of intervals (k) ($\alpha = 5\%$, $M = 150$)

(f) Average runtime w.r.t. significance level (α) ($k = 10$, $M = 150$)

Fig. 4. Parameter settings w.r.t. quality and runtimes. Evaluation on all (36) synthetic datasets with different dimensionalities and number of nodes

Dataset	Algorithm	AUC [%]	Runtimes [s]
Disney	ConSub + DistOut	81.77 ± 0.44	8.93
Nodes:124	ConSub + CODA [10]	67.97 ± 5.8	152.66
Edges:333	LUFS [25] + CODA [10]	44.44 ± 13.5	3.46
Attributes:28	*Fullspace* + CODA [10]	50 ± 0	6.05
Books	ConSub + DistOut	60.02 ± 0.49	2.15
Nodes:1,418	ConSub + CODA [10]	53.52 ± 2.25	14.81
Edges: 3,695	LUFS [25] + CODA [10]	-	
Attributes:28	*Fullspace* + CODA [10]	53.35 ± 0	36.14
Enron	ConSub + DistOut	74.8 ± 0.08	840.54
Nodes: 13,533	ConSub + CODA [10]	60.8 ± 6.98	1130.78
Edges:176,987	LUFS [25] + CODA [10]	48.3 ± 5.48	472.6
Attributes:20	*Fullspace* + CODA [10]	45.7126 ± 0	397.33

TABLE I. QUALITY AND RUNTIME ON REAL WORLD NETWORKS

Enron is our third graph using spammers as known outlier ground-truth [17].

Since most of the approaches are non deterministic, we have executed each algorithm 20 times in order to reduce possible random effects. Table I shows the average AUC values and their standard deviations. For all real world datasets, we observe that some attributes did not show any congruence with the graph structure independent of the parametrization (e.g., sales rank in the *Amazon* network or the average content length for the *Enron* network). This indicates that homophily does not hold in the full space. *CODA* has low quality as it is based on the homophily assumption. However, CODA can be enhanced considerably by the selection of congruent subspaces of *ConSub*. Regarding the new outlier model proposed (*ConSub + DistOut*), it obtains the best results and it is the most robust since the fluctuations between different executions are the lowest ones. *ConSub* can find subspaces where the graph structure and attribute information have dependencies and improves the detection of outliers accordingly. Our subspace selection scheme not only outperforms the other algorithm in terms of average AUC, it also shows robust results with low variance. Similarly to synthetic data *ConSub + DistOut* shows efficient runtime.

Subspaces derive novel insights: To discuss novel knowledge extracted by *ConSub*, we depict results from the largest connected component of the Amazon network with 314,824 nodes and 882,930 edges in Table II. *ConSub* retrieves eight one-dimensional subspaces and three two-dimensional subspaces showing congruence with the graph structure considering a significance level of 1%. Besides the use as pre-processing step, this result is informative regarding the network and its dependencies. We observe that the dependencies between some ratings (e.g., *Rating 5*) and the ratio of helpful votes from the reviews are also congruent with the graph structure. This means that two products are often co-purchased if they have similar number of ratings and similar ratio of helpful votes (e.g., *Rating 5* and *Helpful Votes* appear in our congruent subspace set). The selection of congruent subspaces is not only relevant for outlier mining in order to ensure the underlying homophily assumption, it also provides novel knowledge about the dependencies between node attributes and the graph structure.

	1d-Subspaces	**2d-Subspaces**
Nodes: 314,824	*Rating 1*	*Rating 1 - Helpful Votes*
Edges: 882,930	*Rating 2*	
Attributes: 28	*Rating 3*	
Level of Significance: 1%	*Rating 4*	*Rating 4 - Helpful Votes*
$\mathbf{M} = 150, \mathbf{k} = 10$	*Rating 5*	*Rating 5 - Helpful Votes*
Runtime: 5160.2 s	*Average Rating*	
	Number of reviews	
	Helpful votes	

