Context Selection on Attributed Graphs for Outlier and Community Detection

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften

der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte Dissertation

von
Patricia Iglesias Sánchez
aus Caracas

Datum der mündlichen Prüfung:	11. Mai 2015
Erster Gutachter:	Prof. Dr.-Ing. Klemens Böhm
Zweiter Gutachter:	Prof. Dr. Dorothea Wagner
Dritter Gutachter:	Dr. rer. nat. Emmanuel Müller

Acknowledgements

There are many people who supported me in my work during the past years. First, I want to thank my supervisor Prof. Dr.-Ing. Klemens Böhm for giving me the opportunity to work on an interesting research topic as part of his research group. Many thanks also go to Prof. Dr. Dorothea Wagner for being my second reviewer and for the fruitful discussions with her research group.

I would like to express my sincere gratitude to my third supervisor Dr. rer. nat. Emmanuel Müller for his continuous support, for his patience and motivation. His guidance helped me in all the time of research and writing of this thesis.

Most of the research approaches developed in this thesis are the result of team work. I want to thank everyone who worked together with me. Also many thanks to my student assistants and graduands for their support in the implementation of some of the solutions in this thesis. I also thank my colleagues for the interesting discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last four years.

Last but not the least, I want to especially thank my family for having supported me during all these times. In particular, I want to thank my husband for all his support.

Abstract (English Version)

Today's applications store large amounts of complex data that combine information of different types. In particular, attributed graphs are an example for such a complex database. They 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 characterized by its relationships to other objects (graph structure) and its individual properties (node attributes). For instance, social networks store *friendship relations* as edges and *age*, *income*, and several other properties as attributes. Specifically, each node in an attributed graph may be characterized by a large number of attributes. Given such a complex database consisting of a high dimensional attributed graph, the goal of data mining is to extract automatically patterns combining both sources of information.

Approaches for outlier and community detection on attributed graphs require to capture a group of similar objects w.r.t. both the graph structure and the attribute values. To achieve this, existing approaches exploit the assumption that connected nodes tend to have similar characteristics. This effect is known as *homophily*. However, this assumption may be not fulfilled for the full space of attributes. Some attributes may be highly dependent with the graph structure while others do not show any dependency. For instance, the attribute *shoe size* is not dependent on the existing friendships within a social network. Thus, it shows almost random values when combining its information with the graph structure. Such irrelevant attributes heavily deteriorate the quality of data mining techniques on attributed graphs that consider all attributes. Therefore, combining both information sources for mining attributed graphs requires to select only the relevant ones.

Thereby, the core challenge is the modeling of such a selection. In this work, we call this selection *context*. It consists of a set of connected nodes and a set of relevant attributes that show dependencies with the graph structure. Simultaneously, attributed graphs represent complex data structures where not only the graph size may be large, but also the number of attributes. Thus, the development of novel approaches implies novel algorithmic solutions which are scalable to large and high dimensional attributed graphs. Besides the design of novel models and algorithmic solutions for context selection, the evaluation on real-world data is a relevant issue. However, public benchmarks are not available and, thus, an evaluation using qualitative measures instead of only providing

anecdotal descriptions is not possible. Overall, labeling outliers or communities on real world networks to get the ground truth for the evaluation still represents an open challenge for the data mining community.

This thesis aims to cope with all these challenges. To achieve this, it does not only introduce novel models and algorithms, but it also provides several use cases and benchmarks for the evaluation of outlier mining techniques on attributed graphs. A *context* can be specified in multiple ways depending on the data mining task or the underlying community or outlier definition. Thus, this work introduces several context selection schemes for mining attributed graphs. We bring them together in a taxonomy and compare existing ones with our novel schemes. In particular, this thesis proposes different algorithms that focus on outlier analysis. We classify them in two main categories based on their context selection scheme: *model-dependent* and *generic* context selection.

Model-dependent Context Selection

A model-dependent context selection extracts the relevant attributes depending on the underlying cluster or outlier definition. In this part of the thesis, the proposed approaches are generalizations of well-known traditional models for community or outlier detection to attributed graphs. In particular, the goal is to leverage research efforts from traditional research on vector or graph data by including an efficient context selection for attributed graphs. First, we propose a novel measure called *attribute-aware modularity* which is a generalization of the well-established measure *modularity* for graph clustering. As one of its main properties, it unifies the information of the graph structure with the attribute values being simultaneously robust w.r.t. irrelevant attributes and outlier nodes due to its context selection. In addition to this, we ensure its incremental and numerically stable calculation. This allows the design of efficient algorithms for community and outlier detection on attributed graphs.

The second approach generalizes a well-established model for outlier ranking on vector data where the deviation of each object can be computed comparing each node with its own neighborhood. In order to avoid irrelevant attributes, we locally select the relevant ones of each node neighborhood. Furthermore, we propose a ranking function that does not only measure the deviation of the attribute values, it also combines the information of the graph structure.

In summary, both approaches focus on an efficient context selection that scales linearly with the number of attributes.

Generic Context Selection

Considering a generic selection scheme leads to more general and flexible approaches that can be applied independently from the model used (e.g., outlier or cluster defini-

tion). The most relevant property of such techniques is that their independent design from the outlier or community definition enables the flexibility to apply different mining techniques for attributed graphs. In this part, the thesis proposes a pre-processing step for full space approaches for mining attributed graphs. To achieve this, this technique selects subspaces called *congruent subspaces*, i.e., subsets of attributes, showing dependencies with the graph. The goal is to improve existing techniques that deteriorates with the lack of homophily in the full attribute space.

Following the paradigm of subspace selection, we finally propose and evaluate several ranking functions for outlier mining on attributed graphs. The main property of these functions is that the context selection is based on subspace clustering techniques for attributed graphs. Thus, the functions are independent from the cluster model and further research improvements on clustering attributed graphs can be exploited by the proposed functions.

Overall, the approaches presented in this part are general models for context selection that are based on subspace analysis. This means that they are able to extract multiple contexts out of the attributed graph.

Abstract (German Version)

Heutige Anwendungen speichern große und komplexe Datenmengen, die unterschiedliche Arten von Information kombinieren. Insbesondere sind attributierte Graphen ein Beispiel für eine solche komplexe Datenbank. Sie werden weitgehend genutzt, um soziale Netzwerke, Protein-Protein-Interaktionen, Kommunikationsnetze oder Kaufverhaltensstrukturen darzustellen. Jedes Objekt wird durch seine Beziehungen zu anderen Objekten (Graphstruktur) und seine individuellen Eigenschaften (Attribute der Knoten) beschrieben. In sozialen Netzwerken werden beispielsweise Freundschaftsbeziehungen als Kanten und Personeneigenschaften wie *Alter* oder *Gehalt* als Attribute betrachtet. Zudem werden die Knoten meist mit einer sehr großen Attributzahl beschriftet. In einer solch komplexen Datenbank, die aus einem hochdimensionalen attributierten Graphen besteht, ist das Ziel von Data Mining Verfahren, automatisch Muster aus beiden Informationsquellen zu extrahieren.

Ansätze für Ausreißererkennung und Clustering für attributierte Graphen fordern, dass Objekte bezüglich beider Informationsquellen, der Graphstruktur und den Attributwerten, ähnlich sind. Um das zu erreichen, nutzen existierende Verfahren die Annahme, dass verbundene Knoten die Tendenz haben, ähnliche Eigenschaften zu besitzen. Dieser Effekt wird als *Homophilie* bezeichnet. Jedoch wird diese Annahme nicht für den volldimensionalen Raum der Attribute erfüllt. Einige Attributwerte können von der Graphstruktur abhängen, während andere keine Abhängigkeiten mit dem Graph zeigen. Manche Attributwerte können sogar einen entgegengesetzten Trend verglichen mit den verbundenen Knoten zeigen. Ein Beispiel dafür ist das Attribute *Schuhgröße*, da dessen Attributwerte nicht von existierenden Freundschaften abhängen. Wenn man dieses Attribut mit der Graphstruktur kombiniert, sehen die Attributwerte dadurch fast zufällig aus. Solch irrelevante Attribute verschlechtern die Qualität der Ansätze, die alle Attribute nutzen. Deshalb dürfen nur die relevanten Attribute betrachtet werden, wenn man beide Informationsquellen für die Datenanalyse zusammenführt.

Die Kernherausforderung ist dabei die Selektion der Attribute zu modellieren. In dieser Arbeit bezeichnen wir diese Selektion als *Kontext*, der aus einer Menge von verbundenen Knoten und einer Menge relevanter Attribute besteht. Darüber hinaus stellen attributierte Graphen komplexe Datenstrukturen dar, bei denen nicht nur die Größe des Graphen eine Herausforderung darstellt, sondern auch die Anzahl der Attribute in jedem Knoten. Deshalb erfordert die Entwicklung neuer Verfahren effiziente algorithmische Lösungen, die auf solchen Datenmengen eingesetzt werden können. Zudem ist

die Evaluation dieser Techniken auf realen Daten eine offene Forschungsfrage, da existierende reale Daten eine gegebene Ground Truth nicht beinhalten. Als Konsequenz kann man keine qualitative Maße für die Evaluation berechnen. Daher hat sich eine Evaluation durch anekdotische Fallstudien und Beschreibungen etabliert. Insgesamt ist die Beschriftung von Ausreißern und die Evaluation auf realen Daten noch eine offene Herausforderung der Data-Mining-Forschung.

In dieser Arbeit wollen wir alle diese Herausforderungen behandeln. Um das zu erreichen, stellen wir sowohl neue Modelle und Algorithmen als auch neue Fallstudien und Benchmarks für die Evaluation von Outlier Mining in attributierten Graphen vor. Abhängig von der Data-Mining-Aufgabe und der grundlegenden Ausreißer- oder Clusterdefinition kann man einen Kontext unterschiedlich bestimmen. Deshalb präsentieren wir verschiedene Algorithmen für die Kontextselektion, die die Analyse aus verschiedenen Sichtweisen von attributierten Graphen ermöglichen. Anhand einer von uns vorgestellten Taxonomie diskutieren wir den Unterschied zu existierenden Verfahren. Die in dieser Doktorarbeit vorgestellten Ansätze beschäftigen sich hauptsächlich mit Ausreißererkennung. Wir gruppieren die vorgestellten Algorithmen für Kontextseletion in zwei grossen Kategorien: *Modellabhängige Kontextselektion* und *Generische Kontextselektion*, die wir im Folgenden einführen werden.

Modellabhängige Kontextselektion

Ein Verfahren mit einer modellabhängigen Kontextselektion extrahiert relevante Attribute abhängig von der zugrundeliegenden Ausreißer- oder Clusterdefinition. In diesem Teil der Arbeit schlagen wir Ansätze vor, die etablierte traditionelle Modelle für attributierte Graphen generalisieren. Das Ziel ist dabei etablierte Konzepte der traditionellen Forschungsrichtungen von Graphstrukturen oder relationalen Daten auszunutzen. Um das zu erreichen, generalisieren wir diese Modelle durch die Einbeziehung einer geeigneten und effizienten Kontextselektion für attributierte Graphen. Zuerst schlagen wir ein neues Maß für das Clustering von attributierten Graphen vor, das wir als *attribute-aware modularity* bezeichnen. Dieses neues Maß ist eine Verallgemeinerung des etablierten Maßes *modularity*, das zur Clusterung auf Basis von Graphstrukturen entwickelt wurde. Im Vergleich zu existierenden Ansätzen besteht der wesentliche Vorteil dieses neuen entwickelten Maßes darin, dass es sehr robust ist, sowohl bezüglich irrelevanter Attribute als auch Ausreißer. Darüber hinaus versichern wir eine inkrementelle und numerisch stabile Berechnung dieses neuen Maßes. Dadurch ermöglichen wir die Entwicklung von effizienten Algorithmen zur Clusterung und Ausreißererkennung in attributierten Graphen.

Der zweite vorgeschlagene Ansatz in diesem Teil der Doktorarbeit verallgemeinert ein etabliertes Modell für das Ranking von Ausreißern auf relationalen Daten. Die Abweichung jedes Knoten wird bestimmt, in dem man diesen Knoten mit seiner eigenen Nachbarschaft vergleicht. Um mit irrelevanten Attributen umgehen zu können, führen

wir eine Kontextselektion ein, bei der für jede Nachbarschaft im Graphen die passenden Attribute ausgewählt werden. Dadurch können wir die Abweichung von jedem Objekt präziser bestimmen. Zudem stellen wir eine neue Ranking-Funktion vor, die sowohl die Abweichungen in den Attributwerten als auch die Information der Graphstruktur kombiniert.

Zusammengefasst stellt dieser Teil der Arbeit zwei effiziente Algorithmen vor, die eine lineare Skalierbarkeit bezüglich der Attribute besitzen.

Generische Kontextselektion

Im Gegensatz zur modellabhängigen Kontextselektion kann man Ansätze entwickeln, die unabhängig von der Ausreißer- oder Clusterdefinition sind. Eine solche generische Kontextselektion ermöglicht allgemeinere und flexiblere Ansätze. Diese Eigenschaft erlaubt existierende Verfahren zu verbessern oder zu erweitern. In diesem Teil der Arbeit führen wir zunächst einen Ansatz für ein Preprocessing zur Attributauswahl ein, der für volldimensionale Ansätze zur Clusterung oder Ausreißererkennung betrachtet werden soll. Das Ziel ist dabei existierende Verfahren zu verbessern, die die Annahme der Homophilie für die gegebenen Attribute brauchen. Um das zu erreichen, führen wir das Konzept von *congruent subspaces* ein. Unser Verfahren selektiert Teilmengen der Attribute, die Abhängigkeiten zur Graphstruktur zeigen. Diese Attributteilmengen erfüllen die Annahme der Homophilie für den gegebenen Graph. Eine große Herausforderung ist dabei effiziente Heuristiken zu entwickeln, die die Analyse von Attributteilmengen in exponentieller Anzahl vermeiden.

Im Gegensatz zu dieser vorgeschlagenen globalen Kontextselektion fokussieren sich existierende Verfahren auf eine lokale Kontextselektion bezüglich eines bestimmten Teilgraphs. Um diese lokalen Modelle für Ausreißererkennung zu verwenden, haben wir neue Ranking-Funktionen entwickelt, die die Clustering-Ergebnisse solcher Ansätze analysieren. Diese Funktionen sind unabhängig vom zugrundeliegenden Cluster-Modell. Deshalb können sie für beliebig viele Verfahren eingesetzt werden und dabei zukünftige Entwicklungen in dieser Richtung ermöglichen.

In diesem Teil der Arbeit stellen wir insgesamt zwei unterschiedliche Verfahren vor, die sich auf Analyse von Attributteilmengen fokussieren. Das bedeutet, dass sie in der Lage sind, mehrere Kontexte aus dem attributierten Graphen zu extrahieren.

Contents

1. Overview

1.1. Introduction

Currently, not only a large amount of data is being collected in today's applications, but also the information stored is heterogeneous. For example, social networks consist of individual person characteristics and the friendship relations between the users. Hence, each object is described by attributes and, simultaneously, it is connected to other objects by the relationships between them (graph structure). Such heterogeneous databases are also known as attributed graphs. They are widely used for the representation of social networks, gene and protein interactions, communication networks, or product co-purchase in web stores. The data volume as well as its heterogeneity does not allow a manual analysis to extract hidden knowledge in these databases. Therefore, the recognition of useful and previously unknown patterns has to be done automatically.

In this thesis, we focus on two well-established data mining tasks: outlier and community detection. Outlier analysis is an important task that aims to detect unexpected, rare, and suspicious objects. Network intrusion, rare protein interactions and financial fraud are possible applications of outlier mining on attributed graphs. In particular, we consider electronic platforms as exemplary application of outlier mining on attributed graphs. Electronic marketplaces try to detect and delete fraudulent product placements since their reputation is highly affected by such fraud. Fake products, overpriced products, or manipulated reviews are examples for outliers that have to be detected. Such electronic platforms provide a large number of descriptive *attributes* for each product (e.g., prices of all sellers, ratings, and product reviews) and, simultaneously, the product relations stored in the *graph* of frequently co-purchased products. All this heterogeneous information adds more insights for outlier analysis. However, its complexity does not allow a manual analysis for the comparison between all related products in order to recognize those that are suspicious.

In contrast to outlier mining, the goal of community detection is to group the nodes in the network which are similar to each other, while objects located in other groups are dissimilar. The detection of community structures has been shown to be relevant for social network analysis, group of together co-purchased products or web analytics. Figure 1.1 shows an example in a social network of two groups (*high executives* and the *elite athletes*). Each of these two communities is similar w.r.t. both the connections

between the persons in the community and attribute values (e.g. similar *income* for the high executives and similar *weight* for the elite athletes). Figure 1.1 also depicts an example for outlier. In the group of the elite athletes, the trainer has highly deviating attributes values (e.g., *weight* and *sport hours*) although he is well connected within the group of athletes. Both the community structure and the outlier are only recognizable if both sources of information (graph structure and attributes) are combined. Traditional approaches for outlier or community detection have focused on either vector data or graph structures. However, attributed graphs demand data analysis in combination of both.

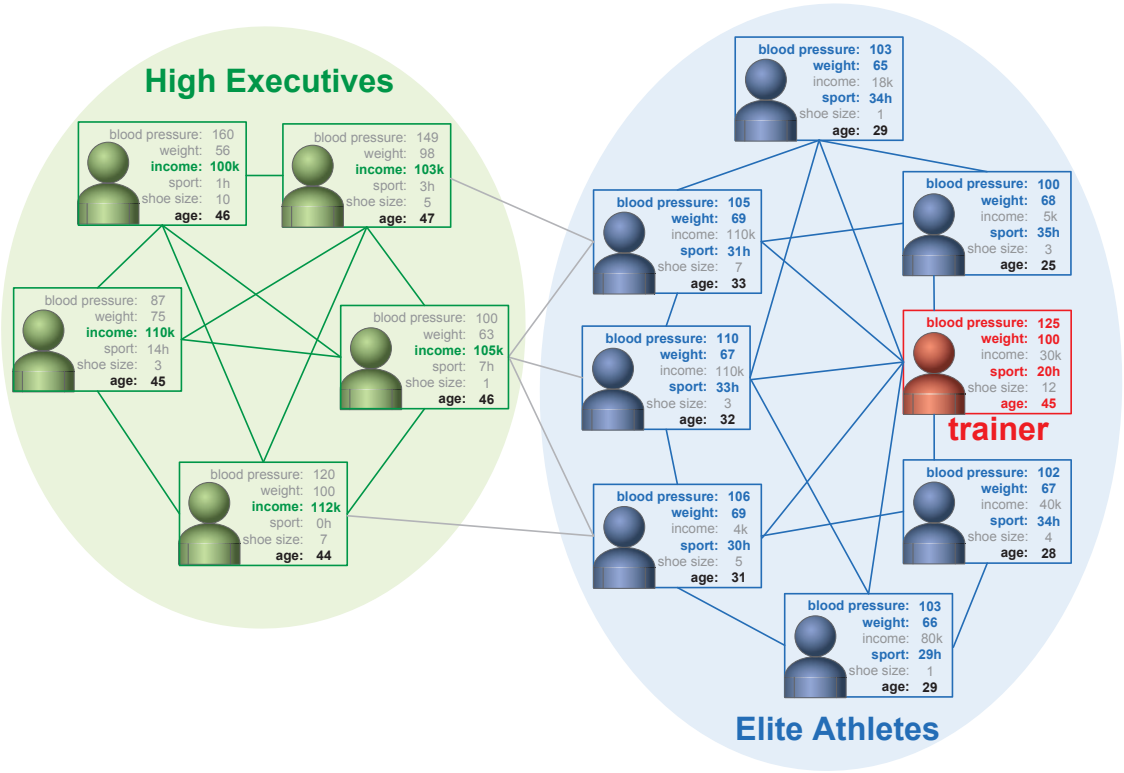


Figure 1.1.: Toy example of a social network with two communities: high executives and elite athletes. It also shows an example of outlier marked as trainer.

The goal of mining techniques for attributed graphs is to unify both sources. To achieve this, existing approaches exploit a well-known effect which is known as *homophily* [MSLC01]. This phenomena ensures that connected nodes tend to have similar characteristics, i.e., attribute values. Nevertheless, this does not occur for all attributes as shown in [New03]. For instance, characteristics such as *shoe size* are not dependent on the existing relationships in the graph. As a result, these attributes show scattered attribute values if they are combined with the graph structure. Following our toy example shown in Figure 1.1, the attribute *shoe size* shows scattered values for both communities depicted in the graph. Thus, this attribute information is contradicting with the

graph structure. This issue raises a major challenge for the simultaneous mining of both information sources. Existing full dimensional techniques for attributed graphs [GLF+10, STM07, XKW+12, Vie12, ZCY09, ZCY10] assume that all the attributes follow the homophily assumption. In contrast to this, this thesis focuses on the design of novel techniques that are aware of the existence of such irrelevant attributes. Specifically, our approaches consider only those attributes showing dependencies with a set of nodes in the graph structure. We call this *context selection*.

Depending on the underlying data mining task, the dependencies between the graph structure and the attribute information can be formalized in multiple forms. Some attributes may be correlated with the entire graph structure [New03] (*global dependency*) while other attributes only reveal local dependencies with the graph structure [GFBS10, GBS11, ATMF12] (*local dependency*). For instance, the attribute *age* presents a global dependency since it has homogeneous attribute values for the two existing communities in the graph (*high executives* and *elite athletes*). On the other hand, attributes such as *blood pressure* or *income* display similar attribute values only within the communities and scattered values in the remaining graph (cf. Figure 1.1). Both cases are two examples of possible specifications for context selection according to the graph perspective. In this thesis, we analyze different context specifications and introduce a novel taxonomy describing possible selection schemes that enable the analysis of attributed graphs from different perspectives.

Further, we can map our approaches onto different stages of the *KDD process* according to the abstraction level of their context formalization. The *KDD process* is a common methodology for the Knowledge Discovery in Databases in data mining [HKP11]. Figure 1.2 depicts the different steps of this process and an overview of our contributions categorized by the step they are mapped onto.

In some cases a pre-processing step is required where data is cleaned (e.g., outliers have to be deleted) or it is prepared (e.g., feature selection). This is particularly true and essential for those algorithms that make previous assumptions about the input data like full dimensional approaches for attributed graphs [ZCY09, ZCY10, ZCY10, STM07, Vie12, XKW+12, HZZL02, GLF+10]. They presume that the effect of homophily is fulfilled for all the attributes and, thus, they require a pre-processing step that selects the relevant attributes such as *age* or *income* showing dependencies with the graph structure. Nevertheless, the literature has not addressed this issue yet. An important consideration in the design of such techniques is that the context selection has to be as generic as possible in order to apply other techniques regardless their underlying model (i.e., cluster or outlier model). In this thesis, we categorize the contexts of such approaches as *generic context selection*.

In contrast to this, existing work on attributed graphs has focused in the data mining step [GFBS10, GBS11, GBFS13, GFRS13, ATMF12, YML13]. They provide multiple context selection schemes which depend on an underlying an underlying cluster model (*model-dependent context selection*). These approaches aim to extract communities

such as *high executives* and *elite athletes* from our example, but they do not have considered contradicting effects such as outlier nodes. Conversely, the approaches in this work are aware of outliers. In particular, we introduce novel model-dependent schemes for outlier detection and propose post-processing steps based on generic context selection schemes.

In general, the development of techniques requires an evaluation step where a quality assessment of the approaches is done. This can be done by manually analyzing and comparing the results with case studies. However, this tends to be subjective. In contrast to this, we design and provide the first benchmarks for the evaluation of outlier mining techniques on attributed graphs. This allows us to do an accurate quality assessment of our techniques.

Overall, this thesis provides different approaches which we map to the KDD process for mining attributed graphs. This enables an user to choose a context selection scheme depending on the application demands or the underlying requirements of the data mining task. In particular, we focus on selection schemes for community and outlier detection since both are well-established techniques for the knowledge discovery process. To achieve this, we introduced multiple concepts and schemes for context selection on attributed graphs. In the following, we describe the main challenges of mining attributed graphs in Section 1.2 and we give an overview over the contributions and structure of this thesis in Section 1.3.

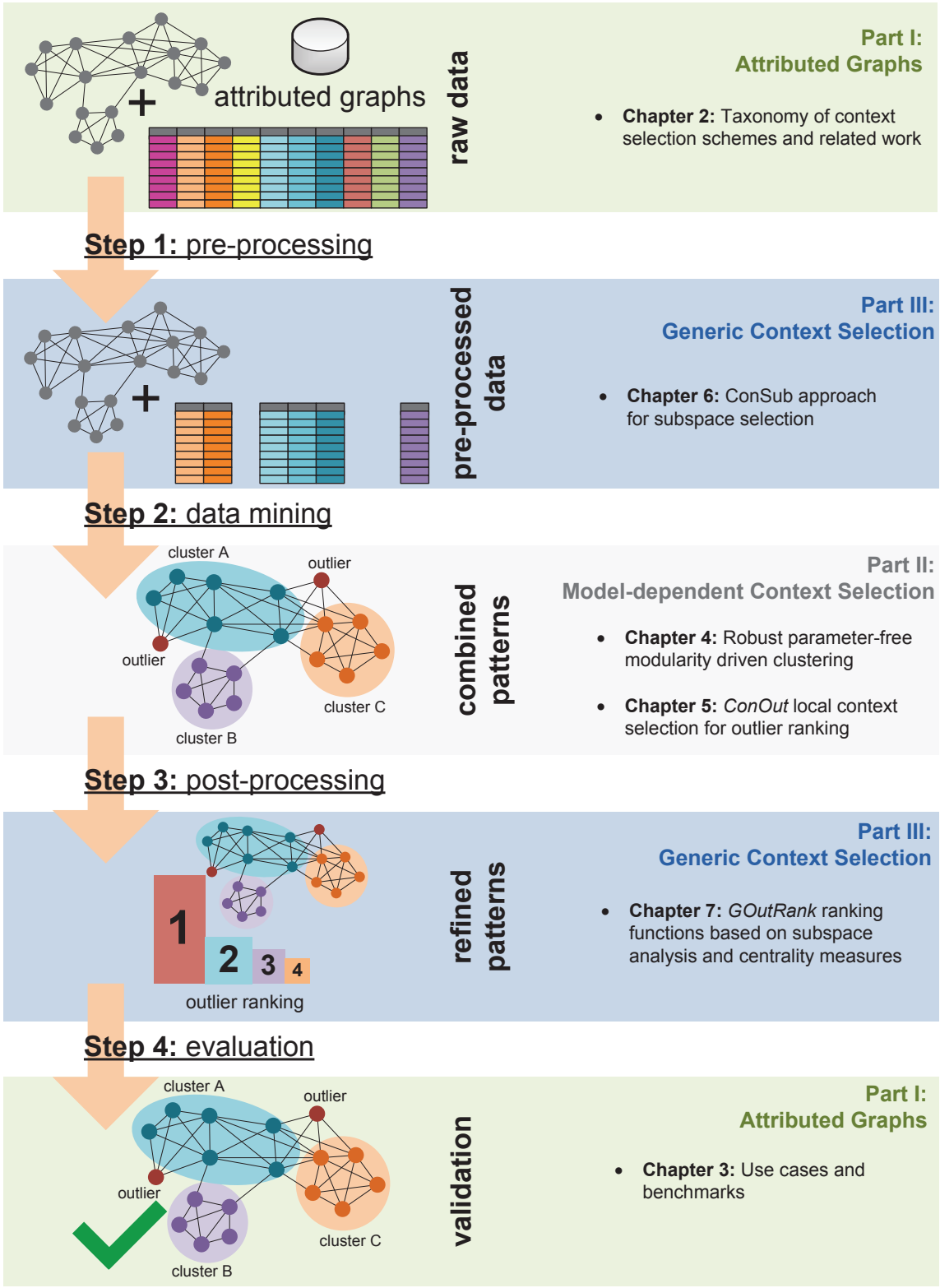


Figure 1.2.: Overview of thesis contributions categorized by the step of the KDD process they are mapped onto

1.2. Challenges of Mining Attributed Graphs

Combining the information of relational data with the graph structure for mining attributed graphs leads to several challenges. Below, we discuss them in detail since tackling these challenges is an important objective of this thesis.

Heterogeneity Mining attributed graphs requires to combine the information of the graph structure and the attribute values. As a consequence of this combination, the extracted patterns have to fulfill the constraints of the graph structure as well as the requirements of the attribute values. However, the attribute information may contradict the information given by the graph structure. For instance, a group of nodes with similar attribute values does not always correspond to a well-defined graph cluster. Thus, a major challenge of mining attributed graphs is to achieve a trade-off between the basic properties of each source. Furthermore, the combination of both sources requires to enhance traditional definitions (e.g., outlier) in order to consider both relational and graph data. For instance, an outlier in an attributed graph is not only an object with highly deviating values, but it is also connected to the graph structure. Overall, mining attributed graphs inherits the challenges of the traditional research areas of graph and relational data, but also adds new challenges due to the combination of both sources.

Robustness The combination of attributes and graph structure is only possible when the homophily effect is fulfilled. This means that connected nodes tend to have similar attribute values. Nevertheless, this phenomena does not occur for all the attributes (e.g., *shoe size*). In particular, it is important for outlier and community detection on attributed graphs to extract groups of nodes showing similarities w.r.t. both the graph structure and the attribute values. However, irrelevant attributes not showing dependencies with the graph structure hinder the detection of such heterogeneous groups. Further, outliers also represent contradicting effects. They are strong embedded in the graph structure as shown in Figure 1.1, but they show highly deviating attribute values in the relevant ones. As a consequence, the core challenge arises when modeling the dependencies between the graph structure and the attributes in order to detect the relevant attributes. Without this selection, the techniques can not be aware of the contradicting effects caused by both irrelevant attributes and outliers. Overall, approaches have to be be robust w.r.t. to both outliers and irrelevant attributes.

Multiple Views When considering different subset of attributes with the graph structure, distinct cluster or outlier structures may appear. For instance, the subspace consisting of the attributes *age* and *income* leads to a group of *high executives*. On the other hand, the new cluster *star wars fans* appears if the attribute *favorite films* is considered. In other words, each node in the attributed graph may belong to different clusters or it may be an outlier depending on the considered subspace. While a single projection enforces each object to be assigned to a cluster, these multiple views provide more information out of the data since they allow the detection of multiple concepts.

Efficiency Due to the increasing volume of data in today's applications (e.g., social media), efficiency is a major requirement for algorithms. In contrast to traditional approaches for either graph or relational data, approaches for mining attributed graphs have to scale w.r.t. both the graph size and number of attributes. Thus, mining attributed graphs requires to consider both sources efficiently in the algorithmic design. Specifically, the large number of attributes adds a further challenge when considering several subset of attributes instead of a single projection. To extract multiple views out of the data requires to analyze the dependencies between each possible subset of attributes and the graph structure. The large number of node attributes induces a huge search space due to the exponential number of possible subspaces. This is particularly true when considering real-world data. For instance, each node of the *Amazon recommendation network* contains 28 attributes (2^{28} subsets). This is a core challenge for subspace selection techniques and the development of heuristics is still an open challenge for attributed graphs in order to avoid this explosion of combinations.

Evaluation An important stage of the *KDD process* (cf. Figure 1.2) is the validation of the data mining results. One possible procedure for the evaluation is to verify the obtained results by experts and, then, describe them. However, the comparison between different algorithms is difficult when validating the results. On the other hand, a given ground truth enables to compute quality measures that allows a quantitative quality assessment. However, obtaining these labels from experts on real-world data is difficult. First, the costs of this process are high due to the required human resources and their time for doing the analysis. Second, the generation of benchmarks for attributed graphs is particularly challenging not only due to the database size (both graph size and attribute dimensionality), but the complexity induced by the combination of two sources of information hinders a good labeling of the data. Overall, the literature has not addressed the design and implementation of benchmarks for outlier mining on attributed graphs.

1.3. Contributions and Outline

This thesis introduces novel models and algorithms for outlier and community detection on attributed graphs. Part I describes the most relevant concepts of attributed graphs for a better comprehension of the remaining parts. In general, a major contribution in this work is the specification of different context selection models. We categorize them in two main categories: *model-dependent* and *generic* selection schemes. Each category corresponds to a part in this thesis. Further, we also map our contributions onto different stages of the KDD process they are involved as shown in Figure 1.2. In the following, we summarize each of the parts and the contributions of this work.

Part I: Attributed Graphs

As a consequence of combining attributes with the graph structure, novel concepts and use cases arise. Thus, the objective of this part is to introduce these notions, to compare our models with existing context selection schemes and to describe the proposed benchmarks that have been used for the evaluation.

Chapter 2 describes basic notions of attributed graphs as well as novel concepts derived from the context selection of mining attributed graphs. In particular, we present a novel taxonomy of context selection schemes based on the existing ones and the proposed models in this work. Based on this, we first give an overview of traditional techniques for graph clustering and relational data and, then, compare our work with the related work for mining attributed graphs.

Chapter 3 describes several use cases and benchmarks we have designed for the evaluation of outlier detection on attributed graphs. Specifically, we focus on the E-commerce platform *Amazon* as novel application for outlier mining on attributed graphs. Chapter 3 describes the first proposed benchmark, where outliers have been manually labeled by experts in a subgraph of the *Amazon co-purchased network*. To achieve this, we have conducted an user experiment which is described in this chapter. Additionally, we provide other attributed graphs with a ground truth for further evaluation of outlier mining techniques on larger graphs. Finally, we discuss the lessons we have learned during the development and implementation of such benchmarks.

Part II: Model-dependent Context Selection

The traditional research areas of graph clustering and outlier mining have already proposed several well-established models for both outlier detection or graph clustering. In order to leverage these research efforts, this part introduces generalizations of well-known traditional models to attributed graphs. Overall, the generalization of these models requires (1) the combination of both information resources (attributes and graph structure), (2) a context selection for the awareness of irrelevant attributes and (3) efficiency w.r.t. both graph size and number of attributes. Regarding the selection scheme, the techniques in this part focus on a single projection of the attributes in order to preserve the efficiency of the traditional models to be generalized. In the following chapters, we introduce approaches for robust clustering and outlier ranking on attributed graphs.

Graph clustering based on *modularity maximization* is a well-established research area with plethora of efficient algorithms. However, modularity does not consider the attribute information. Therefore, Chapter 4 proposes a parameter-free modularity-driven clustering for attributed graphs. In contrast to existing approaches, we focus on the robustness of graph clustering w.r.t. both irrelevant attributes and outliers. To achieve

this, we first introduce *attribute compactness* that locally quantifies the relevance of each dimension within the clusters. This attribute information considers only the relevant attributes for assessing the similarity of the attribute values. Furthermore, it allows to detect an outlier with highly deviating attribute values when it is added to a cluster. Finally, we propose an *attribute-aware modularity* that combines modularity with this attribute information. Since we prove the NP-hardness of *maximizing attribute-aware modularity (MAM)*, we propose several heuristics which are generalizations of well-established algorithms for *modularity maximization*. Although a multitude of strategies has been proposed for modularity maximization, they cannot be applied for the maximization of *attribute-aware modularity* in a straightforward way. A core challenge is to preserve the efficiency by the incremental calculation of the attribute information. This is a first essential requirement when generalizing existing heuristics for modularity maximization. However, fulfilling this condition entails new problems in the algorithmic design. Incremental algorithms for the attribute information require precise and stable calculations. Consequently, we ensure a numerically stable and incremental calculation of *attribute-aware modularity* to ensure efficiency w.r.t. both quality and runtimes.

In contrast to the previous approach for clustering attributed graphs, Chapter 5 describes a technique for outlier detection. In particular, we focus on outlier ranking algorithms that sort the objects according to their degree of deviation. This ranking eases a user-driven exploration of outliers, by looking at the most deviating objects first. For each object, we compare its attribute values to those of its neighborhood. However, only the relevant attributes showing dependencies have to be considered for calculating this outlierness. Based on this idea, *ConOut* selects a local context consisting of both a set of relevant attributes and the graph neighborhood of each node. The relevance of the attributes is measured by a statistical test that compares the local and the global distributions of each attribute. This selection enables a high contrast between outliers and their local context for outlier ranking. Regarding the algorithmic design, we exploit structural properties in order to provide an efficient algorithm. This enables to analyze efficiently the local graph neighborhood for each node. Finally, we propose a novel ranking function which does not only calculate the degree of deviation w.r.t. the attribute values, but also considers the connections within the graph neighborhood of each node.

Part III: Generic Context Selection

Existing approaches for mining attributed graphs and the techniques presented in Part II have focused on the extraction of specific patterns out of the attributed graph (data mining step of the *KDD process* as shown in Figure 1.2). Nevertheless, all these techniques are not able leverage existing research efforts on attributed graphs. For instance, their context selection schemes can not be used for improving existing approaches due to

their dependencies on the underlying model. In contrast, this part introduces novel approaches which are more generic and, thus, can be applied as pre-processing and post-processing steps in the *KDD process*. Flexibility is their most relevant property, but such an abstraction poses simultaneously a challenge in their design. In this part, we also focus on approaches that enable multiple views of the attributed graph for the extraction of multiple contexts.

Existing full dimensional approaches for attributed graphs assume that the homophily effect is fulfilled for all attributes. However, these techniques fail if this assumption does not hold for the entire graph and the full attribute space. Chapter 6 presents *ConSub*. It is a pre-processing step that selects several subspaces in which the required dependency between graph and attribute information becomes prevalent. So, our approach enhances existing full dimensional approaches by (1) ensuring the homophily assumption in a subset of attributes and (2) providing multiple views of the data. First, we introduce the novel notion of *congruent subspaces* that captures the dependency between node attributes and the graph structure. We develop a statistical selection of congruent subspaces, and define a general measure that assesses the degree of congruence. Only subspaces that pass our statistical test are used. The selection of all possible subspaces dependent with the graph structure requires the analysis of an exponential number of subspaces. With *ConSub*, we solve this challenge by introducing an heuristic for the selection of congruent subspaces. In this chapter, we focus on community outlier detection as an exemplary task relying on the dependency assumption, but this pre-processing step can be applied for different mining tasks on attributed graphs.

ConSub selects subsets of attributes showing dependencies with the entire graph (*global dependency*). In contrast to this, the goal of subspace clustering approaches on attributed graphs is to select locally for a subgraph different subspaces (*local dependencies*). In Chapter 7, we exploit the results in this research area by proposing a post-processing step for outlier ranking. The goal of our approach is to detect outliers hidden in local subspaces of attributed graphs. To achieve this, *GOutRank* ranks the outliers according to the multiple views extracted from these subspace clustering techniques. We propose different ranking functions that extract several general features from the subspace clustering results. The main property of these functions is that they do not depend on the underlying cluster model. So, we ensure a flexible approach where different subspace clustering approaches can be applied. Finally, we enrich our proposed ranking functions with the information of centrality measures that are well-known node properties in the graph.

Part IV: Summary

Chapter 8 summarizes the contributions of this thesis. Finally, we conclude this thesis by describing the open research questions regarding the development of approaches for mining attributed graphs. In particular, the use case of electronic platforms arises

more challenges such as missing attribute values, dynamic attributed graphs or node attributes with mixed attribute types. Due to the huge volume of data, the analysis of massive networks does not only require efficient algorithms, but also distributed ones. We discuss all these challenges in the last part of this thesis.

Part I.

Attributed Graphs

2. Contexts for Attributed Graphs

In this chapter, we first introduce the most relevant concepts of attributed graphs used later in other chapters. This thesis proposes different context selection schemes depending on the data mining perspective, they are designed for, or the stage of the KDD process, they are involved. Therefore, we bring them together in a novel taxonomy in this chapter. Based on this, we finally review the variety of context selection schemes that has been proposed for traditional data mining and, finally, discuss the existing gap in this area for attributed graphs.

2.1. Preliminaries

Attributed graphs combine the information of two sources: the graph structure and relational data. In the following, we introduce the basic notions relevant for this thesis according to each source of information. Then, we formally define a database consisting of an attributed graph.

Plain Graphs A large number of different domains such as social networks, biological networks, bibliographic networks or auction networks require graphs for their representation [EK10] . In general, *graphs* are the mathematical formalization of a *network* that describes a real world phenomena (e.g., social networks). Formally, a graph is a data structure that consists of a tuple $G = (V, E)$, where V is a finite set of $nodes$ and $E \subseteq V \times V$ is the set of pairs of nodes, called $edges$. Each of these pairs represents a connection between two nodes in the graph (e.g., a friendship in a social network). We focus on $undirected$ graphs where an edge is an unordered pair of two nodes $\{u, v\} \in E$. In this thesis, we mainly consider that graphs are $unweighted$ and do not contain $self - loops$, i.e., an edge incident to the same node $\{v, v\} \in E$. A subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is a graph where $V' \subseteq V$ and $E' \subseteq E$. A $walk$ is a sequence of nodes $v_0 \cdots v_k$ of G such that $\forall_i \{v_i, v_{i+1}\} \in E$. A $path$ is a walk where the nodes are not repeated. A graph G is $connected$ if for every $u, v \in V$ there is a path in G from u to v. This means that each node is reachable from any other node. Otherwise, the graph is $disconnected$. A multitude of definitions for graph anomalies have been proposed [Cha04, NC03, EH07, ATK14]. However, we consider only single nodes as outliers and we do not consider anomalous edges, irregular subgraphs, and other suspicious structural anomalies.

Relational Data A plethora of databases store information based on the relational model where each object is described by attribute values. For instance, a product in a E-commerce platform is characterized by attributes such as *price* or *number of ratings*. In particular, we focus on numerical attribute values in this thesis. Thus, each node v is formally described by a vector $\vec{v} = (x_1, ..., x_d) \in \mathbb{R}^d$. The set of attributes is named $D = \{d_1, \ldots, d_d\}$, and its cardinality is $d = |D|$. The terms *attribute* and *dimension* are equivalent. A *subspace* S is a non empty subset of attributes $S \subseteq D$. In this thesis, we use the concept of subspace for approaches based on the multi-view paradigm. Otherwise, we employ the term *projection* for referring to a subset of attributes. When all attributes are considered in a subspace $S = D$, we call this *full dimensional space*. Outliers are objects with highly deviating attribute values. Outlier rankings score each object according to the *degree of deviation* measured by a function $score : DB \rightarrow \mathbb{R}$. This score provides a real-valued measure of the objects' *outlierness*. Depending on the ranking function, outliers have low scores, and regular objects have high scores or vice versa. In each Chapter, we will specify this when introducing the ranking function.

Having introduced the most relevant concepts from both data types, we define formally an attributed graph as connected undirected graph $G = (V, E, \alpha)$ where

Definition 2.1:

Attributed Graph
An attributed graph consists of an *undirected* and *connected* graph $G = (V, E, \alpha)$ and its attribute information D where:

- Each node v is a graph vertex $v \in V$ and connected by edges $\{v, p\} \in E$ to other nodes $p \in V$ in the graph structure.
- Each node v is described by a vector $\vec{v} = (x_1, ..., x_d) \in \mathbb{R}^d$ where the attributes are named $D = \{d_1, \ldots, d_d\}$.
- The function $\alpha : V \longrightarrow \mathbb{R}^d$ maps each node v to a vector $\vec{v}$.

In this thesis, we denote $\alpha_i(v)$ as the projection of vector $\vec{v}$ on dimension d_i and $\alpha_S(v)$ as the projection of vector $\vec{v}$ on subspace S. Overall in this thesis, the concept of database DB refers to the entire graph $G = (V, E, \alpha)$ and its size is determined by the number of nodes in the graph $|DB| = |V|$. Additionally, we use the terms *vertex*, *node* and *object* interchangeably.

One major challenge in this thesis is the robustness w.r.t. irrelevant attributes and outliers. In particular, we focus on outlier nodes that appear in combination of the graph structure and the attributes as defined in [GLF+10] (cf. Figure 1.1). However, we extend the definition by considering that outliers are hidden in a subspace S of the attributes. Thus, we focus on outliers that are embedded within a subgraph G' and show highly deviating attribute values in a subspace S which is relevant for G':

Definition 2.2:

Outlier
An outlier is a node $v \in G$ where:

- node v belongs to a **connected subgraph** $v \in G'$
- subspace $S \subseteq D$ is **relevant** for G'
- node v has highly attribute values in $S \subseteq D$

Given Definition 2.2, one main research question, tackled in this thesis, is how we can formally model the relevance of the attributes according to a subgraph G'. We call the tuple (G', S) as the *context* of an attributed graph and it can be formalized in multiple ways as explained in the following section.

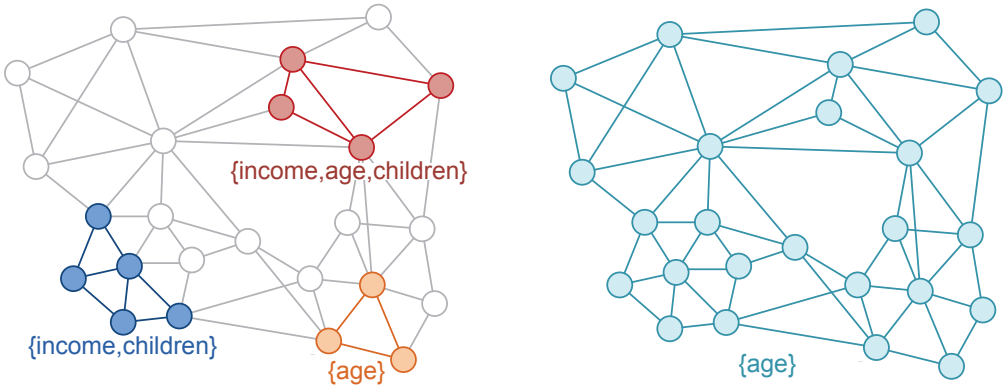
2.2. Taxonomy for Context Selection

Some events (patterns) are deeply hidden in the data and they only appear under some circumstances, which we call *contexts*. For instance, let us consider the database of an university with the information of all students and their activities. A female student may be peculiar if we analyze the students of the computer science faculty. However, this same student is not distinctive when analyzing the students attending language courses in the university. These circumstances (e.g., analyzing only students of computer science) induce different meaningful patterns and, thus, data mining techniques have to be aware of them. To achieve this, they formalize these *contexts*. Contexts have been already defined in multiple ways for traditional data structures such relational data [SWJR07, WD09, KMB12, MSS11] or graph data [WD09, AMF10].

Regarding relational data, existing approaches for context selection have mainly focused on contexts which are determined by a subset of attributes [KMB12, MSS11, KKZ09]. We call this a context specification based on the *attribute perspective*. All these approaches have shown improvements w.r.t. traditional approaches without a context selection since the database contains many irrelevant attributes hindering the detection of meaningful patterns. This is particularly true for attributed graphs since not all the attributes show dependencies with the graph structure as shown in [New03]. Therefore, we consider the attribute selection as one requirement for the context formalization for attributed graphs. The formalization of the attribute perspective can be done in two different ways. *Single-view* context schemes only consider a single projection of the attributes. On the other hand, different attribute combinations (subspaces) may lead to different views of the data (multiple contexts). In this thesis, we call this paradigm *multiple-views*. Section 2.4 introduces in more detail each of these context selection schemes according to the attribute perspective.

If the graph structure is combined with the attributes, it also enables different specifications for context selection. Following our previous example of the female student, different patterns can be extracted if we only consider her close friends instead of using the entire social network of the university. Therefore, some approaches for graph mining focus on the analysis of local subgraphs (local context) while other techniques aim to extract global patterns out of the networks (global context). We call this type of context formalization *graph perspective* and we discuss the most relevant approaches for this thesis in Section 2.3. Overall, each data type enables to formalize contexts in different ways. They enable to detect hidden patterns that appear under different circumstances.

Mining attributed graphs require to combine the information of both: graph structure and attributes. As a consequence, context formalizations can also be done based on combinations of: the *attribute perspective* and the *graph perspective*. For instance, different group of persons in a social network may be characterized by different attributes as shown in Figure 2.1. Thus, the attribute selection has to be done based on local subgraphs G' where $V' \subset G$. On the other hand, an attribute such as *age* may show dependencies with the entire graph as shown for assortative networks [New03]. This means that all the persons in the entire social network tend to have friends with similar ages. If one aims to extract such knowledge, the attribute selection has to be done globally (considering the entire graph structure) (cf. Figure 2.1). In general, a global context selection is more general since it does not depend on a specific subgraph definition.



(a) *Local Context Selection* (b) *Global Context Selection*

Figure 2.1.: Context Selection Schemes regarding the graph perspective

Given these two perspectives and their different paradigms, we analyze different context selection schemes in this thesis. To do this, we introduce a taxonomy which is depicted in Figure 2.2. According to this, we discuss the existing context selection schemes for attributed graphs and the research gaps this thesis addresses in Section 2.5.

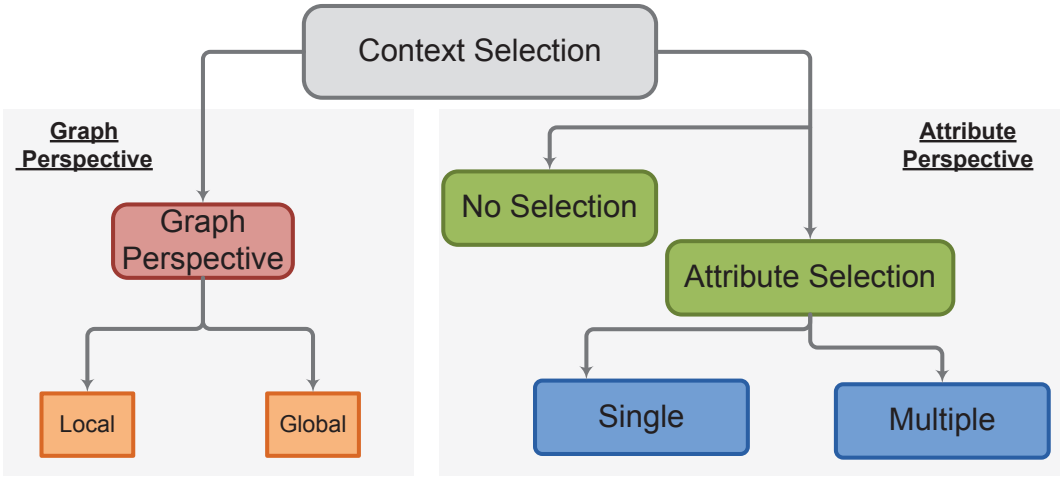


Figure 2.2.: Taxonomy of context selection schemes for attributed graphs

2.3. Graph Perspective for Context Selection

Different graph mining techniques have been proposed for knowledge discovery on real world networks [AW10]. We classify them in two categories: (1) *global* that consider the entire graph and (2) *local* that exploit the information of local subgraphs.