TABLE II. CONGRUENT SUBSPACES IN AMAZON NETWORK

VIII. CONCLUSION

With this work, we tackle the general problem of subspace selection in attributed graphs. We propose the novel notion of *congruent subspaces* that captures the dependency between node attributes and the edge structure of a graph. As our main contribution, we develop a statistical selection of congruent subspaces, and define a general measure that assesses the degree of congruence. We evaluate our subspace selection scheme on community outlier mining, a graph mining task relying on dependency between attributes and edges. We show that *ConSub* outperforms traditional full space outlier detection and recent feature selection. Nevertheless, outlier mining is only one graph mining task. As general pre-processing step, *ConSub* can also be used for clustering or pattern mining algorithms, which utilize both graph and attribute information and rely on the homophily assumption. For future research, we aim to provide selection schemes for a mixture of attribute types such as categorical, binary or continuous values. Additionally, we would like to explore extensions of our subspace selection scheme into unsupervised mining tasks but also for semi-supervised tasks such as link prediction or label propagation. We are convinced that subspace selection can be a useful pre-processing step for these and other graph mining paradigms.

ACKNOWLEDGMENTS

This work is supported by the Young Investigator Group program of KIT as part of the German Excellence Initiative, by a post-doctoral fellowship of the research foundation Flanders (FWO), and by the German Research Foundation (DFG) within IME Graduate School at KIT.

REFERENCES

[1] C. Aggarwal. *Outlier Analysis*. Springer, 2013.

[2] C. C. Aggarwal. Outlier ensembles: Position paper. *SIGKDD Explorations*, 14(2):49–58, 2012.

[3] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, pages 207–216, 1993.

[4] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos. PICS: Parameter-free identification of cohesive subgroups in large attributed graphs. In *SIAM SDM*, pages 439–450, 2012.

[5] M. Breunig, H. Kriegel, R. Ng, J. Sander, et al. LOF: identifying density-based local outliers. *Sigmod Record*, 29(2):93–104, 2000.

[6] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 2009.

[7] C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *ACM SIGKDD*, pages 84–93, 1999.

[8] M. Davis, W. Liu, P. Miller, and G. Redpath. Detecting anomalies in graphs with numeric labels. In *ACM CIKM*, pages 1197–1202, 2011.

[9] P. Erdős and A. Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl*, 5:17–61, 1960.

[10] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han. On community outliers and their efficient detection in information networks. In *ACM SIGKDD*, pages 813–822, 2010.

[11] S. Günnemann, I. Färber, B. Boden, and T. Seidl. Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. In *IEEE ICDM*, pages 845–850, 2010.

[12] F. Keller, E. Müller, and K. Böhm. HiCS: High contrast subspaces for density-based outlier ranking. In *IEEE ICDE*, pages 1037–1048, 2012.

[13] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110, 2008.

[14] J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007.

[15] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *Advances in Neural Information Processing Systems 25*, pages 548–556, 2012.

[16] M. McPherson, L. S. Lovin, and J. M. Cook. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[17] V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam filtering with naive bayes-which naive bayes. In *CEAS*, volume 17, pages 28–69, 2006.

[18] E. Müller, I. Assent, P. I. Sanchez, Y. Mülle, and K. Böhm. Outlier ranking via subspace analysis in multiple views of the data. In *IEEE ICDM*, pages 529–538, 2012.

[19] E. Müller, P. Iglesias, Y. Mülle, and K. Böhm. Ranking outlier nodes in subspaces of attributed graphs. In *Workshop on Graph Data Management at IEEE ICDE*, 2013.

[20] E. Müller, M. Schiffer, and T. Seidl. Statistical selection of relevant subspace projections for outlier ranking. In *ICDE*, pages 434–445, 2011.

[21] M. Newman. Modularity and community structure in networks. *National Academy of Sciences*, 103(23):8577–8582, 2006.

[22] M. E. Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, 2003.

[23] M. Shiga, I. Takigawa, and H. Mamitsuka. A spectral clustering approach to optimally combining numericalvectors with a modular network. In *ACM SIGKDD*, pages 647–656, 2007.

[24] A. Silva, W. Meira Jr, and M. Zaki. Mining attribute-structure correlated patterns in large attributed graphs. *PVLDB*, 5(5):466–477, 2012.

[25] J. Tang and H. Liu. Unsupervised feature selection for linked social media data. In *ACM SIGKDD*, pages 904–912, 2012.

[26] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[27] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *PVLDB*, 2(1):718–729, 2009.