Global The retrieval of global properties (e.g., who is the most influential in a social network) is a relevant task for social network analysis. In the literature, a large number of centrality measures have been proposed in order to find and analyze central nodes [Was94, Fre79]. These properties are not only used to obtain an overview of the network, these global measures can be also used to assess other graph mining tasks such as outlier detection [DKB+12]. In contrast to the analysis of single nodes, graph clustering or community detection algorithms search for the best partition of the entire graph into group of nodes [AW10, For10]. In such a global partition, it is ensured that each node belongs to exactly one graph cluster (community). To achieve this, some algorithms restrict dense subgraphs from sparsely connected graph regions network by optimizing globally a quality measure within the entire graph. Several quality measures have been proposed for graph clustering [For10, AW10, BGW07]. In this thesis, we have specially focused on the enhancement of the quality measure *modularity* [NG04] (cf. Chapter 4). The research area of modularity is a well-established one with theoretical foundations [BDG+08, FB07] and an extensive number of empirical evaluations [RN11, BGW07]. It has also been generalized for weighted and directed graphs [New04a, LN08]. Additionally, modularity enables a parameter-free graph clustering and an incremental calculation of the clustering quality. This last property has allowed the development of a plethora of efficient algorithms for its maximization [New04b, BGLL08, SM13, RN11]. However, all these techniques are not aware of anomalies in the graph structure. To overcome this, a clustering algorithm presents a parameter-free approach which ex-

tracts the cluster structure as well as anomalous edges [Cha04]. Finally, outliers in a graph can be also categorized by the underlying graph type since some approaches have been designed only for bipartite graphs [SQCF05, TL11].

Local In contrast to the previous techniques, some graph mining algorithms analyze locally the graph structure. In particular, this local analysis has shown to improve global approaches for outlier detection when outliers are hidden locally w.r.t. the entire graph structure [AMF10]. The approach presented in [AMF10] extracts features of the graph structure such as the number of neighbors of a node and, then, apply outlier mining techniques on this vector data to compute the outlierness score of each node. Considering the local density of a node does not only enables to detect outliers, but also hubs in the graph structure as byproduct of a clustering algorithm [XYFS07, SHH$^+$10]. To achieve this, the structural similarity of two nodes is analyzed based on a measure that considers the common neighbors of the node neighborhood. So, it is able to detect sets of highly connected nodes and to output the residual set of sparsely connected nodes as outliers. In contrast to a global optimization, this technique enables to define a community only considering this structural similarity. Global clustering algorithms require that the entire graph has to be known. However, the huge sizes of real world networks sometimes avoid to know this. As a consequence, local approaches for partitioning the graph consider only local subgraphs [Cla05, ACL06]. Thereby, these techniques enable to be more efficient w.r.t. global partitions algorithms. Finally, some algorithms search for anomalous subgraphs using the minimum description length principle that aims to minimize the number of bits required to encode the subgraph in the graph [NC03, EH07].

Overall, the application of the search schemes (global or local) heavily depends on the user requirements. Global approaches are meaningful when one aims to get a global overview of the data. On the other hand, local approaches provide more details since they detect patterns that are hidden locally in the data. All aforementioned approaches introduce search schemes based only on the graph structure and this causes an information loss for mining attributed graphs. In the following, we explain the possible context selection schemes for vector data before explaining those for attributed graphs.

2.4. Attribute Perspective for Context Selection

In the literature, multiple paradigms have been developed for context selection according to the attribute perspective. In particular, the most relevant ones are: (1) the subspace paradigm which aims to select multiple views of the data by selecting all subset of the attributes showing correlation between them and (2) projected paradigm that does a single selection of the attributes. In the following, we first discuss full dimensional approaches that consider all the attributes and, then, introduce the context selection schemes relevant for this thesis.

No attribute selection For several decades, outlier mining on vector data has been studied [CBK09]. Furthermore, different paradigms have been proposed such as supervised [VW09], deviation-based [RL87], distance-based [KN98] or density-based methods [BKNS00]. In this work, we mainly focus on density-based outlier ranking, which proposes scores to measure the deviation of each object w.r.t. the object's local neighborhood. Similar to this, a plethora of clustering approaches has been proposed in the literature [Ber06]. In particular, we focus on clustering techniques which are aware of outliers such as the technique proposed in [EKSX96]. However, all these approaches consider the full dimensional space (all the attributes). A core problem is that some irrelevant attributes will scatter the full attribute space [BGRS99], and outlier or clustering is hindered. As a consequence, more recent developments include a context selection scheme considering the attribute selection [AY01, KKZ09].

Single The goal of this paradigm is to select only a single subset of attributes by removing all attributes which are not relevant. Well-established approaches, such as principal components analysis [Jol86], are used to reduce the data space to a single projection as a pre-processing step for a large number of data mining tasks such as clustering (cf. Figure 1.2). In contrast to these pre-processing steps, projected clustering aims to select the single projection simultaneously while they extract the clusters [MZK$^+$09, CBK09]. So, they do not only output each cluster, but they also provide the set of relevant attributes characterizing each of them. A representative of this paradigm is the partitioning algorithm PROCLUS [AWY$^+$99] based on the k-medoids method. However, it requires the number of clusters as parameter. To solve this, an approach proposes a agglomerative hierarchical clustering that calculates the relevance of each dimension to a cluster and does not require any parameter setting [YCN04]. In particular, the outlier-awareness of this approach is also an additional property since these exceptional objects deteriorate the clustering quality [YCN04]. This is one relevant issue that we have considered for the development of our clustering techniques on attributed graphs. In general, single view approaches propose efficient processing schemes w.r.t. the dimensionality of the database [MZK$^+$09, MGAS09].

Multiple In contrast to the a single projection of attributes, multiple-views approaches select several subsets of attributes. This allows the retrieval of multiple perspectives from the database. For instance in a customer database, a client may be clustered in the group of young and rich people considering the attributes income and age, but this same customer can be an outlier in another view regarding his weight and age. In general, subspace techniques provide a set of subspaces where different patterns (e.g., clusters or outliers) can be found in distinct projections of the attributes [MZK$^+$09, CBK09]. Hence, all these techniques provide more information from the data in contrast to single view approaches. So, they have enhanced existing techniques of outlier mining since they are able to detect hidden outliers in the subset of attributes. We categorize the selection of subspaces into two main categories of approaches: (1) *model-dependent* and (2) *generic selection*. Several subspace selection schemes for clustering high dimensional highly depend on the underlying cluster model [CFZ99, KKK04] or outlier

model [AY01, MSS11]. This loss of generality in the selection does not enable to apply these schemes for other data mining tasks. To avoid this, general selection schemes have been recently proposed [KMB12, NMV+13, NMB13]. Regarding outlier mining, general ranking schemes have also been proposed that exploit the results of subspace clustering by proposing a post-processing step [MAIS+12]. Overall, the main concern of subspace selection approaches is their time complexity, which is still an open challenge for large databases.

Overall, all these approaches neglect the graph structure, since all these context selection schemes are only designed for relational data.

2.5. Context Selection for Attributed Graphs

To combine graph structure with the attribute information induces multiple context selection schemes since different perspectives can be joined as shown in Figure 2.2. Despite of this, most of the existing work for attributed graphs neglects the attribute perspective [ZCY09, ZCY10, ZCY10, XKW+12, HZZL02, STM07, Vie12, GLF+10]. To solve this, several techniques have proposed context selection schemes considering the *attribute perspective* and local graph perspective [GFBS10, GBS11, GBFS13, GFRS13, ATMF12, PAISM14, YML13, YJCZ09].

Table 2.1 summarizes the related work on attributed graphs grouped by the context selection schemes presented in our taxonomy (cf. Figure 2.2). Although a variety of model-dependent schemes have been proposed for community detection, context selection schemes for outlier mining are still an open research question. In the following, we discuss existing approaches according to the attribute perspective they propose in their context specifications.

		No Selection	**Attribute Selection**	
			Single View	Multiple View
Local	community	✗	[GFRS13, ATMF12, YJCZ09] **Chapter 4**	[MCRE09, GFBS10, GBS11, GBFS13, PAISM14, YML13]
	outlier	✗	**Chapter 5, Chapter 4**	[PAISM14], **Chapter 7**
Global	community	[ZCY09, ZCY10], [STM07] [Vie12, XKW+12], [HZZL02], [GLF+10]	[TL12]	**Chapter 6**
	outlier	[GLF+10]	[TL12]	**Chapter 6**

Table 2.1.: Overview of related work on attributed graphs grouped by their context selection schemes. It also shows the categorization of the schemes presented in this thesis

No Attribute selection

Traditional approaches for either relational or graph data neglect one information source. Thus, multiple graph clustering techniques have been proposed that combine both resources of information (graph structure and all attributes). All these approaches search for a global partition of the attributed graph, and, then, the context is defined by the combination of the entire graph with all attributes. The core idea presented in [ZCY09] is to convert attribute values into graph nodes for attributed graph clustering Following this idea, an efficient extension of this approach is proposed in [ZCY10] that provides an incremental calculation. In contrast to these distance-based techniques, the work presented in [XKW+12] proposes a Bayesian probabilistic model which defines a jointly probability distribution over the space of all attributed graphs and all possible partitions. Its goal is to find clusters where the edge connections and the attribute values follow a common distribution. However, all these approaches are limited to categorical attribute values.

For numerical attribute values, a novel distance function for combining the information of biological networks with the gene expression data is introduced in [HZZL02]. Regarding clustering techniques based on the network modularity, an extension to spectral clustering incorporates the attribute values as edge weights into the clustering process [STM07]. Instead of using edge weights for including the attribute information, the work in [Vie12] proposes an enhanced objective function which combines the similarity of all attributes with modularity. All previously explained approaches are not aware of nodes that are embedded within the graph clusters and have highly deviating attribute values. Thus, a non-parameter-free approach for community outlier detection [GLF+10] focuses on outlier nodes that deviate from a community of similar nodes w.r.t. both the graph structure and node attributes. To achieve this, this approach presents an outlier-aware clustering for attributed graphs.

In conclusion, all these techniques for attributed graphs exploit the correlation between the graph structure and all node attributes for the enhancement of traditional techniques. Specifically, they assume that all the node attributes are correlated with the entire graph structure. However, this assumption does not always hold as some attributes do not show dependencies as shown in [New03]. Therefore, to consider all the attributes leads to a deterioration of the quality.

Multiple Views

In order to avoid a quality decrease caused by the irrelevant attributes, several local context selection schemes have been introduced [MCRE09, GFBS10, GBS11, GBFS13, PAISM14, YML13]. As common property of these context formalizations, they select the relevant attributes for each subgraph locally. Further, an object may belong to

different clusters. Thus, they provide *multiple views* of each object in the attributed graph.

Recently, *multi-view* schemes for binary attributes has been recently proposed in [YML13]. This work focuses on the detection of overlapping communities. Each community is characterized by a vector where the attributes are weighted according to their relevance. Each object can belong to different communities and this is specified by a community membership vector that contains the affiliation weight. The attribute weights, that indicate their relevance to each cluster, is determined automatically by the algorithm. Instead of this unsupervised selection, the work in [PAISM14] proposes a user-driven selection of the relevant attributes. With these focused attributes, the approach searches for subgraphs structurally dense where the node attributes are similar to those the user has previously defined. Its outlier-awareness is another interesting property of this algorithm, but we focus on unsupervised techniques in this thesis.

Considering numerical node attributes, the work in [MCRE09] combines the concept of mining cohesive subgraphs with the search for subspace clusters in the attribute space. Its goal is to detect all subgraphs fulfilling a specific edge density with their relevant subspace. The combination of different subgraphs with different subspaces results in a huge number of results (subspace clusters). However, this approach does not provide a pruning strategy in order to avoid this. In contrast, the approach proposed in [GFBS10] introduces a redundancy model. Its cluster definition is based on quasi-cliques w.r.t. the graph structure. Regarding the attribute values, nodes within a grid-cell are considered similar following the basic idea of grid-based approaches [KKZ09]. In general, this cluster model is very restrictive. Thus, a more flexible cluster definition is presented in [GBS11]. Its core idea is based on density-based clustering approaches for vector data [KKZ09]. Overall, the time complexity of all these approaches is extremely high. Therefore, the work in [GBFS13] introduces a more efficient algorithm following the cluster model presented in [GFBS10].

In general, the selection of multiple contexts heavily depends on the underlying local subgraph definition. This means that all these approaches provide *model-dependent* context selection schemes. Thus, they can be considered specific solutions to the problem of subspace selection, but they lack generality and are not designed as preprocessing step for other graph mining models (cf. Figure 1.2). In this thesis, we solve this by proposing a global context selection scheme that ensures the dependencies between the attributes and the entire graph structure (cf. Chapter 6).

Single View

Although multi-view approaches are able to extract more information out of the data, their major drawback is their time complexity. Regarding the database dimensionality, the number of possible subspaces is exponential. To avoid this, few techniques

based on the paradigm of projected clustering have been proposed for attributed graphs [ATMF12, GFRS13]. Specifically, they have focused on a local projection of the attributes (single-view context selection). A parameter-free technique proposes to use the idea of compression [ATMF12], but it only considers binary attributes. For numerical node attributes, an approach based on spectral clustering has been also proposed in [GFRS13], but it is not parameter-free and aware of contradicting effects caused by outlier nodes. Similar to the multi-view context selection scheme proposed in [YML13], the work in [YJCZ09] introduces a statistical model where a weight vector is assigned to each community. However, each object belongs only to a single community. Overall, all these techniques for numerical node attributes require the number of clusters as a parameter which is difficult to set. Regarding generic context specifications, feature selection approaches have recently started to use the graph structure in order to improve their attribute selection [TL12]. Nevertheless, this work does not focus on the selection of attributes showing dependencies with the graph. It has been designed for improving feature selection on vector data. (cf. Table 2.1).

In general, all these context selection schemes have been proposed for community detection as main data mining task. In this thesis, we propose both (1) a context selection scheme for outlier ranking (cf. Chapter 5) and (2) a scheme for a parameter-free and outlier-aware clustering (cf. Chapter 4).

3. Use Cases and Benchmarks

A large number of public available datasets or benchmarks are available for the evaluation of data mining approaches, but they only consists of either a graph structures or relational data. In the case of attributed graphs, it is difficult to find networks with numerical node attributes. This is particularly true if one aims to analyze networks with a large number of them. Recently, some attributed graphs have been public available [GFBS10, GBS11, YML13], but either the number of dimensions is small or the attributes are not numerical. Furthermore, these networks do not provide a ground truth for the evaluation of outlier detection.

Benchmarks are an important issue in order to provide an objective and accurate evaluation of the designed techniques. Thus, we have put some efforts to solve this in this thesis. In this chapter, we describe different use cases and benchmarks we have designed for the evaluation of our approaches. In particular, we focus on the detection of outstanding, rare or suspicious products on electronic platforms. For this use case, we have generate two benchmarks based on (1) an user experiment and (2) collaborative tagging. Finally, we have also considered a communication network where we generated a large number of attributes from the information of emails. All these attributed networks and benchmarks have been already been published in [ISML+13, MISMB13]. They have allowed us to evaluate and compare our approaches based on a quantitative assessment, i.e., calculating a quality measure.

3.1. Co-purchase Network

E-commerce platforms such as *Amazon* and *eBay* have become an important marketplace for both private and professional users. In general, they provide the possibility to sell or buy products in well-known and trusted platforms. Unfortunately, the reputation of these platforms has been deteriorated because of fraudulent or suspicious sellers. For example, fraudulent users are offering fake products or they try to sell overpriced products by exploiting the trust of customers in these platforms. Platform operators have realized these issues and try to help honest users by providing a large number of descriptive attributes for each product, for instance, the price history, product reviews, product ratings, links to similar products in a recommendation network, and

many more. However, this large number of attributes does not allow a manual analysis for the comparison between all related products in order to recognize those that are suspicious. In this thesis, we propose to exploit all information sources of such platforms (co-purchase network together with the product information) for the detection of outstanding, suspicious or rare items.

The *Amazon co-purchase network* has been analyzed in a plethora of case studies for traditional graph clustering [LAH07, CNM04, LKSF10, CZG09]. They have shown that these traditional approaches extract group of products with similar general characteristics (e.g., items of the same category such as books). Furthermore, a benchmark for community detection has been proposed where the ground truth is defined by the attribute *group* of each product. However, all this work has mainly focused on the clustering structure without considering the combination of both sources of information. Combining both sources of information leads to more refined clustering results as we empirically show in Chapter 4. For instance, clustering techniques for attributed graphs are able to retrieve more refined clusters consisting of different type of books. In contrast to this, we aim to design a benchmark for outlier detection that considers both the graph structure and the node attributes. To achieve this, we have conducted an user experiment where labels have been manually assigned to and we have also used the idea of collaborative tagging. Before introducing these benchmarks in detail, we start describing the extraction of our network with numerical node attributes.

	Entire Network	**Giant Component**
Nodes	548552	314824
Edges	1788725	882930

Table 3.1.: Statistics of the Amazon co-purchase network

Data Pre-processing The co-purchased network is public available in [LAH07][1]. Table 3.1 shows some basic statistics of this network. However, it only consists of the graph structure and the provided information of the products consists of few attributes (e.g., *ASIN*, *average rating* or *sales rank*). Therefore, we have extended the product information with product prices extracted from the Amazon website on March 2012. Additionally, we have aggregated several numerical attributes from the reviews. As a result, we obtained an attributed network with 30 numerical node attributes that are described in Table 3.2. Such an attributed graph represents a challenge for the approaches due to both: its high dimensionality and its database size. In particular, we use the largest connected component of this network to show the efficiency of some of our approaches (cf. Chapter 4 and Chapter 6). However, this large network does not contain a ground truth for outlier detection.

[1] `http://snap.stanford.edu/data/amazon-meta.html`

	Description
Price Information	
MinPriceUsedItem	Minimum price offered for an used item
Amazon_price	Price offered for a new item by Amazon
MinPricePrivateSeller	Minimum price offered for a new item by a private seller
Rating Information	
Rating_1_Ratio	Ratio for Rating 1
Rating_2_Ratio	Ratio for Rating 2
Rating_3_Ratio	Ratio for Rating 3
Rating_4_Ratio	Ratio for Rating 4
Rating_5_Ratio	Ratio for Rating 5
Rating_of_review_with_least_votes	Rating of the review with the fewest number of votes
Rating_of_review_with_most_votes	Rating of the review with the most number of votes
Rating_span	Time between the first rating and the last rating
Avg_Rating	Average rating of the product
Top_reviewer_rating	Rating of the user who has written the most number of reviews
Rating_of_most_helpful_rating	Rating of the most helpful review
Rating_of_least_helpful_rating	Rating of the less helpful review
Reviews Information	
Number_of_reviews	Number of reviews
Review_frequency	Frequency of the reviews
Number_of_different_authors	Number of different authors
Min_Helpful	Minimum number of helpful votes of a review
Min_Votes	Minimum number of review votes
Max_Helpful	Maximum number of votes
Avg_Votes	Average number of votes
Avg_Helpful	Average of helpful reviews
Max_Votes	Maximum number of votes
Other Product Information	
Min_Categories_Depth_of_this_Product	Minimum depth of the categories of the product
Max_Categories_Depth_of_this_Product	Maximum depth of the categories of the product
Sales_Rank	Sales rank
No_of_Categories	Number of categories of the product
Product_group	Product group

Table 3.2.: Node attributes of the Amazon co-purchase network

3.2. User Experiment for Benchmarking

One important issue for the development of data mining techniques is their quality assessment. Besides their evaluation on synthetic data, where the ground truth is already known, it is essential to know the quality of the designed models on real world networks. In this thesis, we have conducted an user experiment for obtaining manual labels from experts. So, we have obtained a baseline for an objective assessment of our techniques for outlier mining. The complexity (dimensionality and graph size) of the previously explained co-purchase network makes the design of such an user experiment difficult since the experts are not able to consider the entire database. This may result in errors during the labeling process. Hence, we decided to start with a small subgraph where we were able to have more control during the experiment. In the following, we describe more details of the experiment conducted at our chair.

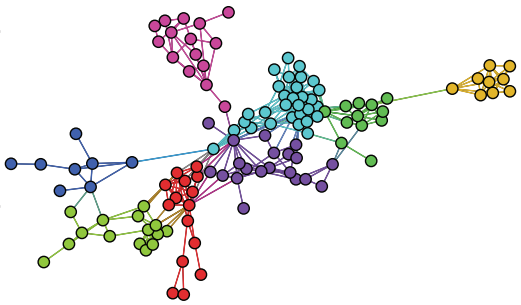
User description In total, 20 girls have participated in our user experiment. The study group included participants of different ages (10 - 16). For such manual analysis of the data, it is essential that users have some knowledge about the data domain. In our experiment, all the students were familiar with the electronic platform of *Amazon*. They also had already watched or bought multiple *Disney* films. These two characteristics ensured that the participants were expert on the data domain. Simultaneously, they objectively searched for patterns in the data since they were not familiar with none of the existing data mining techniques. In other words, they focused on products that they considered rare or outstanding. In order to avoid some random or subjective labeling (e.g., a film they do not like), we briefly explained them the general outlier concept as: *"an object that deviates from other in the attribute values"*. During these clarifications, we did not describe any model in detail since we wanted to avoid a possible bias in our results. Another relevant issue in an user experiment is the motivation of the users for correctly doing the task. To achieve this, we demonstrated them with examples how important data mining on real world applications is (without describing a specific model). Additionally, we had explained them that without these labels future developments for data mining were difficult and, therefore, their help was required. Overall, the girls took seriously this mission since the planned time for this task took more than expected (45 min).

Data Selection and Processing First, we selected a domain of products which is well-known by our participants (young girls). We decided to use the *Disney* products where our users are experts. Then, we extracted a connected subgraph with these products from the large *Amazon* co-purchase network (cf. Section 3.1). Table 3.3 shows the network statistics. Both the size and the dimensionality was too complex for our young students. This complexity may lead to errors during the data analysis. Therefore, we simplified the information given to the participants. First, we chose those attributes which are available on the Amazon website (not aggregated attributes from Table 3.2). With this decision, we wanted to provide information that the user is already familiar

with. Besides the attribute information, to analyze 124 products together may produce unsatisfactory results. In particular, we have focused on contextual outliers in this thesis. These outliers appear locally in the database. Thus, we ensured that students do not label global outliers (e.g., product with the highest price of the database), but they label contextual outliers inside graph clusters and subset of attributes. To achieve this, we clustered the Amazon subgraph. For clustering the graph, we use a modularity based technique [BGLL08].

Nodes	124
Edges	334
Average Clustering Coefficient	0.437
Average degree	5.403

Table 3.3.: Statistics of Disney network



Experiment Sequence

The visualization of the data was an important issue for this user experiment due to our targeted participants. Therefore, we designed a friendly interface similar to *Amazon*. Each student had to analyze a group of products given by the extracted clusters. Figure 3.1 shows a visualization of one product group. For each graph cluster (in total 8 clusters), all products in a graph cluster were shown to the students in the browser. In total, they manually analyzed all 124 products, distributed in 8 clusters, and the time consumed for this task was 45 minutes. Since several products in a cluster may have highly deviating values, we allowed to label 1 or 2 products as outliers for each group. Finally, participants had to fill out a form (cf. Figure 3.2) for each labeled product. In this form, users had to indicate the attributes, where the product has highly deviating values, and write a general explanation why they had labeled the product. In the following, we present the analysis of these forms and describe how we generated the final benchmark.

Results In total, 49 products were labeled as outliers in our user experiment. However, few products were marked multiple times by different students. Only two products (*B00004R99B* and *B00006LPHB*) were labeled by 70% of the users as outliers as shown in Figure 3.3(a). This means that 14 girls considered both products as clear outliers in the database. Besides the agreement between the students, we also analyzed the products according to the provided reasons for being an outlier. Figure 3.3(b) shows the frequency of the attributes where highly deviating attribute values were observed. Product prices and the average rating of each product are the most relevant characteristics used as reason for labeling a product as outlier. In particular, the most labeled products represent items that are overpriced or have low ratings w.r.t. the other co-purchased products. For example, Product with *ASIN B00005T5YC* corresponds to

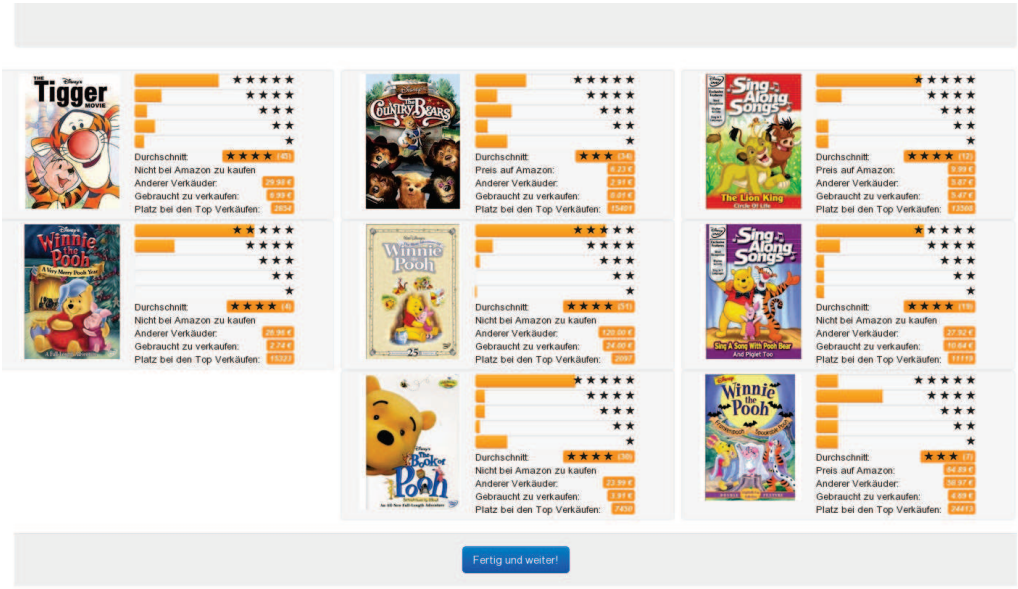


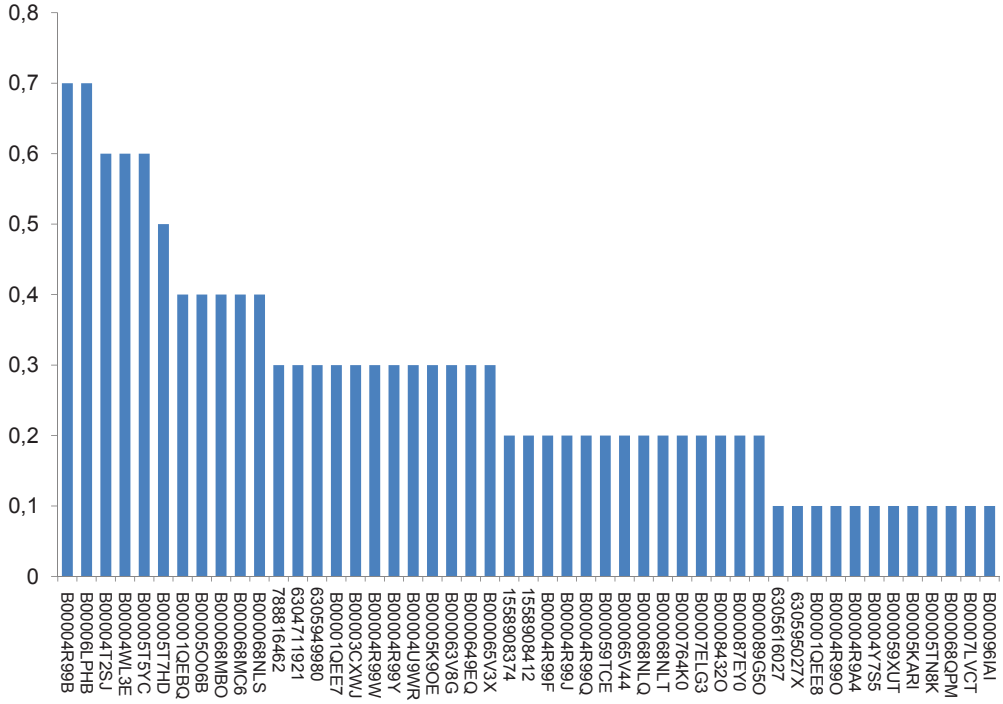
Figure 3.1.: Visualization of one graph cluster during the user experiment

Figure 3.2.: Form which had to be fulfilled by the user for detailed explanations

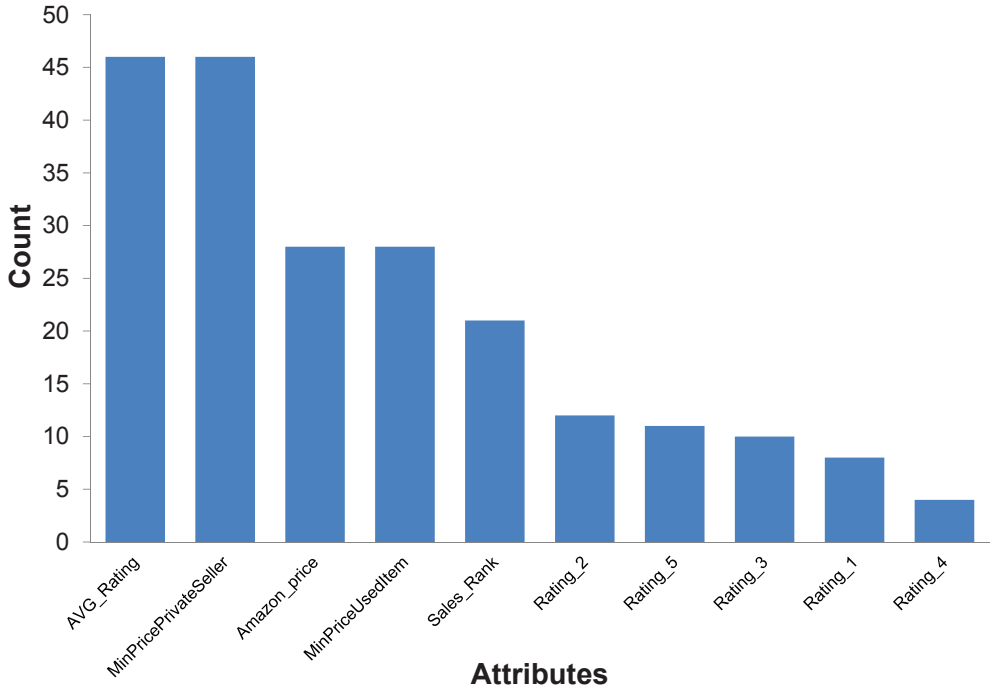
the overpriced film *The Jungle Book (1994)* of *Rudyard Kipling's* hidden in a group of *Read-Along Disney* films. On the other hand, the product with *ASIN B00004T2SJ* was labeled because of their low ratings w.r.t. the other *Pixar* films. Table 3.4 shows some other descriptions of the most frequently products marked as outliers. In addition to this, Figure 3.4 depicts the frequency of each selected attribute for the labeled products. We can observe that the prices of the used items (offered by private sellers) and the average rating are frequently selected by the user as reason for being an outlier. In some products, we can also recognize that a large number of students had a high agreement with the selected attributes (e.g., *B00005T5YC*, *B00004R99B* or *B00006LPHB*). On the other hand, this agreement in the attributes is not present for products like *B00004R99W* or *B000068NLS*. For our benchmark, we finally selected those products, where at least 50% of the students considered the products as outliers (in total 6 outliers were selected).

Product Asin	**Detailed Description**
B00004R99B	*"The other products have a better rating"*
	"The price of the used item is the same as a new one by private sellers"
B00006LPHB	*"Used costs 0.01 €"*
	"The other products are much more expensive"
B00004T2SJ	*"Overall bad ratings compared to others"*
	"Large difference between the ratings"
B00005T5YC	*"Too expensive for being a film"*
	"It costs more than 100 €"
B00004WL3E	*"Only 7 reviews? The other products have more than 100"*
	"The price offered by private seller is more expensive"

Table 3.4.: Some descriptions of the most labeled products in our user experiment



(a) *Ratio of number of labels assigned to the 49 products marked as outlier*



(b) *Frequency of attributes where highly deviating attributes were observed*

Figure 3.3.: Statistics of the user experiment

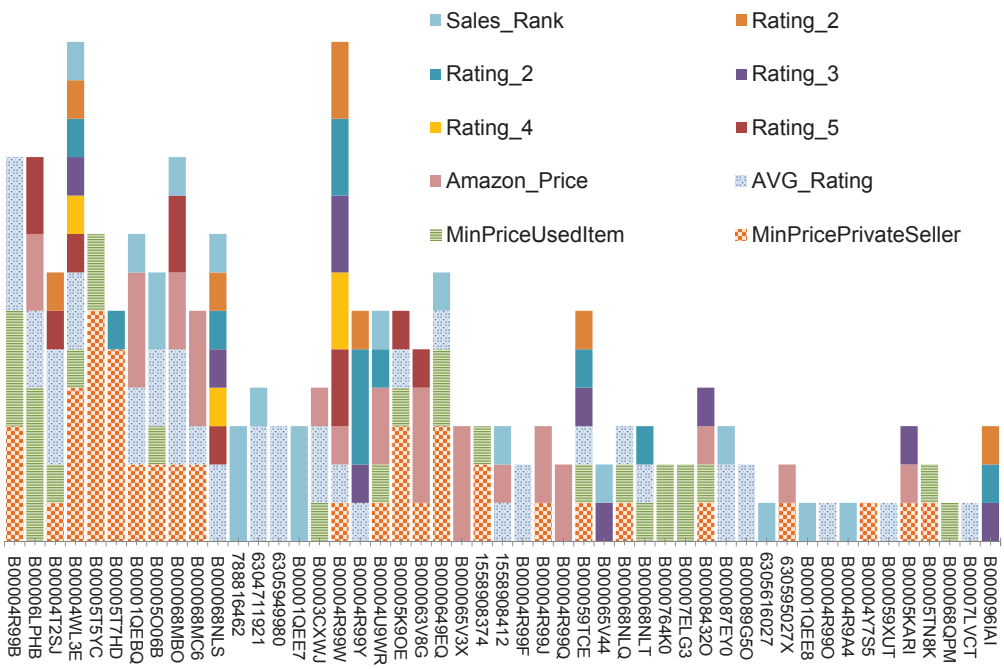


Figure 3.4.: Frequency of selected attributes for each of the products marked as outlier in our user experiment

3.3. Collaborative Tagging for Benchmarking

Three years ago (2012), customers were allowed to label each product with tags in the electronic platform *Amazon*[2]. They were designed for a better organization of the products. Users started to provide tags like *love story* for indicating products where a love story is presented. It has also enabled the user to do recommendations (e.g., the tag: *highly recommended*). In particular, several tags were also used by customers to indicate how novel, suspicious or rare a product is. For instance, peculiar products were tagged with the tag *amazon oddities* or suspicious ones with *wtf* or *fake*. Table 3.5 shows some examples of tags in *Amazon* [3].

However, users tend to tag subjectively products as explained in [SVR09]. For instance, some users assign a tag based on a personal disagreement with the product. They also create new tags before ensuring a similar one exists. In other words, tags in these platforms are typically very noisy. As one solution for this problem, the number of contributors, .i.e., the number of customers using the same tag is provided as shown in Table 3.5. In addition to this, the tags for each product are shown in conjunction with

[2]Currently, the tags feature of *Amazon* has been discontinued

[3]`http://www.amazon.com/gp/tagging/cloud/ref=tag_cld_cl_icld_sm?ie=UTF8&length=1000`

the number of users that have used it for tagging the product. For instance, Product with *ASIN 0870334336*[4] has been labeled with the tag *amazon oddities* by 48 persons and with the tag *wtf* by 21 persons. In the following, we describe the creation of our benchmark based on this collaborative tagging.

	Tag Name	**Contributors**
Peculiar Items		
	amazon oddities	3089
	wtf	3451
High prices or rare sales rank		
	overpriced	4534
	amazon fail	3310
	kindle price too high	4423
Outstanding Products		
	highly recommended	8165
	best cancelled tv shows	8761

Table 3.5.: Tags Examples and the number of persons, that used the tag (contributors), in the electronic platform Amazon

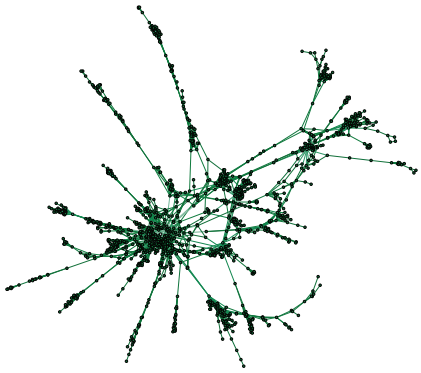
Data Pre-processing First, we extracted all existing tags for each product in the *Amazon* co-purchase network (cf. Section 3.1). Then, we filtered the products that contained tags interesting for outlier mining and analyzed our co-purchase network considering the labeled products. In order to avoid the problem of noisy tagging, we focused on tags where both the number of contributors and the number of persons labeling the product with the same tag were high. For the entire network, few products have a high number of tags while a extremely large number of products do not contain any tag. During this data analysis, we discovered that a large number of books were labeled with the tag *amazon fail* which has a high number of contributors. Therefore, we extracted a subgraph consisting of such books. Table 3.6 shows the basic statistics of the *Books* network. As attributes, we included the node attributes described in Table 3.2.

Ground Truth We use the popular tag *amazon fail* as outlier ground-truth. This tag has been used for few years to let users express their disagreement with the sales ranks. In particular, some users had the opinion that ratings and sales rank from some books were misplaced by *Amazon*. So, users started to label these books with the *amazon fail* tag in order to show disagreement with the manipulation of these products [5]. For the generation of our benchmark, we labeled as outlier a product when it was labeled as *amazon fail* at least by 20 users.

[4] `http://www.amazon.com/dp/0870334336`
[5] `http://news.bbc.co.uk/2/hi/technology/8000401.stm`

Nodes	1468
Edges	3695
Avg. Clustering Coefficient	0.483
Avg. degree	5.212

Table 3.6.: Statistics of Books network

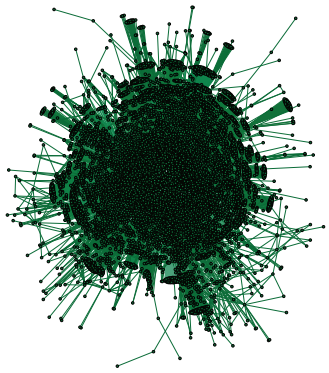


3.4. Communication Network

In contrast to the previously explained benchmarks based on the electronic platform *Amazon*, communication networks are also an important domain where outlier mining on attributed graph can be applied. For this purpose, we generated an attributed network from the large set of email messages of the Enron corporation [6]. This data was made public available during a legal investigation. Although an *Enron network* is public available[7], it has neither attributes nor a given truth for outlier mining. Thus, we have preprocessed the email dataset in order to obtain a high dimensional attributed graph with a ground truth. In the following, we describe in detail the generation of this network and its attributes.

Nodes	13 533
Edges	176987
Avg. Clustering Coefficient	0.321
Avg. degree	26.156

Table 3.7.: Enron network



Data Pre-processing First, we generate the network (graph structure) from the e-mail addresses, that have sent at least one email and represent the nodes in the graph. Two nodes are connected by an edge if one address has sent at least one email to the another one. In particular, we extracted the largest connected component of this extracted communication network. Table 3.7 shows its basic statistics. Besides the graph structure, we have focused on the extraction of a large number of attributes out of the email information since the goal of our techniques is to handle high dimensional attributed graphs.

[6] `http://www.cs.cmu.edu/~enron/`
[7] `http://snap.stanford.edu/data/email-Enron.html`

	Description
Recipient and Sender Information	
EnronMailsTo	Ratio of recipients belonging to Enron ($\frac{\#TO_{Enron}}{\#TO_{total}}$)
OtherMailsTo	Ratio of recipients not belonging to Enron ($\frac{\#TO_{notEnron}}{\#TO_{total}}$)
AverageNumberTo	Average number of recipients in the e-mails
EnronMailsCc	Ratio of carbon copies to recipients belonging to Enron ($\frac{\#CC_{Enron}}{\#CC_{Total}}$)
OtherMailsCc	Ratio of carbon copies to recipients not belonging to Enron ($\frac{\#CC_{notEnron}}{\#CC_{Total}}$)
AverageNumberCc	Average number of carbon copies
EnronMailsBcc	Ratio of blind carbon copies to recipients belonging to Enron ($\frac{\#Bcc_{Enron}}{\#Bcc_{Total}}$)
OtherMailsBcc	Ratio of blind carbon copies to recipients not belonging to Enron ($\frac{\#BCC_{notEnron}}{\#BCC_{Total}}$)
AverageNumberBcc	Average number of blind carbon copies
Subject Information	
AverageSubjectLength	Average length of the Subjects
AverageDifferentSymbolsSubject	Average number of different symbols in a subject
Content Information	
MimeVersionsCount	Number of the differents Mime-Versions
DifferentCosCount	Number of different Content-Types
DifferentCharsetsCount	Number different charsets
DifferentEncodingsCount	Number different encodings
AverageContentLength	Average number of words in the content of an e-mail
AverageDifferentSymbolsContent	Average number of different symbols in the content of a message
AverageContentForwardingCount	Average content length by forwarding an e-mail
AverageContentReplyCount	Average content length by replying a e-mail
Frequency Information	
AverageRangeBetween2Mails	Average time range between 2 e-mails (ms)

Table 3.8.: Node attributes of the Enron communication network

In total, each node contains 20 attributes describing aggregated information about the e-mail address such as its average content length, the average number of recipients or the range of time between two e-mails. Table 3.8 contains all aggregated attributes and their description.

Ground Truth This Email dataset has been frequently used as benchmark for the evaluation of spam detection techniques [MAP06]. Although the goal of this thesis is not to develop specific spam detection techniques, we have decided to use this ground truth as baseline for the evaluation of some of our outlier mining techniques. Thus, we marked spammers as outliers as they may represent an example of possible *community outlier*. The list of spam email addresses was token from the public the public available benchmark in [MAP06].

3.5. Lessons Learned

Overall, the design and implementation of benchmarks for outlier mining represents a major challenge. A core problem in the design of benchmarks for outlier detection relies on the difficulty to determine when an object is an outlier. This is particularly true for contextual outliers since the concept of an object with highly attribute values or

its context can be interpreted differently by different persons. For instance, an user may consider an object an outlier in a context while another person may think the context is not relevant and, thus, the object is not an outlier. In particular, we observed these different disagreements in our user experiment. Few products were clearly labeled as outliers with the same reasons.

In this Chapter, we have presented different type of benchmarks for the evaluation of outlier mining techniques. First, we have conducted an user experiment where outliers have been manually labeled in a small dataset. Two important issues were considered in the design of our user experiment: (1) participants knew deeply the domain of the data and (2) they were not familiar outlier mining models in order to avoid a possible bias when labeling. Furthermore, the users showed a good motivation that reduced a simple random labeling of the objects. We asked them for the reasons they had labeled the products and observed that some students have tried to realize a good work. Another important requirement in our user experiment was to simplify the task for our participants. Although our previous data pre-processing (clustering of the products) avoids the labeling of global outliers and possible errors due to the database size, we have introduced some bias to the results which should be avoided for future developments. It is still an open research question how to select and show the data to the users for ensuring a good analysis and labeling.

For the generation of our second benchmark, we have exploited the idea of collaborative tagging. We have used the tags the users have assigned to products in the electronic platform. The large number of existing user tags requires a previous analysis of them in order to determine which tags represent outliers. Furthermore, only relevant tags (e.g., high number of contributors) has to be considered for benchmarking since this tagging is strong subjective. For instance, some labels were assigned when an user does not like an author. In contrast to our user experiment, all products have not been considered for the labeling. This leads to a none uniform distribution of the tags. In other words, a small percentage of the products of our co-purchase network has a tag while other products have no tags. This happens because the product is not popular or unknown, but it may be also an outlier. Overall, this collaborative tagging may be a good idea for future design of benchmarks in other domains, but the distribution of the labels and noise should be more controlled.

Finally, we have proposed a benchmark for a communication network based on a spam detection benchmark. Although it has been objectively created, i.e., without any bias given by the settings of an user experiment, it is important to remark that nodes with highly deviating values are not necessarily only spammers. For instance, we observed that an email address, which is strong connected to other addresses, may write also frequently a large number of long mails. It fulfills the definition of community outlier (cf. Definition 2.2), but this address was not a spammer.

Overall, the creation of benchmarks is a challenging and time consuming issue, but it is necessary for a quality assessment of the techniques. It is important not only to rely

on quality measures since benchmarks may be also somehow controversial. Therefore, a combination between case studies and quality measures on real world networks is a good trade-off.

Part II.

Model-dependent Context Selection

4. Modularity-driven clustering

Clustering methods based on modularity are well-established and widely used for graph data. However, today's applications store additional attribute information for each node in the graph. This attribute information may even be contradicting with the graph structure, which raises a major challenge for the simultaneous mining of both information sources. For clustering attributed graphs it is essential to be aware of such contradicting effects caused by irrelevant attributes and highly deviating attribute values of outlier nodes.

In this Chapter[1], we focus on the robustness of graph clustering w.r.t. irrelevant attributes and outliers. We propose a modularity-driven approach for parameter-free clustering of attributed graphs and several efficient algorithms for its computation. The efficiency is achieved by our incremental calculation of attribute information within these modularity-driven algorithms. In our experiments, we evaluate our modularity-driven algorithms w.r.t. the new challenges in attributed graphs and show that they outperform existing approaches on large attributed graphs.

4.1. Motivation

A wide range of applications in industry and sciences use graph clustering for knowledge discovery. Analysis of social networks or gene interactions in biological networks are two examples. Many domains require the extraction of clusters as sets of similar nodes. Different definitions of a graph cluster have been proposed [For10]. The most commonly used notion is to group nodes that are densely connected within a cluster and have sparse connections to other clusters. A quality measure is used to score each cluster based on this model. The overall clustering result is then obtained by optimizing a function (e.g., the sum of scores) over all clusters which consists of the best set of clusters that optimize the overall value of this quality measure.

In particular, *modularity* [NG04] is a well-established quality measure for graph clustering. It enables a parameter-free computation of the graph clusters, and it has been generalized for weighted [New04a] or directed graphs [LN08]. Furthermore, a core property

[1] This chapter is an extension of the published work in the Proceedings of the SIAM International Conference on Data Mining (SDM 2015) [ISMK$^+$ar]

of modularity is that it enables the incremental calculation of the quality of a clustering. Although finding the optimal clustering using modularity is NP-hard [BDG+08], that characteristic has given way to the development of a broad set of efficient greedy strategies for scalable computations [CNM04, BGLL08, RN11, SM13]. Overall, *modularity maximization* is a well established research area that has resulted in scalable processing schemes [BGLL08, RN11, SM13].

However, each object is not only characterized by its relationships to other objects (graph structure), but also by individual properties (node attributes). For example, a social network consists of both: (1) the friendship relationship between persons represented by the graph and (2) the personal information of each person such as *age*, *income* or *gender*. A core challenge with attributed graphs is that they contain a large number of attributes, and not all of them are relevant for each cluster [GFBS10, ATMF12]. Some attributes may have scattered values, or they may contradict the graph structure (e.g., *dissasortative* networks [New03]). A social group *athletes* may be characterized by both its graph connectivity and their *sports activity level*, but the node attribute *income* is not relevant due to the different salaries within this group. This group only forms a cluster w.r.t. the attribute *sports activity level* and the graph structure. If *income* is considered, this reduces the similarity of these nodes. In addition, some nodes may have highly deviating attribute values (i.e., low *sports activity level* of a coach) although they are embedded in a well-connected cluster considering the graph structure. Overall, enhancing modularity with attribute information requires to be robust w.r.t. irrelevant attributes and outliers.

In contrast to existing approaches [GFBS10, ISML+13, GFRS13, GLF+10, Vie12, ZCY10, ZCY09, STM07, ATMF12, YML13, YJCZ09, XKW+12], our modularity-driven approach enables a parameter-free clustering of graphs with numerical node attributes that is robust w.r.t. both irrelevant attributes and outliers. To achieve this, we introduce *attribute compactness* that quantifies the relevance of the attributes within a cluster. With this, we consider only the relevant attributes for assessing the attribute similarity. It allows us to detect outliers with highly deviating attribute values within a cluster. However, taking both attribute compactness and conventional modularity into account for the graph clustering calls for a careful algorithmic design to ensure efficiency. We first prove the NP-hardness of our approach for attributed graphs. We then provide algebraic expressions of *attribute compactness* required for its numerically stable and incremental calculation. With this, we ensure efficiency and accuracy when generalizing well-established concepts of modularity maximization to attributed graphs. We focus on such generalizations in order to leverage well-established and efficient ideas from modularity maximization. Finally, we compare our approach to conventional modularity maximization and state-of-the-art algorithms for clustering attributed graphs. We do not only show an improvement of the results, but also better runtimes due to the incremental calculation. Further, our evaluation with several very different real-world data sets of different graph sizes provides anecdotical evidence that

clustering based on attribute-aware modularity yields somewhat more meaningful results than approaches relying on conventional quality measures.

4.2. Comparison to Related Work

In Table 4.1, we summarize existing clustering techniques grouped by their paradigms. The robustness w.r.t. both outliers and irrelevant attributes and its parameter-freeness are the unique value proposition of our work. Basic approaches in attributed graph clustering consider all attributes [ZCY09, ZCY10, XKW$^+$12, STM07, Vie12, GLF$^+$10]. Although some techniques have considered the local selection of the relevant attributes [YML13, YJCZ09, GFBS10, GBS11, GFRS13], none of these provide a parameter-free approach that is robust w.r.t. outliers. Next, there are proposals for pre-processing steps to select relevant attributes. Feature selection methods [TL12] select a single attribute projection, while subspace search [ISML$^+$13] considers multiple projections that are correlated with the entire graph structure. However, these global selection schemes do not select attributes relevant for each cluster locally.

		Robustness		
Algorithms	**Numerical Attributes**	**Irrelevant Attributes**	**Outlier**	**Parameter-free**
random walks [ZCY09, ZCY10]	✗	✗	✗	✗
MDL principle [ATMF12]	✗	✓	✓	✓
statistical models [YML13, YJCZ09, XKW$^+$12]	✗\✓\✗	✓\✓\✗	✗	✗
subspace selection paradigm [GFBS10, GBS11]	✓	✓	✗	✗
modularity-based [STM07, Vie12]	✓	✗	✗	✗
spectral clustering [GFRS13]	✓	✓	✗	✗
outlier-aware clustering CODA [GLF$^+$10]	✓	✗	✓	✗
this work	✓	✓	✓	✓

Table 4.1.: Related clustering methods for attributed graphs

4.3. Problem Overview

In this Chapter, we extend the graph definition of Definition 2.1 to weighted graphs in order to avoid loss of generality w.r.t. modularity. The graph G has a positive edge weighting $\omega : E \longrightarrow \mathbb{R}_0^+$. If two nodes u and v are not connected by an edge, we set $\omega(\{i,j\}) = 0$. Self-loops are allowed. In the following, we introduce the require notation for this chapter. $\mathcal{C} = \{C_1, \cdots, C_k\}$ is a partitioning of the nodes V into disjoint clusters, and $\mathcal{C}(v)$ is the cluster node v belongs to. The set of all possible clusterings of G is $\mathcal{A}(G)$. $W(E) = \sum_{e \in E} \omega(e)$ is the sum of all edge weights in G. $deg(v) = \sum_{\{u,v\} \in E} \omega(\{u,v\})$ is the total weighted degree of a node v. It can be generalized to a set of vertices $S \subseteq V$: $deg(S) = \sum_{u \in S} deg(u)$.

Let $E_k = \{\{v, w\} \in E : v, w \in C_k\}$ be the set of intra-cluster edges and $W(E_k) = \sum_{e \in E(k)} \omega(e)$ the sum of all edge weights of E_k.

In general, we propose a parameter-free modularity-driven approach to cluster attributed graphs that fulfills the following requirements:

Robustness Basic algorithms exploit the dependencies between the graph G and all attributes D for graph clustering [GLF$^+$10, Vie12, ZCY10, ZCY09, STM07]. However, some attribute information may be contradicting with the graph structure. First, the assumption of homophily [MSLC01] (i.e. connected nodes tend to have similar characteristics) may not be fulfilled for the full attribute space [GFBS10, ATMF12, GFRS13, ISML$^+$13]. Thus, irrelevant attributes not showing dependencies with the graph structure do not have to contribute to the attribute similarity assessment within a cluster. Additionally, contradicting effects between graph structure and attribute information are observed by outlier nodes. Outliers are strong embedded within a well-formed graph cluster with similar attribute values, but they have highly deviating attribute values [GLF$^+$10]. As a consequence, outliers hinder the detection of homogeneous graph clusters with similar attribute values. These objects with highly deviating values are recognizable if the cluster is known previously. Thus, outliers have to be recognized during the clustering process. Overall, the design of a robust measure for the attribute information is challenging since it has to exclude all irrelevant attributes while being robust w.r.t. outliers for the assessment of the nodes similarity. Furthermore, this measure has to enable an efficient calculation. In particular, a modularity-driven approach for clustering attributed graphs requires an incremental computation in order to generalize well-established efficient algorithms for modularity maximization to attributed graphs.

Efficiency We prove that the MAM problem (i.e. considering both graph structure and attributes) is at least as hard as finding an optimal clustering under conventional modularity. Additionally, it is essential to ensure the incremental calculation of attribute-aware modularity for the generalization of existing strategies. In the following, we first show the time complexity of the computation of our measure. We then describe the challenges regarding its incremental computation.

As the problem of optimizing modularity (*MODOPT*) is NP-hard [BDG$^+$08], the following theorem corroborates the complexity of our measure.

Theorem 4.3.1. *The maximization of attribute-aware modularity (MAM) is NP-hard.*

Proof. We reduce the classic problem of modularity optimization to the maximization of attribute-aware modularity ($MODOPT \leq_p MAM$). We map the input graph $G = (V, E, \omega)$ of MODOPT to an input $G' = (V', E', \omega', \alpha')$ of MAM with $V' = V$, $E' = E$, $\omega' = \omega$ and $\forall u \in V, \alpha'(v) = c$ where c is a constant value $c \in \mathbb{R}$. In the transformed graph, all nodes have the same attribute value, and this transformation can be done in polynomial time. We have to show that a clustering result

$\mathcal{C}_{G'}$ of MAM corresponds to a solution of MODOPT $\mathcal{C}_G$. If all nodes in the database have the same attribute value, the attribute compactness is always maximal regardless of the cluster chosen ($\forall C_k \subseteq V' : AC(C_k) = 1$) (cf. Def. 4.1). This implies that the quality of the cluster C_k is evaluated considering only the graph structure: $AQ(\mathcal{C}_{G'}) = \sum_{C_k \in \mathcal{C}_{G'}} Q(C_k) = Q(\mathcal{C}_G)$. Thus, solving MAM for the instance G' leads to the solution of MODOPT ($\mathcal{C}_{G'} = \mathcal{C}_G$). □

This theorem justifies the use of heuristics for graph clustering on attributed graphs with attribute-aware modularity. We exploit well-established ideas from multilevel algorithms [RN11, BGLL08] and hierarchical agglomerative clustering [New04b]. In general, these greedy strategies proposed for modularity maximization are based on the increase and decrease of the score when moving a node between clusters or joining clusters. For every possible move, the algorithms have to compute a new score for each newly generated clustering. Thus, efficiency relies on a careful design that avoids to recompute the new scores for each new possible clustering result. However, they are not directly applicable to attributed graphs. Therefore, both modularity and attribute information have to allow an incremental calculation.

4.4. Attribute Information

To exclude irrelevant attributes for the assessment of a cluster, we introduce attribute compactness. Due to its sensitivity to outliers, our graph-cluster score based on attribute compactness decreases if an outlier is part of a cluster. For its efficient computation we provide the algebraic transformations required for an incremental calculation. This allows us to score both *modularity* and *attribute compactness* of a cluster for large attributed graphs.

Robustness

The variance indicates how scattered data points are. However,it does not measure the relevance of an attribute within a group of objects. To this end, a measure known as *relevance* has been introduced [YCN04]. Given the local variance $\sigma_i^2(C_k)$ of projected values on dimension d_i within the cluster C_k and the global variance $\bar{\sigma}_i^2$ on d_i, the relevance of attribute d_i for cluster C_k is defined as follows:

$$R_i(C_k) = \begin{cases} 1 - \frac{\sigma_i^2(C_k)}{\bar{\sigma}_i^2} & \text{if } \bar{\sigma}_i^2 \neq 0 \\ 1 & \text{if } \bar{\sigma}_i^2 = 0 \end{cases} \tag{4.1}$$

Relevance compares the variance within a cluster (local variance) to the one of the entire data set (global variance) for an attribute. Low variance within the cluster means that

the attribute value shows high compactness compared to the whole database. When the attribute shows scattered values in a cluster ($\sigma_i^2(C_k) \geq \bar{\sigma}_i^2$), the relevance of the attribute is negative ($R_i(C_k) \leq 0$).

While the relevance quantifies the similarity degree of an attribute in a cluster, it does not exclude irrelevant attributes as it has been designed for a single attribute. Thus, we introduce attribute compactness to summarize the similarity of all attribute values within a cluster being robust with the irrelevant ones:

Definition 4.1:

Attribute Compactness
Given a cluster $C_k \in \mathcal{C}$, we define the attribute compactness $AC(C_k)$ as follows:

$$AC(C_k) = \frac{1}{d} \sum_{d_i \in D} max(R_i(C_k), 0)$$

$AC(C_k)$ is the sum of the relevances of each attribute for the cluster C_k. Irrelevant attributes show highly scattered values within the cluster C_k compared to the entire database. As a consequence, the relevance of such attributes is negative or zero ($R_i(C_k) \leq 0$). The attribute compactness leaves aside such scattered values since it considers only the positive values (relevant attributes). In consequence, irrelevant attributes do not have any impact on the quality assessment. The higher the value of $AC(C_k)$, the more attributes have similar values within the graph cluster. Since the number of relevant attributes may be different between clusters, we normalize the score with the dimensionality of the database, to ensure the same upper bound w.r.t. modularity. Hence, the range of this measure is: $0 \leq AC(C_k) \leq 1$. The maximum value is achieved when all the attributes have the same values within the cluster. If all attributes show locally scattered attributes, the attribute compactness has its minimum value. A global variance of zero $\bar{\sigma}_i^2 = 0$ indicates that the attribute has one value for the whole graph structure. In this case, attribute d_i is relevant regardless of the selected cluster w.r.t. the graph structure.

Attribute-aware modularity

The next step is to combine attribute compactness with modularity of the graph structure to one score. A clustering taking graph structure and vector data into account has to trade off: (1) high intra-edge and low inter-edge density and (2) similar attribute values of nodes within a cluster. In this work, we use the following simple function, but other alternatives are possible. This particular function has shown good results in our experimental evaluation (cf. Section 4.6) and it fulfills some interesting properties described in the following.

Definition 4.2:

Attribute-aware modularity $AQ(\mathcal{C})$
For a clustering $\mathcal{C}$, the attribute-aware modularity $AQ(\mathcal{C})$ is:

$$AQ(\mathcal{C}) = \sum_{C_k \in \mathcal{C}} AC(C_k) \cdot Q(C_k)$$

Since $0 \geq AC(C_k) \leq 1$, the range of *modularity* is preserved as $-\frac{1}{2} \leq AQ(\mathcal{C}) \leq 1$ with this score function [BDG+08]. A value of $AQ(\mathcal{C}) = 0$ means that no attributes has any dependency with the structure (e.g., disassortative network [New03]), or the edges within the cluster are not better than random. Both the minimum and maximum value of $AQ(\mathcal{C})$ appear when the node attributes have the same values for all nodes ($\forall d_i \in D, \bar{\sigma}_i^2 = 0$). In these cases, the graph structures are special cases (e.g., a graph without edges) as discussed in [BDG+08]. In contrast to traditional modularity, a clustering with maximum $AQ(\mathcal{C})$ may have a cluster which consists of a single node regardless of its connections. This case appears for nodes with highly deviating values (outliers), as they reduce the compactness of a cluster when being added to it. Thus, the quality is decreased when adding them to a cluster result ($AC(C_k) \approx 0$) instead of considering these nodes as singletons ($AC(C_k) = 1$). Overall, it is likely that the resulting clusters are smaller as more information is considered for the clustering. Our empirical evaluation in Section 4.6 corroborates this. Overall, the higher the value of attribute-aware modularity, the better is the result w.r.t. both the graph structure and the attribute values. Thus, we have to find an optimal clustering for the maximization of attribute-aware modularity (MAM): $\arg\max_{\mathcal{C} \in A(G)} AQ(\mathcal{C})$. In the following we discuss the algorithmic challenges derived from this new problem setting.

Incremental Calculation

Attribute compactness requires the local variance of each cluster. The sum of squares S_i required for calculating the local variance of attribute i can be computed efficiently by scanning the data points once if one deploys the traditional *one-pass equation* for the sum of squares:

$$\sigma_i^2(C_k) = \frac{S_i(C_k)}{|C_k|} = \frac{\sum_{u \in C_k} \alpha_i(v)^2 - \frac{1}{|C_k|} \cdot \left(\sum_{u \in C_k} \alpha_i(v) \right)^2}{|C_k|} \tag{4.2}$$

However, this traditional expression leads to numerically unstable algorithms. In particular, precision decreases when $|C_k|$ is large and the variance is small. This loss of precision can be so severe that there is a negative value for $\sigma_i^2(C_k)$ even though the variance must be always positive [CGL83]. Hence, we have to design algorithms

that calculate the attribute compactness being both incremental and numerically stable. In the following, we provide the algebraic transformations for the incremental stable computation of *attribute-aware modularity* in order to generalize the several strategies based on node or cluster movements. Several strategies are focused on node or cluster movements. Therefore, we have to provide the algebraic transformations required for: (1) adding a node to a cluster, (2) removing it, (3) joining two clusters and (4) removing a cluster.

Adding a Node For the stable calculation of the unweighted sum of squares, the following recursive computation scheme has been proposed in [YC71], where v is the node to be added, C_k is the cluster v is added to and u is a node forming a singleton cluster:

$$T_{i,\{u\}} = \alpha_i(u), S_{i,\{u\}} = 0 \tag{4.3a}$$

$$T_{i,C_k\cup\{v\}} = T_{i,C_k} + \alpha_i(v) \tag{4.3b}$$

$$S_{i,C_k\cup\{v\}} = S_{i,C_k} + \frac{\left(|C_k|\cdot\alpha_i(v) - T_{i,C_k\cup\{v\}}\right)^2}{|C_k|\cdot(|C_k|-1)} \tag{4.3c}$$

The variable $T_{i,C_k\cup\{v\}}$ contains the sum of the attribute values of C_k together with the one of node v. Then, the sum of squares $S_{i,C_k\cup\{v\}}$ is updated as well. This procedure avoids the subtraction of squared terms, and it is mathematically equivalent to Equation 4.2. At the same time, it allows a numerically stable calculation of the sum of squares in constant time. With this algorithm, we can compute the variance change when adding a node v to a graph cluster C_k with the following expression:

$$\Delta\sigma^2_{i,C_k\cup\{v\}} = \sigma_i^2(C_k\cup\{v\}) - \sigma_i^2(C_k) = \frac{S_{i,C_k\cup\{v\}}}{|C_k|+1} - \frac{S_{i,C_k}}{|C_k|} \tag{4.4a}$$

$$= \frac{\left(|C_k|\cdot\alpha_i(v) - T_{i,C_k}\right)^2 - (|C_k|+1)\cdot S_{i,C_k}}{(|C_k|+1)^2\cdot|C_k|} \tag{4.4b}$$

We determine the increase in *attribute compactness* when adding a node v to a cluster as follows:

$$\begin{aligned}\Delta AC_{C_k\cup\{v\}} =& \frac{1}{d}\sum_{d_i\in D}\Big[max(1 - \frac{\sigma_i^2(C_k) + \Delta\sigma^2_{i,C_k\cup\{v\}}}{\bar{\sigma_i^2}}, 0) \\ & - max(1 - \frac{\sigma_i^2(C_k)}{\bar{\sigma_i^2}}, 0)\Big]\end{aligned} \tag{4.5}$$

As our measure depends on both modularity and the attribute compactness, the increase of attribute-aware modularity when adding a node to a cluster has to take into account both measures. Let $E_{v,C_k} = \{\{u,v\} \in E : u \in C_k\}$ be the set of intra-cluster

edges added if node v is added to the graph cluster C_k.The increase of the modularity $\Delta Q_{C_k \cup \{v\}}$ can be computed as:

$$\Delta Q_{C_k \cup \{v\}} = \frac{W(E_{v,C_k})}{W(E)} - \frac{2 \cdot deg(C_k) \cdot deg(v)}{4 \cdot W(E)^2} \tag{4.6}$$

Taking this into account, the increase of attribute-aware modularity of cluster C_k when vertex v is added to it is given by:

$$\begin{aligned} \Delta AQ_{C_k \cup \{v\}} =& Q(C_k) \cdot \Delta AC_{C_k \cup \{v\}} + \Delta Q_{C_k \cup \{v\}} \cdot AC(C_k) \\ & - \Delta Q_{C_k \cup \{v\}} \cdot \Delta AC_{C_k} \end{aligned} \tag{4.7}$$

Removing a node A similar procedure can take place to evaluate the decrease of attribute-aware modularity when removing a node. In this case, we first consider the following recursive expression to compute the decrease of the sum of squares:

$$\begin{aligned} T_{i,C_k \setminus \{v\}} &= T_{i,C_k} - \alpha_i(v) \\ S_{i,C_k \setminus \{v\}} &= S_{i,C_k} - \frac{\big(|C_k| \cdot \alpha_i(v) - T_{i,C_k}\big)^2}{|C_k| \cdot (|C_k| - 1)} \end{aligned}$$

Then the variance change by removing node v from cluster C_k can be formulated as:

$$\Delta \sigma_{i,C_k \setminus \{v\}} = \frac{(|C_k| - 1) \cdot S_{i,C_k} - (|C_k| \cdot \alpha_i(v) - T_{i,C_k})^2}{|C_k| \cdot \big(|C_k| - 1\big)^2}$$

Thus, the difference of *attribute compactness* is:

$$\begin{aligned} \Delta AC_{v \setminus C_k} = & \frac{1}{d} \sum_{d_i \in D} max(1 - \frac{\sigma_i^2(C_k) - \Delta \sigma_{i,C_k \setminus v}}{\bar{\sigma_i^2}}, 0) \\ & - \frac{1}{d} \sum_{d_i \in D} max(1 - \frac{\sigma_i^2(C_k)}{\bar{\sigma_i^2}}, 0) \end{aligned} \tag{4.10}$$

Finally, removing a node v implies a change in the *attribute modularity* (cf. Definition 4.2):

$$\begin{aligned} \Delta AQ_{v \setminus C_k} = & \Delta Q_{C_k \setminus v} \cdot \Delta AC_{C_k \setminus v} - M(C_k) \cdot \Delta AC_{C_k \setminus v} \\ & - \Delta Q_{C_k \setminus v} \cdot AC(C_k) \end{aligned}$$

Joining a cluster Using the incremental recursive algorithm for calculating the sum of squares (cf. Equation 4.3), the resulting sum of squares by joining clusters C_k and C_p

can be formulated as:

$$T_{i,C_k \cup C_p} = T_{i,C_k} + T_{i,C_p} \tag{4.11a}$$

$$\begin{aligned} S_{i,C_k \cup C_p} =& S_{i,C_k} + S_{i,C_p} \\ &+ \frac{|C_k|}{|C_p| \cdot (|C_k| + |C_p|)} \cdot \left(\frac{|C_p|}{|C_k|} \cdot T_{i,C_k} - T_{i,C_p} \right)^2 \end{aligned} \tag{4.11b}$$

So the variance change can be formulated as:

$$\begin{aligned} \Delta\sigma^2_{i,C_k \cup C_p} =& \frac{S_{C_p}}{|C_k| + |C_p|} + \frac{|C_k| \cdot \left(\frac{|C_p|}{|C_k|} \cdot T_{C_k} - T_{C_p} \right)^2}{|C_p| \cdot (|C_p| + |C_k|)^2} \\ &- \frac{|C_p| \cdot S_{C_k}}{|C_k| \cdot (|C_k| + |C_p|)} \end{aligned} \tag{4.12}$$

Removing a cluster We use a similar recursive expression (cf. Equation 4.11a and Equation 4.11b) to calculate the variance change when removing a cluster C_p from C_k:

$$\begin{aligned} \Delta\sigma^2_{i,C_k \setminus C_p} =& \frac{|C_p| \cdot S_{C_k}}{|C_k| \cdot (|C_k| + |C_p|)} - \frac{S_{C_p}}{|C_k|} \\ &- \frac{\left(\frac{|C_p|}{|C_k|} \cdot T_{C_k} - T_{C_p} \right)^2}{|C_p| \cdot (|C_k| + |C_p|)} \end{aligned} \tag{4.13}$$

In consequence, we cannot only update the attribute information for each node or cluster movement in constant time, but we can ensure high quality results due to the numerically stable calculation of the attribute compactness. These expressions are essential for the generalization of all strategies for modularity maximization that compute the quality of a clustering incrementally. In the following we generalize of several relevant components of multilevel algorithms [RN11] for this new score.

4.5. Algorithms

We leverage well-established ideas from multilevel algorithms [RN11, BGLL08]. However, these algorithms have been proposed for conventional graphs (without attributes), and are not applicable to attributed graphs. In addition to the formal property of incremental calculation of the quality measure (cf. Section 4.4), further considerations specific to more complex data structures (attributed graphs) are necessary. The so-called *coarsening phase* of multilevel algorithms requires the contraction of the graph structure and movements of nodes in this new structure. Attributes have to be considered in each of its steps. In particular, it has two major phases: (1) The input graph is

contracted to a new graph whose nodes are clusters from an initial clustering $\mathcal{C}_0$ (*contraction* step), and (2) local movements are applied to this new graph [BGLL08]. In the following, we describe the generalizations of them to attributed graphs since they are the basis for further development of multilevel algorithms.

Local Move (LM)

This heuristic considers only local moves of single nodes. For each possible node move, when v is added to C_k, the change of attribute-aware modularity is calculated. v is moved to the cluster with the highest increase in attribute-aware modularity.

Algorithm Given an initial clustering $\mathcal{C}_0$, we first compute the change of attribute-aware modularity $\Delta AQ_{C_k \setminus \{v\}}$ when v is removed from its current cluster C_k. As candidates for possible movements, we select all the graph clusters in $\mathcal{C}_0$ that contain a node adjacent to v. Then we compute the increase of attribute-aware modularity $\Delta AQ_{C_n \cup \{u\}}$ if v is added to $C_n \in candidates$. We only consider node moves that increase the attribute-aware modularity. Finally, we add v to the cluster C_{dest} where the increase of *attribute-aware modularity* is maximal. If no move of v to other clusters does not increase the quality of the clustering, then node v is not moved. Additionally, some nodes increase the score when they form a singleton. For example, an outlier with highly deviating attribute values heavily brings down attribute compactness when seen as part of a cluster. So, to remove this node from a cluster improves the attribute compactness; this in turn leads to an increase of attribute-aware modularity. Therefore, we also consider the empty cluster in the set of candidates (Line 7). The algorithm terminates if there are no more score-increasing vertex movements.

Complexity As each node is considered once for a local movement, the algorithm requires at least $|V|$ iterations. The current cluster of a vertex u can be retrieved in $\mathcal{O}(1)$ time (Line 4). For the decrease of attribute-aware modularity, we have to compute the decrease of modularity and attribute compactness. Updating the value of modularity when removing a node v from its cluster C_k can be done in constant time. Due to the incremental calculation of attribute compactness, we determine it in constant time for each dimension ($\mathcal{O}(d)$) as well. Overall, the decrease of attribute-aware modularity is calculated in $\mathcal{O}(d)$ time. For the selection of candidates (Line 7), we have to scan all edges of node u to determine the neighboring clusters. For all candidates, the increase of attribute-aware modularity is calculated in $\mathcal{O}(deg(u) \cdot d)$. Thus, all nodes are analyzed once in $\mathcal{O}\Big(|V| + \sum_{u \in V} d \cdot deg(u)\Big) = \mathcal{O}(|V| + |E| \cdot d)$. Regarding the graph size, we have achieved the same time complexity as the same heuristic that only considers the graph structure [RN11], due to the incremental calculation of attribute compactness. On the other hand, the time complexity w.r.t. the number of attributes is linear. This enables an efficient calculation of attribute-aware modularity for both large and high-dimensional graphs, as shown in Section 4.6.

Algorithm 1 Local Move (LM)

Input: $G = (V, E, \alpha, \omega)$, initial clustering $\mathcal{C}_0$
Output: new clustering result $\mathcal{C}$

```
1:  C := C_0                                  ▷ Initialize with an initial clustering
2:  do
3:      for v ∈ V do
4:          C_k := C(v)                       ▷ Select graph cluster of node v
5:          decrease := ΔAQ_{C_k \ {v}}       ▷ cf. Equation 4.9
6:          remove v from C_k
7:          cand := {C(u) | {u, v} ∈ E}
8:          C_dest := arg max_{C_n ∈ cand} ΔAQ_{C_n ∪ {v}}
9:          if increase + decrease ≥ 0 then
10:             add v to C_dest and update C
11:         end if
12:     end for
13: while node movement
```

Coarsening Phase

Considering only local movements of a single node can lead to a local maximum. To avoid this, multilevel algorithms create a new graph to allow movements on a higher level [BGLL08, RN11]. In this graph, nodes represent clusters. Thus, a set of objects are moved for each of these local movements instead of single nodes. Therefore, the contraction of the attribute values has to be done carefully, to ensure the efficiency in the contraction phase. As the nodes of G' represent a set of nodes, we use expressions to compute incrementally the attribute compactness for joining and removing a cluster. Since our modularity-driven approach follows the general definition of modularity (weighted graphs and self-loops are allowed), we are able to contract the graph using attribute-aware modularity.

Algorithm First, we build the nodes of the new contracted graph G' for the coarsening phase. For each cluster C_k, we create a new node v_k and a new edge e_{new} which is a self loop. It stores the sum of the weights of the edges within C_k. Regarding the attributes, both the sum of the attribute values and their sum of squares is also stored in the new graph as attributes. Then, we create the edges that connect the nodes in the new graph G_{new}. If two clusters are connected in the original graph, we add an edge connecting the nodes representing these two clusters. The weight of this new edge is the sum of all the edge weights connecting both clusters in the original graph. To complete the coarsening phase, we apply the algorithm *Local Move* (cf. Algorithm 1) to the contracted graph.

Complexity The coarsening phase requires the creation of a new graph and the update

of the attribute information. In the worst case, each node is a singleton in the initial clustering $\mathcal{C}_0$. The creation of the new nodes in the contracted graph is $\mathcal{O}(|V| \cdot d)$. The creation of connecting edges between the new nodes can be done in $\mathcal{O}(|E|)$ if we only scan the edges twice. In total, the contraction phase can be done in $\mathcal{O}(|V| \cdot d + |E|)$. Finally, the *Local Move* (cf. Algorithm 1) on a contracted graph can be done in $\mathcal{O}(|V| + |E| \cdot d)$, due to the incremental calculation when clusters are removed or moved to other clusters. Overall, coarsening requires $\mathcal{O}((|V| + |E|) \cdot d)$. A careful design considering the attribute values and the incremental calculation of attribute-aware modularity has allowed to preserve the efficiency of the original strategy regarding the graph size. At the same time, the attribute information has been considered by providing a strategy which is linear with the number of attributes.

Algorithm 2 Coarse Iteration

Input: $G = (V, E, \alpha, \omega)$, initial clustering $\mathcal{C}_0$
Output: clustering $\mathcal{C}$

create the graph $G_{new} = (V', E', \omega', \alpha')$
$V' = \{v_{C_k} | C_k \in \mathcal{C}_0\}$ ▷ Node v_{C_k} representing graph cluster C_k
$E' = \{v_{C_k}, C_k \in \mathcal{C}_0 \mid \forall v_{C_k}\{v_{C_k}, v_{C_K}\}\}$ ▷ Creation of self-loops
$E' = E' \cup \{\{v_{C_k}, v_{C_j}\} \mid \exists\{w, u\} \in G$ with $w \in C_k \wedge u \in C_j\}$
for $C_k \in \mathcal{C}_0$ **do** ▷ Update edge weights and attribute information
    for $\{p, v \in V \mid \exists\{p, v\} \in E$ with $p \in C_k \wedge v \in C_k\}$ **do**
        $\omega'(\{v_{C_k}, v_{C_k}\}) := \omega'(\{v_{C_k}, v_{C_k}\}) + \omega(\{p, v\})$ ▷ weights of self-loops
    end for
    for $\{p, v \in V \mid \exists\{p, v\} \in E$ with $p \in C_k \wedge v \in C_j\}$ **do**
        $\omega'(\{v_{C_k}, v_{C_j}\}) := \omega'(\{v_{C_k}, v_{C_k}\}) + \omega(\{p, v\})$
    end for
    for $d_i \in D$ **do**
        $\alpha'_i(v_{\mathcal{C}_k}) := (S_{i,C_k}, T_{i,C_k})$ ▷ Store recursive variables: see Equations 4.3
    end for
end for
LM$(G_{new}, \mathcal{C}_{new})$ ▷ Algorithm 1 using Equations 4.12 and 4.13

In the framework proposed in [RN11], other algorithms can be combined for the generation of the clustering result for the contraction phase (e.g., merging clusters instead of local movements). Thus, we also have generalized the approach proposed in [NG04] for merging clusters globally instead of local movements of the nodes. However, we show that local approaches (i.e., *LM*) show better quality results and more robustness w.r.t. outliers.

Global Merge

We generalize an algorithm following the paradigm of agglomerative hierarchical clustering [New04b]. In contrast to the previous algorithms [BGLL08, RN11], the work in [New04b] focuses on the merge of clusters instead of moving nodes locally. The goal is to find a merge of clusters that maximizes attribute-aware modularity. The generalization of this particular heuristic relies on the incremental calculation when joining two clusters. Since the attribute-aware modularity values of all possible merges have to be computed, this is the most relevant consideration for its generalization.

Algorithm 3 Global Merge

Input: $G = (V, E, \alpha, \omega)$, initial clustering $\mathcal{C}_0$
Output: clustering $\mathcal{C}$

```
1:  increase := 0
2:  initialize priority queue pq
3:  for {C_k, C_n ∈ C_0 : C_n ≠ C_k} do
4:      for C_n ∈ C_0 do
5:          if ∃u, v ∈ V : u ∈ C_k, v ∈ C_n ∧ {u, v} ∈ E then
6:              score := ΔAQ_{C_k ∪ C_n}                    ▷ cf. Equation 4.12
7:              if score > 0 then
8:                  pq.add(score,C_k,C_n)
9:              end if
10:         end if
11:     end for
12: end for
13: while ¬pq.isEmpty() do
14:     (score, C_k, C_n) := pq.pop()
15:     increase += score
16:     C_new := merge C_k with C_n
17:     for {(−, C_m, C_p) ∈ pq | C_m = C_n ∨ C_m = C_k} do
18:         pq.remove(-,C_m,C_p)
19:         score:= ΔAQ_{C_new ∪ C_p}                       ▷ cf. Equation 4.12
20:         if score > 0 then
21:             pq.add(score, C_new, C_n)
22:         end if
23:     end for
24: end while
```

Algorithm We use a priority queue where each merge $(C_k \cup C_n)$ is associated with a priority given by the increase of *attribute-aware modularity* when joining them. First, we compute the new scores of all possible merges between two clusters. Scores are only computed for neighboring clusters as non-connected clusters disagree with the

properties of *modularity* [BDG+08]. In addition, we add only merges with a positive increase to the priority queue as the goal is to increase the quality of the clustering with merge operations. The top of this priority queue is the merge between two clusters with the highest increase. So, we extract iteratively the pair of clusters with the highest increase in *attribute-aware modularity*. In each iteration, two clusters are merged to a new one C_{new} and the information of the priority queue is updated. We delete old scores belonging the clusters C_n or C_k as they have been already merged to the new cluster C_{new}. As new candidate for possible merges, we extract the pending clusters from the priority queue in order to compute the new score when merging them with C_{new}. Finally, this process ends when the priority queue is empty. This occurs if any merge increases the *attribute-aware modularity*.

Complexity At the beginning, we have to compute all possible merge candidates. In the worst case, the initial clustering consist of singleton clusters. Thus, we have $|E|$ candidates for each node as we only consider clusters having at least an edge between them. For an efficient calculation of the attribute compactness, we use the incremental computation for joining clusters (cf. Equation 4.12). The increase of *modularity* in constant time updating a list of the edge weight between the graph clusters in each merge. Therefore, we can update the *attribute-aware modularity* in $\mathcal{O}(d)$ in each step. The update of the priority queue can be done in the worst case in $\mathcal{O}(log|E|)$ if it is implemented as a binary heap. Overall, the first loop (Lines 3 - 12) requires $\mathcal{O}(|E| \cdot (log|E| + d))$ for the initialization of the priority queue. At the beginning, the algorithm requires initial merge scores as each node is a singleton. In the case that the whole graph is a cluster, the second loop has to iterate $|V| - 1$ times in the worst case. As $\mathcal{O}|V|$ scores have to be updated, the total cost is $\mathcal{O}(|V| \cdot (d + log(|E|)))$. Thus, the time complexity of the second loop is $\mathcal{O}(|V|^2 \cdot (log(|E| + d)))$. Similarly to Local Move and Coarsening, the incremental calculation of an attribute-aware modularity has enabled to preserve the time complexity regarding the graph size. This algorithm has the worst time complexity (quadratic w.r.t. the number of nodes) as already shown in [RN11]. However, the combination of this algorithm with other strategies for maximizing attribute-aware modularity leads to faster results. Before showing the experimental results of each of these algorithms, we first describe our scaffold algorithm proposed for the maximization of attribute-aware modularity.

Scaffold Algorithm

For the maximization of attribute-aware modularity, we additionally propose a scaffold algorithm that combines the different heuristics previously described. All these algorithms can be used separately, or they can be combined using the output clustering as input of the next one. In our evaluation in Section 4.6, we show that the combination of multiple techniques may lead to better or faster results than using a single one. Algorithm 4 shows our framework for attribute-aware modularity.

Algorithm 4 Scaffold Algorithm

Input: $G = (V, E, \alpha, \omega)$
Output: a clustering $\mathcal{C}$
for all $d_i \in D$ **do** ▷ Precompute global variables
    $\bar{\sigma_i^2}$:= the global variance of dimension d_i
end for
compute weighted degree m for each node $u \in V$
$\mathcal{C} := \{\{u\} \mid u \in V\}$ ▷ Initialize clustering as singletons
while strategy to be applied **do**
    $strategy := \{LM, CLM, GM\}$ ▷ Select strategy
    $\mathcal{C}$:= run($strategy$,$\mathcal{C}$)
end while

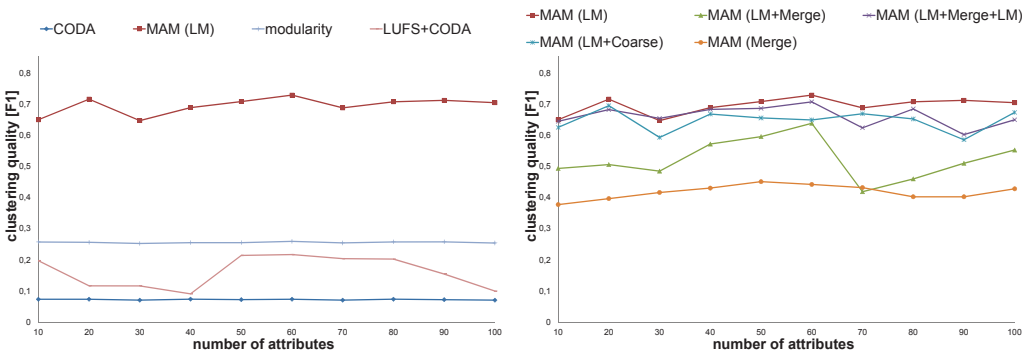
4.6. Experiments

We compare our approach *MAM (maximization of attribute-aware modularity)* with *CODA*, an outlier-aware graph clustering algorithm for attributed graphs [GLF+10], *LUFS+CODA*, a feature selection algorithm as pre-processing [TL12], and *modularity* in a conventional multilevel algorithm [RN11]. For all competitors we tried to find optimal parameters, while our method does not require fixing any parameters. Internally, we evaluate *MAM* with its different strategies *Merge*, *LM*, *Coarse*, and its combinations (*LM+Merge*, *LM+Merge+LM*). We use both synthetic and real world data and the *F1-value* (harmonic mean of precision and recall) for quality assessment. In order to facilitate comparability and repeatability of our experiments, we provide all datasets, algorithms, and parameter settings[2].

Synthetic Data

We generated synthetic datasets of different graph sizes, dimensionalities, and outlier ratios. For each experiment, we generate five graphs to average over random effects in the generation process. The generated graphs are based on benchmark graphs [LFR08], with power law distributions in both degree and community size. We have extended this generator adding numeric node attributes as well as community outliers. For each graph cluster C_k, we randomly select a number of relevant attributes $x \in [1, d]$ and choose their attribute values based on a Gaussian distribution with a mean μ_{C_k} and a very small standard deviation ($\sigma = 0.001$). The values for irrelevant attributes are chosen following a Gaussian distribution with $\sigma = 1$. Finally, we generate outliers by selecting clustered nodes and manipulating a random number of their relevant attribute values.

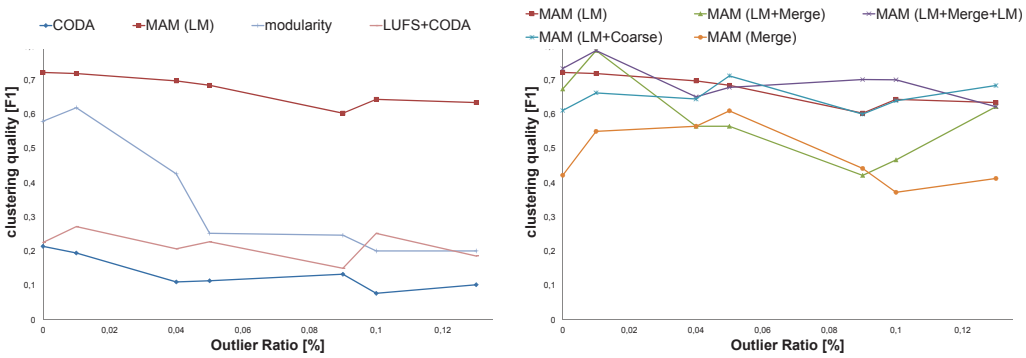
[2] http://ipd.kit.edu/~muellere/mam/



(a) *Comparison with competitors* (b) *Comparison between strategies*

Figure 4.1.: Quality w.r.t. number of attributes on synthetic data

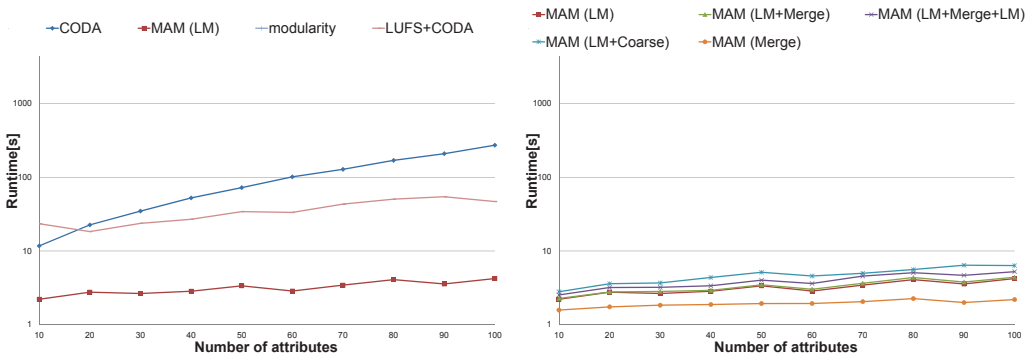
Quality and Robustness We evaluate quality on graphs with an increasing number of attributes and of outliers, see Figures 4.1 and 4.2. The robustness of our attribute-aware modularity is clearly visible for all of our strategies. Results are best for *LM*, *LM+Coarse*, and *LM+Merge+LM*. Since traditional modularity does not consider attribute information, the obtained clusters show dissimilarity w.r.t. the attribute values. For instance, they include outliers with highly deviating values. On the other hand, *CODA* does not perform a local attribute selection and is prone to irrelevant attributes. Although *CODA* is slightly improved by the feature selection (*LUFS + CODA*), it does not work well as a pre-processing step. This is because it does not select the relevant attributes for each cluster locally.



(a) *Comparison with competitors* (b) *Comparison between strategies*

Figure 4.2.: Quality w.r.t. outlier ratio on synthetic data

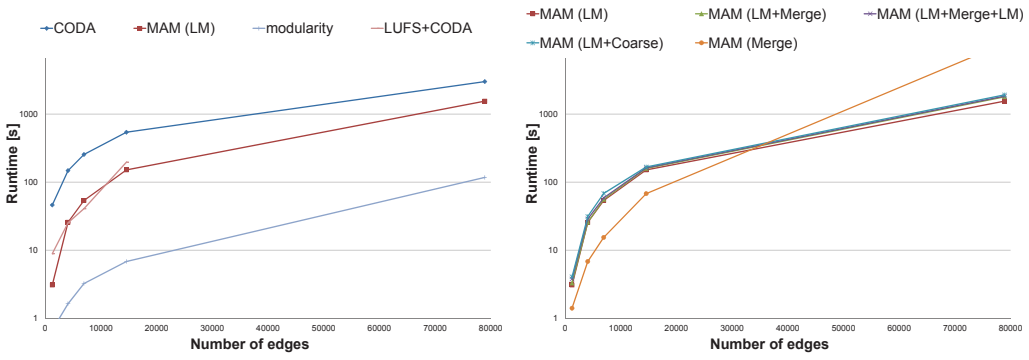
Runtime In contrast to the traditional modularity, the runtime of the algorithms for attributed graphs depends on both the graph size and the number of attributes, see Figures 4.3, 4.4 and 4.5. The runtime of our attribute-aware modularity is slightly higher



(a) *Comparison with competitosr* (b) *Comparison between strategies*

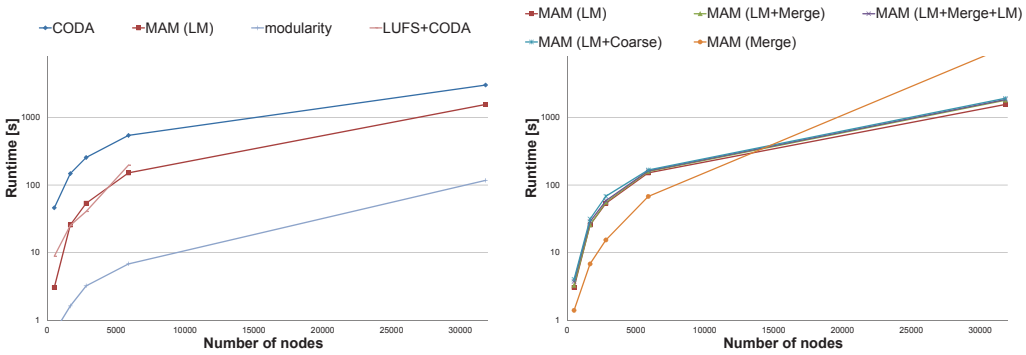
Figure 4.3.: Runtime w.r.t. $|D|$ on synthetic data

than traditional modularity due to the combination of both information sources. However, due to the incremental calculation, all our strategies scale linearly with the number of attributes (cf. Figure 4.3). In contrast, *CODA* does not scale w.r.t. these parameters. *LUFS+CODA* perform better w.r.t. the number of attributes since *CODA* only considers few attributes. However, the pre-processing step *LUFS* does not scale with the graph size. The strategy with the worst performance is *Merge*, due to its time complexity cf. [New04b]. However, the combination with other strategies (*LM+Merge*) causes significant speedups. Overall, our incremental and numerically stable calculation of attribute-aware modularity leads to accurate results (Figure 4.1 and Figure 4.2) as well as to efficient computation (Figures 4.3, 4.4 and 4.5).



(a) *Comparison with competitosr* (b) *Comparison between strategies*

Figure 4.4.: Runtime w.r.t. $|E|$ on synthetic data



(a) *Comparison with competitosr* (b) *Comparison between strategies*

Figure 4.5.: Runtime w.r.t. $|V|$ on synthetic data

Real World Data

We have evaluated our approach on attributed graphs from different application domains. Table 4.2 shows a summary of statistics for these real-world networks. All of them have already been public available in [MISMB13, ISML$^+$13, GFRS13, ISMIB14]. Since there is no ground truth given, it is difficult to impossible to do a quality assessment. However, we compare algorithms regarding the attribute-aware modularity achieved and runtime. This is commonly done in the literature to compare strategies for modularity maximization [LFR08, RN11]. Further, we describe interesting clusters and outliers found as case studies.

Graph	#nodes	#edges	#attributes
Disney [MISMB13]	124	333	28
DFB [GFRS13]	100	1106	5
ARXIV [GFRS13]	856	2660	30
IMDB [GFRS13]	862	2660	21
DBLP [ISMIB14]	28112	95410	46
PATENTS [GFRS13]	100000	188631	5
Amazon [ISML$^+$13]	314824	882930	28

Table 4.2.: Real World Networks

Attribute-Aware Modularity and Runtime

We analyze performance (i.e., $AQ(\mathcal{C})$) and runtime) w.r.t. different graph sizes and number of attributes in real world networks. Table 4.3 shows the clustering score

$AQ(\mathcal{C})$ obtained by different strategies for its maximization and the competitors. Similarly to the synthetic data, *Merge* performs worst. Considering both runtime (cf. Table 4.4) and quality, local movement *LM* achieves very good results compared to more complex strategies such as *LM+Merge+LM*. Further, it is the only algorithm on attributed graphs that scales up to the largest network in our evaluation. Other schemes were not able to provide a result within 5 hours. This scalability is due to the incremental calculation of attribute-aware modularity. Traditional modularity achieves good clustering results; however, it neglects attribute information and does not find homogeneous attribute values in each cluster. Although CODA takes attribute information into account, it degenerates with an increasing number of attributes. $AC(\mathcal{C})$ is best on low dimensional data (*DFB* and *PATENTS* with 5 attributes only). On the other hand, pre-processing for the selection of attributes (*LUFS+CODA*) increases the quality. However, it does not scale with the graph size, and it does not select the relevant attributes for each cluster locally.

	LM	**Merge**	**LM+Merge**	**LM+Merge+LM**	**Coarse**	**modularity**	**CODA**	**LUFS+CODA**
Disney	**0.368**	0.038	0.182	0.210	0.329	0.306	0.0	0.164
DFB	**0.123**	0.001	0.001	0.0031	**0.122**	0.0686	0.001	0.015
ARXIV	**0.246**	0.0126	0.135	0.209	0.235	0.159	0.0	0.059
IMDB	0.205	0.012	0.077	0.0806	**0.218**	0.128	0.0	0.0
DBLP	0.429	-	0.155	0.224	**0.436**	0.365	0.0	-
PATENTS	0.353	-	0.389	**0.406**	-	0.162	0.064	-
Amazon	0.513	-	-	-	-	0.064	-	-

Table 4.3.: Clustering quality achieved $AQ(\mathcal{C})$ on real world networks

	LM	**Merge**	**LM+Merge**	**LM+Merge+LM**	**Coarse**	**modularity**	**CODA**	**LUFS+CODA**
Disney	0.91	0.63	0.99	1.27	1.32	0.40	6.35	2.2
DFB	0.88	0.71	0.94	1.11	1.25	0.47	1.68	1.76
ARXIV	4.09	3.52	4.47	7.33	8.51	0.97	46.45	13.43
IMDB	5.17	4.99	6.57	9.66	10.90	1.89	13.03	10.87
DBLP	135	-	270.46	403.84	317.15	32.94	1294.47	-
PATENTS	477.11	-	729.52	933.41	-	95.27	711.601	-
Amazon	1354.98	-	-	-	-	493.5	-	-

Table 4.4.: Runtime [s] on real world networks

Case Studies

In the following, we discuss some results when maximizing attribute-aware modularity in real world networks.

Disney This dataset is a subgraph of the Amazon co-purchase network. Each product (node) is described by attributes such as prices or review ratings [MISMB13]. In contrast to traditional modularity, the clusters found by our method are smaller and more specialized. For instance, modularity extracts one cluster with family films such as *Spy*

Kids, *Inspector Gadget* or *Flubber*. In contrast, our method splits this into two clusters. This is because the sequels of *Honey, We Shrunk Ourselves* have very good ratings and many reviews, compared to the other movies. Similarly to modularity, our approach also finds a cluster consisting of *Read Along films* since all these films show similar Amazon prices. However, the overpriced film *The Jungle Book* is detected as an outlier. Figure 4.6 shows more examples of the differences and similarities of the extracted clusters.

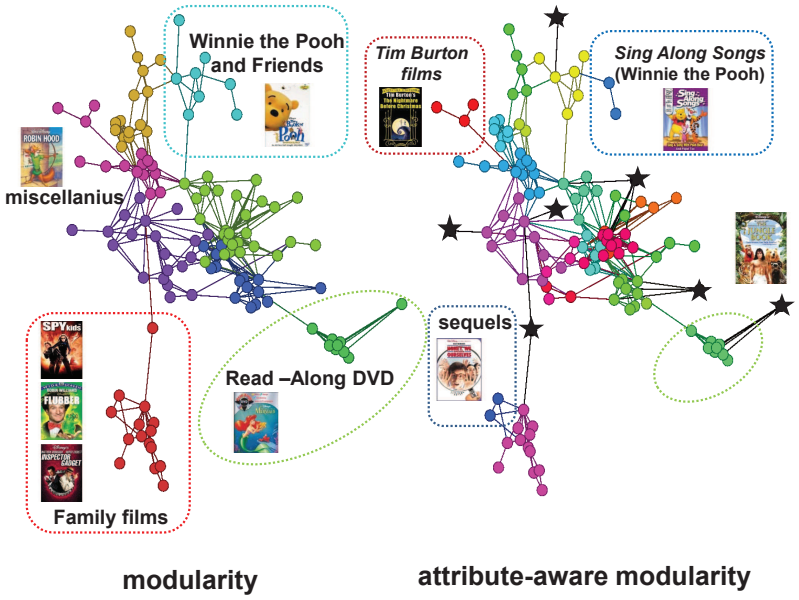


Figure 4.6.: Comparison between extracted clusters: traditional modularity vs. attributed-aware modularity. Outliers nodes are marked with stars and clusters are marked with different colors.

DBLP This graph contains authors (nodes) with co-authorship information (edges) and 40 numeric attributes (publication ratios at certain conferences) [ISMIB14]. Graph clustering based on attribute-aware modularity retrieves research communities with similar publication ratios on some conferences. Traditional modularity neglects the attribute values and brings down this similarity. For instance, it includes highly deviating outlier nodes in one of the clusters. For example, we have detected *Ted E. Senator* as an outlier. He has published several papers as single author on *ICDM* and *KDD*. He also has collaborated with individuals from two different clusters: *Henry G. Goldberg* belonging to a data mining and machine learning cluster (publishing in *ICDM*, *KDD*, *ICML*, etc.) and *Ping Shyr*, *J. Dale Kirkland*, and *Tomasz Dybala*, belonging to an artificial intelligence cluster (*AAAI\IAAI*) (*AAAI\IAAI*) (cf. Figure 4.7). *Ted. E. Senator* has highly deviating attribute values compared to both of these communities. Including him in one of the clusters would reduce the clustering quality.

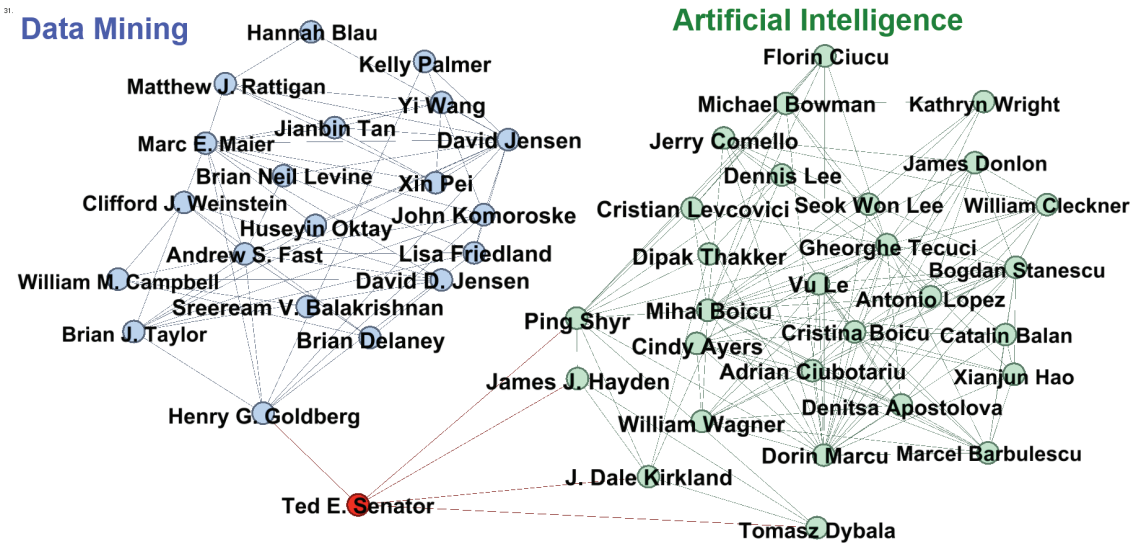


Figure 4.7.: Two clusters (data mining and artifical intelligence) and the outlier Ted. E. Senator with highly publishing ratios on KDD and ICDM compared to the clusters he is connected to

German Soccer League (DFB) This network contains soccer players characterized by attributes such as the number of goals or of games. Two players are connected if they have played in the same club [GFRS13]. One of the clusters found is a subset of players with similar numbers of goals and penalty kicks. Figure 4.8 shows the subgraph with the player names. One exceptional player is *Ulf Kirsten*, who was one of the best German goalgetters between 1980 and 1990. He has high values in several of the attributes (e.g., *number of games*, *number of penalty kicks* and *number of goals per game*). Although he is not the player with most goals or number of games in the database, the attribute values are highly deviating compared to the ones in his graph neighborhood; therefore, it is appropriate that our clustering has identified him as an outlier.

Amazon The *Amazon* co-purchase network [ISML+13] is most challenging due to both the graph size and the number of attributes. We found a cluster with highly similar ratings, number of reviews, and prices that consists of books for *day trading* (e.g. *Day Trade Part-time*, *Day Trade Online*, or *The Compleat Day Trader*). The prices by private sellers or for used books are similar. Additionally, all books have similar rating ratios for both *rating_5_star* and *rating_4_star*. In contrast to traditional modularity, our algorithm has detected this homogeneous cluster within a large subgraph including more general financial books. We conclude that attribute-aware modularity is a reasonable quality measure for cluster structures in real-world networks.

Figure 4.8.: A cluster of players with similar number of goals and penalty kicks and the outlier Ulf Kirsten

4.7. Summary

In this chapter, we have proposed a modularity-driven approach to cluster attributed graphs. To achieve this, we have proposed attribute compactness which enables a robust clustering w.r.t. irrelevant attributes and outliers. More specifically, *attribute compactness* quantifies the relevance of the attributes within a cluster. It does so by comparing the local variance to the global one. Simultaneously, its sensitivity to outliers enables to be aware when adding such a node to a cluster. Then, we have enriched modularity with the attribute compactness introducing an attribute-aware modularity which has to be maximized for clustering attributed graphs. Since we have proven that maximization of attribute-aware modularity (MAM) is an NP-hard problem, we then have aimed for heuristic strategies. We have generalized several existing strategies for modularity maximization to attribute-aware modularity. This has only been possible by providing an incremental and numerically stable calculation of this measure. Our evaluation on synthetic and real world networks shows the high quality and scalability of our approach. Specifically, our evaluation with several very different real-world data sets of different graph sizes provides anecdotical evidence that clustering based on attribute-aware modularity yields somewhat more meaningful results than approaches relying on conventional quality measures (i.e., modularity).

5. Local Context Selection for Outlier Ranking

Outlier ranking aims at the distinction between exceptional outliers and regular objects by measuring deviation of individual objects. To achieve this, it is essential to compute accurately the deviation of each node considering only the local context of each object. It is an open research question to detect such meaningful local contexts in attributed graphs.

In this Chapter[1], we tackle this challenge by proposing a local context definition. For each object, our technique determines its subgraph and its statistically relevant subset of attributes. So, this context selection enables a high contrast between an outlier and the regular objects. Out of this context, we compute the outlierness score by incorporating both the deviation in the attribute values and the information of the graph structure. In our evaluation on real and synthetic data, we show that our approach is able to detect contextual outliers that are missed by other outlier models.

5.1. Motivation

Outlier mining is an important task in the field of data management and knowledge discovery. It identifies unexpected, erroneous, rare, and suspicious data. Outlier ranking algorithms sort the objects according to their degree of deviation, instead of coming only to a binary decision for each object. This ranking eases a user-driven exploration of outliers, by looking at the most deviating objects first. In the past, outlier mining techniques have focused on vector data or graph data separately [Agg13]. However, more and more applications such as network intrusion, rare protein interactions, financial fraud, or data cleaning demand outlier analysis on combinations of both. They store relationships between objects represented as a *graph* and additional *attributes* for each node, and mine outliers in this combined data space.

In particular, we consider electronic platforms as exemplary application of outlier mining on attributed graphs. Electronic marketplaces try to detect and delete fraudulent

[1] This chapter is an extension of the published work in the Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM 2014) [ISMIB14]

product placements since their reputation is highly affected by such fraud. Fake products, overpriced products, or manipulated reviews are examples for outliers that have to be detected. Such electronic platforms provide a large number of descriptive *attributes* for each product (e.g., prices of all sellers, ratings, and product reviews) and the product relations stored in the *graph* of frequently co-purchased products. All of this publicly available information can provide more information for the detection of outliers. However, with more and more information (attributes, nodes, edges) becoming available, not all of it is relevant for data analysis. For instance, an object may be an outlier only w.r.t. a *selection of the attributes* and a *local graph neighborhood*. We call this the *context of an outlier*, in line with publications on contextual outliers and community outliers [GLF+10, SWJR07].

product	sales	reviews	price
1	262	76	25
2	25	30	30
3	155	47	**150**
4	69	105	20
5	80	8	35
6	182	7	15
7	22	5	8
8	234	**28**	12
9	102	8	5
10	248	6	13
11	10	4	10
...	...	...	...

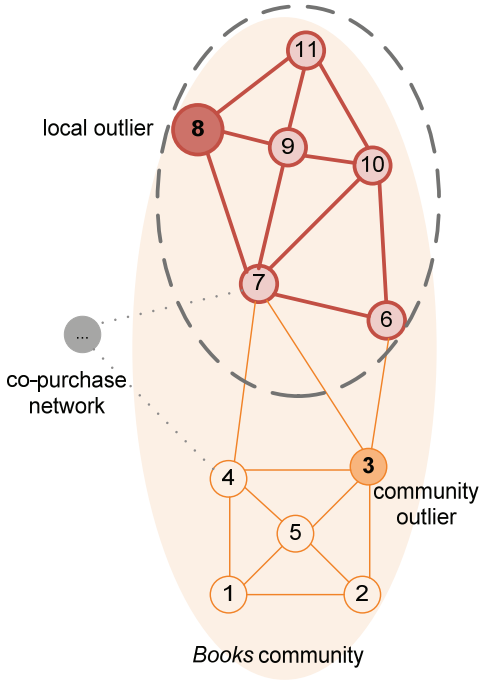


Figure 5.1.: Toy Example of local contextual outlier in an electronic platform

In Figure 5.1 we have illustrated a compact version of this problem setting on an electronic marketplace with both graph and attribute information. *Product 8* is an outlier for the following reason: It has an exceptionally high number of *Reviews*, in contrast to all of its co-purchased *Products 6, 7, 9, 10, and 11*. Although high values in this attribute are normal over the entire database, it is exceptional for this specific context (i.e., set of co-purchased products). Furthermore, *Product 8* belongs to a global graph partition described by products with similar prices (e.g., *Books* community). However, only the local context selection of *Product 8* (subgraph: {6, 7, 9, 10, 11}, {*Reviews, Price*}) in

both graph and attributes reveals the local deviation of this outlier. In this chapter, we focus on the selection of such local contexts for each node in order to detect contextual outliers.

Traditional contextual outlier mining [SWJR07, BKNS00] only consider the numeric attribute space neglecting the graph structure. On the other hand, current techniques [MCRE09, GFBS10, MISMB13, ISML+13] combining both graph structure and multiple node attributes are not able to do an individual selection of the graph neighborhood and its relevant attributes for each node. Thus, they are not able to provide local contexts for each node in the database in order to compute accurately the outlierness of an object w.r.t. its own neighborhood. In the search for local contexts, one open challenge is the increasing number of attributes in today's applications. Not all the attributes show dependencies with the graph structure and they have almost random values for the residual attributes [ISML+13]. In particular, only some attributes are relevant for a certain graph cluster [MCRE09, GFBS10, ATMF12]. A core problem is that, even if one has selected a specific graph neighborhood, some irrelevant attributes will scatter the full attribute space [BGRS99], and outlier detection is hindered [MISMB13, ISML+13]. The outlierness measure is a further challenge, as both the graph structure as well as numeric deviation in the attribute space have to be considered. The definition of a scoring function poses challenges regarding the unification of these two properties.

We propose *ConOut*, the first statistical attribute selection, which enables the detection of contextual outliers in graphs with multiple node attributes. Our context selection allows a good distinction of outliers w.r.t. both the selected attributes and the local graph neighborhood. With this model we select relevant attributes that show similar attribute values for the selected graph neighborhood. Thus, we can discern outliers from regular objects even in the presence of many irrelevant attributes. This context selection allows the detection of local outliers that would not be detectable considering the entire graph or a global partition. Finally, we measure outlierness of each object by unifying structural and numeric information. In our experimental evaluation, we compare against several baselines [XYFS07, BKNS00, AY01] on either graph or numeric data and recent competitors using both graph and numeric data [GLF+10, MISMB13, ISML+13]. Finally, the results highlight the benefit of context selection in graphs with multiple numeric node attributes.

5.2. Comparison to Related Work

We discuss outlier mining in (1) vector data, (2) graphs, (3) combinations of both. In Table 5.1, we summarize existing outlier mining techniques grouped by their paradigms. Traditional approaches for outlier mining have focused either on relational [VW09, RL87, KN98, BKNS00, AY01, LK05, MSS11, KMB12] or graph data [NC03, EH07, Cha04, XYFS07, SHH+10, AMF10]. In contrast to approaches considering all the

attributes [CPF06, DLMR11, GLF+10], this work focuses on outlier nodes and on a context selection including an attribute selection.

Although general approaches have been proposed as pre-processing step for the selection of relevant attributes [TL12, ISML+13], they do not consider a local selection of the attributes w.r.t. the node neighborhood. The main drawback of approaches based on the subspace selection paradigm [MAIS+12, MISMB13, ISML+13] is their time complexity, as there is an exponential number of possible subspaces. Overall, it remains an open issue to efficiently select a local projection of the relevant attributes w.r.t. the individual graph neighborhood for each node.

	Data Type		Context Selection	
	Graph	**Attributes**	**Graph Perspective**	**Attribute Perspective**
Traditional Approaches				
relational data (full dimensional) [VW09, RL87, KN98, BKNS00]	✗	✓	✗	✗
relational data (multiple views) [AY01, LK05, MSS11, KMB12]	✗	✓	✗	multiple
anomalous edges [Cha04]	✓	✗	global	✗
anomalous subgraphs [NC03, EH07]	✓	✗	local	✗
outlier nodes as by-product clustering [XYFS07, SHH+10]	✓	✗	local	✗
node neighborhood analysis [AMF10]	✓	✗	local	✗
anomalous labelled subgraphs [DLMR11]	✓	✓	local	✗
Attributed Graphs				
semi-supervised [CPF06]	✓	✓	global	✗
community outlier mining [GLF+10]	✓	✓	global	✗
subspace cluster analysis *GOutRank* [Chapter 7]	✓	✓	local	multiple
global subspace selection *ConSub* [Chapter 6]	✓	✓	global	multiple
ConOut	✓	✓	**local**	**single**

Table 5.1.: Overview of outlier mining techniques

5.3. Problem Overview

Overall in this chapter, we use the notation described in Chapter 2. We focus on outlier ranking that provides a sorting of all objects o given in a database DB. In particular, outliers become the highest values (outlierness) in the ranking provided by *ConOut*. Thus, they appear on top of the ranking. An accurate calculation of the deviation w.r.t. an object's neighborhood requires a local context selection, i.e., selection of the relevant attributes, and a ranking function to compute the outlierness. In the following, we describe the challenges and introduce the most important definitions of this work.

Local Context Selection Local approaches for outlier ranking have shown to improve the quality w.r.t. global approaches as they are able to compare carefully each object with its own neighborhood. Thus, they are able to detect hidden outliers which cannot be detected if one considers the whole database [BKNS00, AMF10]. However, these traditional local approaches have focused on vector [BKNS00] or graph data [AMF10]. Thus, they are not able to detect *community outliers* that appear in combination of

the graph structure and the node attributes. For example, *Product 3* shown in Figure 5.1 is such a *community outlier*. It belongs to a community of related products (e.g., *Books*) with similar price values and it shows highly deviating values in the attribute *price*. Only a context selection combining both the graph structure with node attributes enables the detection of such outliers [GLF+10].

However, with more and more attributes describing these nodes in such attributed graphs, not all the attributes have to depend on the underlying graph structure. Hence, they have almost random values for the residual attributes (e.g., attribute *sales*). This effect hinders the clear distinction of outliers from regular objects as all nodes seem to be outliers if one considers all attributes [New03, ISML+13]. *Product 3* is only deviating w.r.t. the attribute *price*. It is essential for outlier ranking to consider only these relevant attributes for an accurate measurement of the deviation. In order to avoid this, pre-propressing techniques have been proposed for the selection of the relevant attributes [ISML+13]. Nevertheless, to ensure the correlation of the attributes with the entire graph structure is a global perspective of the database which does not allow the local extraction of the relevant attributes for each community.

Following our previous example, related *Books* have similar prices if one only considers this community in a co-purchase network, but this attribute may be not relevant for other communities (e.g., *Hardware* products). To achieve this, one can use graph clustering techniques [MCRE09, GFBS10, ATMF12] in order to exploit local selections of attributes in each community for outlier ranking [MAIS+12, MISMB13]. Overall, these approaches are not able to detect local outliers in graphs with multiple node attributes as they are not able to provide a local context selection for each node. *Product 8* is an example of such a local outlier. It belongs to the global community of *Books* and it also shows a similar price w.r.t. them. However, its own local context consists of more specific products (e.g., *Tolkien's books*) that show not only similar prices but also similar number of reviews. Only such a local context selection allows us to detect this product as a local outlier. It highly deviates in a relevant attribute (e.g., number of *reviews*) of its own neighborhood.

We define this as *local context* of an object o which consists of a tuple formed by a selected subgraph and its relevant attribute projection:

Definition 5.1:

Local Context
Given an object o, we define its local context as the tuple $(C(o), R(o))$ consisting of:

- the graph context $C(o) = (V', E')$, $V' \subset V$ and $E' \subset E$
- its relevant attribute projection $R(o) \subseteq D$.

Please note, that for our problem setting we do not consider isolated nodes. This is because they do not have a neighborhood regarding the graph structure. Thus, the set of relevant attributes based on their local graph neighborhood and their outlierness w.r.t. their local neighborhood cannot be determined. Given Definition 5.1, two main questions remain: (1) how to define the graph context showing similar graph structure between the nodes and (2) how to model the relevance of an attribute given this graph context. We address these questions in Section 5.4. Based on this careful selection of a local context, the ranking function is able to compute accurately the deviation of each object w.r.t. its neighborhood.

Context based Ranking Traditional scoring functions in the vector space [BKNS00, Agg13] are only based on the object attributes $\vec{o}$, while graph methods [AMF10, Cha04] consider only the graph structure $G = (V, E)$ for the scoring function. In contrast to these traditional rankings, we propose a score that incorporates information of both resources based on a previous local context selection. The vector space provide essential information about the deviation of an object regarding the attribute values. On the other hand, the graph structure can enrich this with valuable information about the affinity between the objects as observed in several studies [MSLC01, CNM04]. A strong connected subgraph of nodes is an evidence that they share some similarities in contrast to isolated nodes that can be the result of a casual relation. Thus, an object showing high deviation in a selected set of attributes within a highly connected subgraph should be ranked first in the result compared to an object low connected w.r.t. its local neighborhood. For this reason, the score has to integrate the information from the deviation within the relevant attributes w.r.t. the connections in its local context. This score gives way to new challenges, as one has to unify the information from both components defined in Definition 5.1: the deviation in the relevant attribute values and the connections within the graph context. We give more details on an instantiation of such score in Section 5.4.

5.4. ConOut Model

Our general idea is to measure locally the outlierness of each object in a projection of the given attributes. Both outlierness measure and projection are determined within the local graph neighborhood of each object. In contrast to other graph mining approaches, we do not consider a global partitioning of nodes. This is because we aim to compute accurate ranking values w.r.t. the local neighborhood of each node. For each node neighborhood, our approach selects carefully only the subset of attributes showing similar attribute values. Hence, each object determines its own local neighborhood in conjunction with its relevant attributes. This local context selection for each node ensures a high contrast in this projection between an outlier and its neighbors, that serves as a basis for computing the deviation. In the following, we describe our

statistical selection of attributes in local graph neighborhoods and our novel outlierness measure.

Local Context Selection

In the following, we explain the local context selection of each object o formed by its graph context $C(o)$ and its relevant attribute projection $R(o)$ (cf. Definition 5.1).

Graph Context For each node o we select a subgraph $C(o) \subset V$. It represents its local context, which shows high similarity in the graph structure between nodes belonging to this context. Intuitively a context $C(o)$ has the following property:

$$\forall p, q \in C(o): \quad p \text{ is structurally similar to } q$$

As graph similarity, we rely on the shared nearest neighborhood (SNN) [XYFS07, SHH+10]. Based on this similarity we define formally the graph context $C(o)$.

Definition 5.2:

Graph Context $C(v)$
Given two nodes $v, p \in V$ and a threshold $\varepsilon \in [0, 1]$, the structural similarity is defined as:

$$sim(v, p) = \frac{|Adj(v) \cap Adj(p)|}{\sqrt{(|Adj(v)|) \cdot (|Adj(p)|)}}$$

where $Adj(v) = \{p \in V \mid \exists (v, p) \in E\} \cup \{v\}$. It forms the basis for the transitive closure of similar nodes in the graph context $C(o)$, as defined by:

$$C(v, \varepsilon) = \{p \in V | \quad \exists q_1, \ldots, q_k \in DB, \; sim(q_i, q_{i+1}) \geq \varepsilon \text{ with } v = q_1 \text{ and } p = q_k\}$$

Overall, we define the context of an object o as the reflexive transitive closure of adjacent nodes with high similarity. It restricts the object neighborhood by a similarity threshold ε, which controls the structural similarity of the context. This selection of the local neighborhood is only a first step in the context selection and it can be also achieved by other local graph context definitions (e.g., extensions of local PageRank [ACL06]). Outliers show up if one focuses on a context of nodes which share common properties, both in structure and in attribute values. Hence, this selection of the local neighborhood is only a first step in the context selection. Further restrictions are defined by the attribute context.

Relevant Attribute Selection In addition to the graph context $C(o)$, we require a subset of the attributes $R(o) \subseteq D$ where the attributes show similar values. For many

attributes the values show almost random distribution with high variance. These scattered attributes (i.e., showing high variance) are not relevant for the selected graph context. We propose a statistical test to exclude such irrelevant and scattered attributes for each individual object in the database. The idea is to include only attributes that show significantly lower variance in $C(o)$ than the overall data distribution.

Definition 5.3:

Attribute Context $R(o)$

$$R(o) = \{d_i \in D \mid \begin{array}{l} d_i \text{ has significantly lower variance} \\ \text{in } C(o) \text{ than the overall database} \end{array} \}$$

As basic properties we have to compute the mean $\mu_i(o)$ and variance $\sigma_i^2(o)$ of a given graph context $C(o)$, as follows:

$$\mu_i(o) = \sum_{p \in C(o)} \frac{\alpha_i(p)}{|C(o)|} \qquad \sigma_i^2(o) = \frac{\sum_{p \in C(o)} (\alpha_i(p) - \mu_i)^2}{|C(o)| - 1}$$

Similarly we compute the overall mean $\overline{\mu_i}$ and variance $\overline{\sigma_i^2}$ for attribute d_i in the entire database. Both the local distribution and the global distribution are compared to each other.

Definition 5.4:

Attribute Context Test
For the global variance $\overline{\sigma_i^2}$ and the local variance $\sigma_i^2(o)$ in context $C(o)$ we define hypotheses H_0 and H_1:

$$\begin{array}{ll} H_0: & C(o) \text{ with similar distribution to all nodes in } V, \text{ i.e., } \sigma_i^2(o) = \overline{\sigma_i^2} \\ H_1: & C(o) \text{ with individual distribution, i.e., } \sigma_i^2(o) < \overline{\sigma_i^2} \end{array}$$

ensuring a significance level:

$$P(H_0 \text{ is rejected } | H_0 = TRUE) \leq \alpha$$

Our test is based on a statistical significance test aiming at reducing the probability that an irrelevant attribute passes into the set of relevant attributes. We test against the null hypothesis that objects are distributed with the same local and global distribution, i.e., $\sigma_i^2(o) = \overline{\sigma_i^2}$. We expect a relevant attribute to show significantly lower variance in a local context $C(o)$ when compared to the entire database. This means that the structural context has selected a subgraph with very similar attribute values in d_i. We exclude scattered attributes that do not fulfill this requirement. Furthermore, by setting

a very low significance value $\alpha = 0.05$, we ensure that irrelevant attributes pass the test with a very low probability.

Depending on the data characteristics, different statistical tests can be applied for our novel attribute selection in graph contexts. In this work, we examine two different statistical tests and evaluate them in Section 5.6.

First, we use the F-Test as a statistical tool to analyze two Gaussian distributions by the comparison of their variances [Ree01]. The F-test derives the threshold required for rejecting H_0 out of the degrees of freedom, i.e., the size of the context and the size of the database. As test statistic, this test uses the quotient of the two variances observed. Formally,

$$F = \frac{\overline{\sigma_i^2}}{\sigma_i^2(o)}$$

is the observed test statistic and $F_{k1,k2}$ is the critical value of a F-distribution with the degrees of freedom: $k1 = |V| - 1$ and $k2 = |C(o)| - 1$. H_0 is rejected when $P(F_{k1,k2} \geq F)$ is under the significance level α. The F-Test ensures that $R(o)$ contains only attributes A_i with low variance in $C(o)$. In particular, we limit the probability of having an attribute with high variance in $R(o)$ by α. Let us illustrate this test with our toy example in Figure 5.1 and *Product 8* with its local context $C(o) = \{6, 7, 8, 9, 10, 11\}$. Testing attribute *sales* means to check if the local variance is lower than the variance of the entire database (e.g., the entire co-purchase network with size $|V| = 36$). With $P(F_{35,5} \geq 0.7) = 0.76$, this attribute is clearly above the significance threshold α and is considered irrelevant. In contrast to this, *price* obviously shows low local variance in $C(o)$. In particular, $P(F_{35,5} \geq 5.2) = 0,01$. In general, attributes with *p-values* under the significance level will be selected as relevant attributes.

Second, we also analyze our approach with the two sample Kolmogorov Smirnov test that does not require any underlying assumption of the data distribution [Ste70]. This test does not only consider variations in the variance to determine if two samples significantly differ, but it also considers mean variations. To achieve this, it considers the absolute distance between two empirical distribution functions, i.e., the empirical distribution functions of attribute d_i considering the whole database F_V and the individual context $F_{C(o)}$. The calculated test statistic is defined as the maximal difference:

$$D = \sup_{x_{d_i}} |F_{DB}(x_{d_i}) - F_{C(o)}(x_{d_i})|$$

If the calculated test statistic D is larger than the critical value $K_{|V|,|C(o)|}$, the null hypothesis is rejected with a significance level α with $P(K \geq D_{V,C(o)}) < \alpha$.

In general, d_i is only relevant when the H_0 hypothesis is rejected. Without a selection of the attributes by a statistical test, scores are blurred by the high variance of irrelevant attributes. So, it ensures a high contrast between outliers and regular objects. This

provides the basic means for the outlierness scores in the following Section. Regarding the use of a statistical test, other tests for the comparison of samples can be found in the literature. Some of them are non-parametric and aim to be more robust w.r.t. the presence of outliers (e.g., Wilcoxon signed-rank test [Wil45]). Additionally, existing tests can also be modified to avoid an impact of the outliers on the test without assuming high homogeneity in the context (e.g., using the median instead of the mean to compute the variance). However, the focus of this work is not to analyze or improve the statistical tests for the selection of the attributes. We have only selected two well-established representatives to evaluate our framework. We do not expect any difficulty when instantiating the statistical test used with any other statistical test possible.

Context Based Outlier Ranking

As an essential property of our scoring, we measure outlierness locally for each object. We ensure an adaptive scoring in local contexts and aim at the local deviation of each object. So, we follow the well-established paradigm of local outlier ranking [BKNS00, MSS11]. Based on this general idea of local outliers we compare each object with its local neighborhood and measure its outlierness locally in contrast to this set of objects. Furthermore, one intrinsic challenge behind this intuitive outlier notion is that one has to ensure that outlier scores remain comparable. Using one scoring function for different subgraphs and different attribute sets will be biased (e.g., w.r.t. the context size). Hence, we have to normalize the score accordingly for each object. We propose such a normalized and unified score in the following. Before we introduce our novel contextual score to integrate the information of both node attributes and graph structure, we present first the measure to extract the deviation of an object in the vector space and the measure to calculate the edge density of an object w.r.t. its neighborhood.

Attribute-Based Score As attribute-based score we consider the deviation of each selected attribute $d_i \in R(o)$. We measure the deviation of an object o w.r.t. the local mean $\mu_i(o)$ in its graph context. We formalize the attribute-based deviation of a node in the following definition.

Definition 5.5:

Local Attribute Deviation $LAD(o)$
Given an object o and its relevant attributes $R(o)$, we define its *LAD* as:

$$LAD(o) = \frac{\sqrt{\sum_{d_i \in R(o)} \frac{(\alpha_i(o) - \mu_i(o))^2}{{\sigma_i}^2(o)}}}{|R(o)|}$$

where $\mu_i(o)$ and $\sigma_i(o)$ are the mean and standard deviation of attribute d_i in the graph context $C(o)$.

Regular objects with no deviation in their attribute values are clearly separated from outliers, i.e., a regular object o has a low deviation ($LAD(o) \approx 0$). We apply this definition within the local context of each node and we do not apply it for the entire database. Thus, we assume a normal distribution within the local contexts representing the inliers, and outliers are assumed to deviate from the mean of the distribution. These objects are regular observations and should end up at the bottom of our ranking. In contrast, highly deviating objects that are observed will be scored with high outlierness ($1 < LAD(o) < \infty$). Comparability is achieved by our normalization: It is neither biased by the number of selected attributes $|R(o)|$ nor by the different local densities resulting in highly different variance values ${\sigma_i}^2(o)$.

Graph-Based Score Second, we define the structural properties that compare the object connections to the ones of its local graph context. We follow the local adaptation in the attribute-based score and extend this idea to local graph densities.

Definition 5.6:

Local Graph Density $LGD(o)$
Given an object o and its graph context $C(o)$, we define its *LGD* as:

$$LGD(o) = \frac{con(o)}{\frac{\sum_{p \in C(o)} con(p)}{|C(o)|}}$$

with the average connectivity $con(p)$ at node p as:

$$con(p) = \frac{1}{|Adj(p)| - 1} \cdot \sum_{(p,q) \in E} sim(p, q)$$

With $con(p)$ we describe the average connectivity to nodes belonging to the same context. It is based on the same notion of SNN as the one used in our graph context definition. For comparability (i.e., outlier scores in different contexts) we normalize

connectivity of each object w.r.t. the connectivity of its neighborhood. For the local node density, we compare the connectivity of o with the average connectivity in its graph context $C(o)$. Low density ($0 < LGD(o) \leq 1$) highlights a node with only low connectivity in comparison to its local graph context. In these cases, o should get lower weights as a contextual outlier and should be ranked lower in comparison to highly connected nodes ($1 < LGD(o) < \infty$). With $LGD(o) = 1$, we have a baseline for the structural connectivity. In such cases, we consider only the attribute deviation.

Contextual Outlier Score Finally, we integrate graph-based and attribute-based measures to form a unified scoring function, which aims at contextual outliers combining the information from graph structure and attribute values. Our score aims to consider both attribute and graph properties: A local outlier may have a small attribute deviation from a densely clustered neighborhood, or it may have high deviation from a weakly connected neighborhood. Both cases get a high outlierness score. Overall, our outlier score aims to detect local deviation considering both graph and attribute properties.

Definition 5.7:

Contextual Outlier Score

Given an object o with $|C(o)| \geq 2$ and $|R(o)| > 0$ we define its contextual outlier score as:

$$score(o) = LGD(o) \cdot LAD(o)$$

Please note that the product $LGD(o) \cdot LAD(o)$ achieves better outlier detection than its individual factors $LGD(o)$ and $LAD(o)$. It covers several cases of contextual outliers w.r.t. both structural and attribute information that cannot be detected by the individual measures in one of the two information sources. In addition to this, our contextual outlier score exploits the zero property of the multiplication ensuring that regular objects (e.g., objects $LAD = 0$) appear at the bottom of the ranking. In the following, we discuss some of these contextual outlier properties here, and show an empirical comparison to the individual measures and other aggregation functions such as minimum, maximum, and sum in Section 5.6.

$LGD > 1$ & $LAD > 1$
Strong structural connections and high deviation of attributes in this graph context is the most prominent case of a contextual outlier. Such an outlier will be scored extremely high. It shows high attribute deviation although the structural similarity gives way to the expectation of very similar attribute values.
$LGD = 1$ & $LAD > 1$
Average connectivity (similar to its local neighborhood) and high attribute deviation are scored with high outlierness as there is a graph context. However, attribute values are highly deviating from the residual objects in the context.
$LGD \approx 0$ & $LAD > 1$

Low structural density is an indicator for a weak graph context and lowers the overall score of the object.
$LGD \approx 0 \;\&\; LAD \approx 0$
Lower attribute deviation and lower structural similarity is the other extreme case. In such cases there is no indication for a contextual outlier at all. These objects will be ranked last.

We also include the special case with those objects being hubs in the graph. These nodes belong to multiple contexts as they do not have high structural similarity to a single graph neighborhood and share different properties with different communities. In these cases, we score based on their adjacent neighbors and all relevant attributes of their neighbors, i.e., $C(o) = Adj(o)$ and $R(o) = \bigcup_{p \in Adj(o)} R(p)$. Hence, scoring is simply the average deviation from the neighboring contexts.

Summarizing the *ConOut* model, we have proposed a local context definition, a statistical selection of relevant attributes, and a scoring function for contextual outliers. Based on this formal model, we will sketch the algorithmic computation in the following section and examine the quality enhancement for outlier detection in Section 5.6.

5.5. Algorithm

In this section, we describe the *ConOut* algorithm. It computes the outlierness of each node in three steps: (1) compute the local graph context, (2) select its relevant attributes, and (3) compute the local outlierness. Finally, all nodes are sorted by their scores.

As parameter, we require only the threshold ε that states how similar objects have to be in the graph structure. We discuss the choice of this parameter in Section 5.6. A naive algorithm would compute the context of each node individually. Such an approach would not scale for large graphs as it is quadratic with the number of nodes. For this reason, we propose a more efficient algorithm to solve this problem exploiting a property of the similarity measure.

We iterate over each node $o \in V$ that has not yet assigned a local context to (Line 3). In the first step, nodes v adjacent to o, which satisfy the structural similarity, are inserted into a queue. As the structural similarity is reflexive ($\forall\, v, o \in V\; sim(o, v) = sim(v, o)$), all nodes fulfilling this condition have the same context:

$$\forall\, v \in C(o) \Rightarrow C(o) = C(v) \text{ (Line 9)}$$

For each of these nodes, we recursively expand the local context with its adjacent nodes until no further nodes can be added into its context and we mark them as visited in the boolean vector (Lines 4-13). In the second step, we compare the distribution of attribute values in the local context to the distribution in the entire database. A statistical test for this comparison is applied to each attribute (Lines 14-19). Finally, we compute the

outlierness of each object based on its local context and its relevant attributes (Lines 20-26).

Algorithm 5 Local Context Selection *ConOut*

Input: $G = (V, E, \alpha)$, and parameter ε
Output: Ranking of all $o \in V$

```
1:
2:  Initialize boolean vector context for all o ∈ V: false
3:  for all o ∈ V where context[o] = false do
4:      Mark context[o] as true
5:      insert all {p | (o, p) ∈ E} into queue Q
6:      while (Q ≠ ∅) do
7:          if p is similar then        ▷ Graph similarity for graph context (cf. Def. 5.1)
8:              Insert p into C(o)
9:              Mark context[p] as true          ▷ ∀ v, o ∈ V sim(o, v) = sim(v, o)
10:             Insert non-visited q with (p, q) ∈ E into Q
11:         end if
12:         Label p as visited and remove p from Q
13:     end while
14:     for all d_i ∈ D do
15:         Compare distribution of d_i in C(o) with the distribution of d_i in DB
16:         if d_i relevant then                  ▷ Statistical test (cf. Def. 5.3)
17:             Add d_i to R(o)
18:         end if
19:     end for
20:     for all v ∈ C(o) do
21:         if (|C(o)| ≥ 2) ∧ (|R(o)| > 0) then
22:             Compute score based on C(o), R(o) ▷ Ranking function (cf. Def. 5.7)
23:         else
24:             Compute score based on Adj(v), D
25:         end if
26:     end for
27: end for
28: Sort all o ∈ V by score(o)
```

Complexity Analysis Overall we have to iterate over all objects in our database $|V|$. In the first step, we access the graph by means of an adjacency list for each node. This has a cost proportional to the degree of each node. Thus, the cost is linear with the number of edges $|E|$ for each iteration (Line 5-13). In the worst case, when the whole graph represents the local context, it is $|V| + |E|$. In this case, all nodes are marked as visited in the first iteration of the main loop ($context[o] = true$), and the algorithm

iterates only over the boolean vector without searching for new contexts. This is a rare case for a complete graph, or for a parametrization that is too permissive (e.g., $\varepsilon = 0$). In the second step, the computational cost is linear with the number of dimensions d and the number of nodes of the context. Each attribute is tested once for each local context.

To compute the outlierness, we iterate over each node of the local context, and the runtime of scoring is constant in each iteration, since we have pre-computed all values required for the scoring function. In the worst case, the local context is the whole graph, and we must compute the ranking for each node. Finally, the nodes are sorted by the score values. Overall, the runtime of *ConOut* depends on the local context selection, the statistical test of relevant attributes and the sorting of the nodes. Thus, the worst case cost is $\mathcal{O}(|E| + d + |V| \cdot log(|V|))$.

5.6. Experiments

We evaluate the quality, runtime, parametrization and different scoring functions on synthetic and real world datasets. We compare *ConOut* to several competitors:

1. The clustering algorithm SCAN [XYFS07], which considers only the graph structure. It allows the detection of structural outliers.
2. Different paradigms considering only vector data: the full dimensional approach LOF [BKNS00] and the subspace outlier approach SOF [AY01] that analyzes the relevant subspaces in order to exclude irrelevant attributes that hinder outlier detection.
3. As full dimensional approach for attributed graphs, the community outlier mining algorithm CODA [GLF+10], which considers all the node attributes and the graph structure.
4. Two related algorithms based on the subspace selection paradigm that combine both resources: (1) outlier ranking on attributed graphs based on subspace cluster analysis *GOutRank* [Chapter 7] and (2) a global subspace selection algorithm as pre-processing step *ConSub* for mining attributed graphs [Chapter 6].

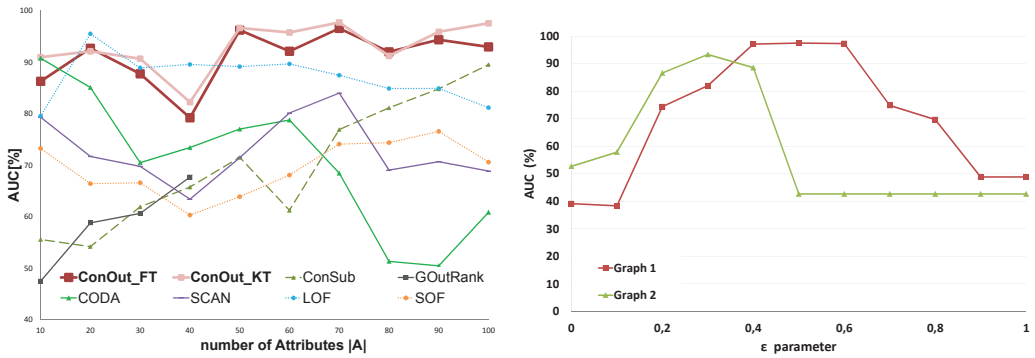
The quality of the obtained outlier ranking has been determined by the *area under the ROC curve* (AUC). For each position in the ranking, we compute the ratio of precision/recall and compute the AUC value as commonly used for the evaluation of outlier rankings [Agg13]. We have implemented all algorithms in Java and performed experiments on an Intel CoreDuo running 1,8 GHz and 4 GB memory. To facilitate comparability of our experiments, we provide code, datasets, and parameter settings online on our project website[2].

[2] `http://www.ipd.kit.edu/~muellere/ConOut/`

Synthetic Data

Generation of the Synthetic DataSet We have based our generator on the graph generator described in [XYFS07]. It allows to generate structural outliers as well as hubs connected to multiple clusters. We have extended this generator with numeric node attributes. We generate graph clusters with intra-cluster connectivity of probability P_{in}, and inter-cluster probability of edges P_{out}. In our setup, P_{in} is higher than P_{out}. For each graph cluster, we randomly select $x \in (1, d]$ relevant attributes and choose their attribute values based on a Gaussian distribution. In contrast to this, all other attributes get values out of a uniform random distribution. The attribute values for hubs and structural outliers are chosen depending on the distributions of their direct neighborhood. In addition to hubs and structural outliers, we insert context outliers that are hard to identify. They are generated by selecting clustered nodes and manipulating a random number of their relevant attribute values. As ground truth for each object, we have marked the outliers generated with a respective label.

Experiment Configuration We generate different graphs with an increasing number of attributes. For each dimensionality, we generate three graphs to average over random effects in the generation process. Additionally, we generate one-dimensional datasets varying the number of nodes and edges for the runtime evaluation. On each of these datasets, we configure the different algorithms as follows: For the algorithm *CODA*, we set the exact value of the outlier ratio and the number of clusters since these parameters are known for each dataset generated. Additionally, we used 10 different initializations for *CODA* and used only the best result. Regarding the unknown parameters for the other algorithms, we try several parameter combinations. Finally, we use the results of the parameter combination showing the best quality results. Detailed information about the exact ranges of each parameter can be found in our website. In particular, *ConOut* achieves the best results with values of ϵ between 0.5 and 0.7.



(a) *Quality w.r.t. number of attributes* (b) *Quality w.r.t. parameter ϵ*

Figure 5.2.: Evaluation of the quality on synthetic data

Quality evaluation First, we evaluate the outlier detection quality w.r.t. the number of

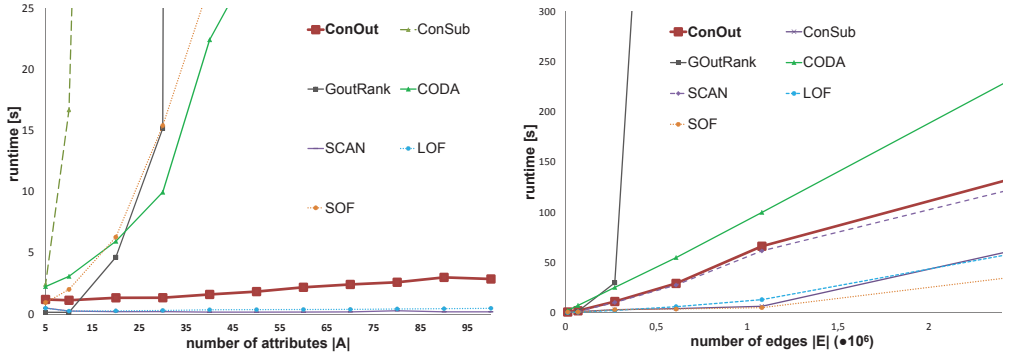
node attributes. We depict average AUC values for all competitors in Figure 5.2(a). For each algorithm, we have tried to find optimal parameter settings. In particular, for CODA we have tested 10 different initializations and have used only the best result. In addition, we evaluate two statistical tests for our approach. Experiments show that the selection of relevant attributes using the Kolgomorov-Smirnov test (*ConOut_KT*) achieves better results than the F-Test (*ConOut_FT*). This is because it is more robust by mean variations w.r.t. the global distribution. Not depending on this choice of statistical tests, our approach outperforms all competitors. It is the only algorithm that can detect the context outliers hidden in the graph. Due to our statistical selection of relevant attributes, we achieve high quality even for a large number of attributes. In contrast, traditional competitors tend to miss some hidden outliers as they only consider one data source (graph structure (SCAN) or vector data (LOF, SOF)).

A detailed analysis of the detected outliers in Figure 5.2(a) reveals that SCAN is performing well on structural outliers having deviating attribute values. Regarding the local approach *LOF*, it neglects the information of the graph structure and it does also not select the relevant attributes for each neighborhood. Thus, its performance decreases with increasing dimensionality. Similar to this, CODA uses all the given attributes and fails because of the irrelevant attributes. Although *ConSub* selects the relevant attributes for the graph structure, this selection is done globally (for the whole graph). Thus, it is not able to select locally the relevant attributes for each neighborhood. Finally, the ranking functions of *GOutRank* heavily depend on the underlying subspace cluster definition and do not consider the local neighborhood of each node. Overall, we have shown that *ConOut* achieves significant quality improvement in the detection of context outliers.

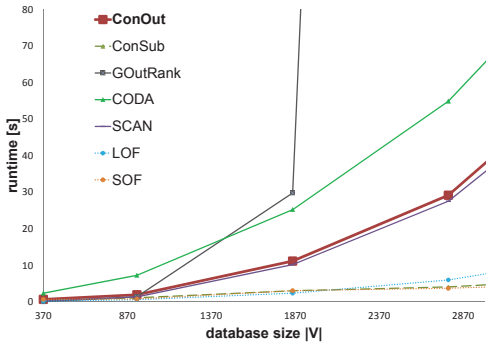
Runtime Evaluation As explained in Section 5.5, the runtime of our algorithm depends on the database size $|V|$, number of edges $|E|$, and the number of attributes $|A|$. In Figure 5.3, we depict scalability results w.r.t. all of these properties in comparison to our competitors. Figure 5.3(a) shows the scalability with increasing number of attributes. The runtime scalability is slightly higher in comparison to traditional approaches due to the combination of both information sources (graph structure and vector data). We deem this tolerable due to the significant quality improvements shown in Figure 5.2(a). Compared to CODA, we show better scalability, as its runtime is quadratic in the number of attributes, due to matrix operations for the multi-variate likelihood function of the underlying Gaussian distribution.

Additionally, approaches based on subspace selection show much higher runtimes w.r.t. the number of attributes in contrast to the linear time complexity of *ConOut*. In particular, *GOutRank* does not scale with high dimensionality (up to 30 attributes). Furthermore, we analyze runtimes w.r.t. the database size and the number of edges in Figure 5.3(b) and Figure 5.3(c). In contrast to our approach, CODA and *GOutRank* do not scale with dense graphs over 2.5 million of edges as shown in Figure 5.3(b). Overall, *ConOut* scales well with increasing graph size ($|V|$, $|E|$, and $|A|$). Although CODA,

GOutRank and *ConSub* consider both graph and attribute information, *ConOut* achieves both better quality and runtime performance.



(a) *number of attributes* (b) *number of edges,* $d = 1$



(c) *number of nodes,* $d = 1$

Figure 5.3.: Runtime scalability w.r.t. $|D|$,$|E|$, *and* $|V|$ *on synthetic data*

Parameters *ConOut* uses the parameter ε to specify the local context of each node depending on its connectivity. To evaluate the sensitivity of our parameter we run experiments with different density characteristics (i.e., highly connected *Graph 1* and weaker connected *Graph 2*). Figure 5.2(b) shows the AUC quality measure w.r.t. the value of ε. We see that parametrization is robust for a range of top quality results, and there is the expected shift of optima w.r.t. the underlying graph density. Only extreme cases show significant decrease in quality: On the one hand, if the value of ε is too permissive (e.g., values between $0 \ldots 0.2$), more nodes are part of the local context, and *ConOut* is hindered in its selection of relevant attributes in this large context. On the other hand, a restrictive setting of ε (e.g., $0.5 \ldots 1$) causes very small contexts in which no outliers can be found.

Ranking Functions The scoring function of *ConOut* unifies the information from the local graph density (LGD) with the attribute deviation (LAD) in order to obtain accurate rankings for the contextual outliers. For the quality evaluation of our scoring

Ranking Function	AUC[%]
$LAD \cdot LGD$	**93.3**
$LAD + LGD$	90.44
LAD	90.63
LGD	51.4
$Max(LGD, LAD)$	75.45
$Min(LGD, LAD)$	87.82

Table 5.2.: Quality of the different ranking functions on synthetic data

function (cf. Definition 5.7), we compare it to different baseline aggregation functions (MIN, MAX, SUM) and the raw measures LAD and LGD. We measure the median AUC values obtained by different scoring functions on the 36 synthetic graphs used for the previous quality evaluation. In Figure 5.2 shows the quality results for the different scoring functions. Local graph density (*LGD*) and local attribute deviation (*LAD*) are not able to accurately detect contextual outliers. They fall prey to the information loss as they use only one of the information sources. Aggregation functions such as MAX and *MIN* use both sources. However, they are dominated by one of the measures. The score is not able to make a clear distinction of contextual outliers. For example, two nodes with high local graph densities can have the same score although the attribute deviation is different for each node. The best quality results for contextual outliers are obtained by sum and product which combine both values. However, due to the design of LAD and LGD (cf. Section 5.4), we achieve best results by weighting LAD with a LGD factor. Our proposed scoring function shows overall highest quality results in comparison with all other scores.

Real World Data

We use two networks from different domains to evaluate our approach on real world datasets. First, we perform a thorough evaluation of our approach in a subgraph of the co-purchase Amazon network. On this dataset, we have the ground truth for objective quality assessment from a benchmark explained in Chapter 3. Second, we use the bibliographic repository provided by DBLP for the evaluation of our approach in a larger attributed graph.

Amazon Network

The dataset is a subgraph of the Amazon co-purchase network. In particular, the considered products are *Disney* films. Figure 5.4 shows the *Disney* network with three

outlier examples and their connectivity to the co-purchase network. Additionally, we also provide their *Amazon Standard Identification Number* for manual verification [3]. In this real-world dataset, each object has been labeled manually by high school students, providing us the ground truth (object is an outlier or not) for quality assessment.

Used data	Paradigm	Algorithm	AUC	Runtime	Speedup
attributes	full space	LOF [BKNS00]	56.85	41	0.20
	subspace selection	SOF [AY01]	65.88	825	4
graph	graph clustering	SCAN [XYFS07]	52.68	4	0.02
both	full space	CODA [GLF+10]	50.56	2596	13
	subspace cluster analysis	*GOutRank* [Chapter 7]	**86.86**	**26648**	**134**
	global subspace selection	*ConSub* [Chapter 6]	**81.77**	**8930**	**45**
	context selection	*ConOut*	**81.21**	**199**	**1**

Table 5.3.: AUC[%] values, Runtime[ms] results and ConOut's speedup w.r.t all competitors on the Disney network.

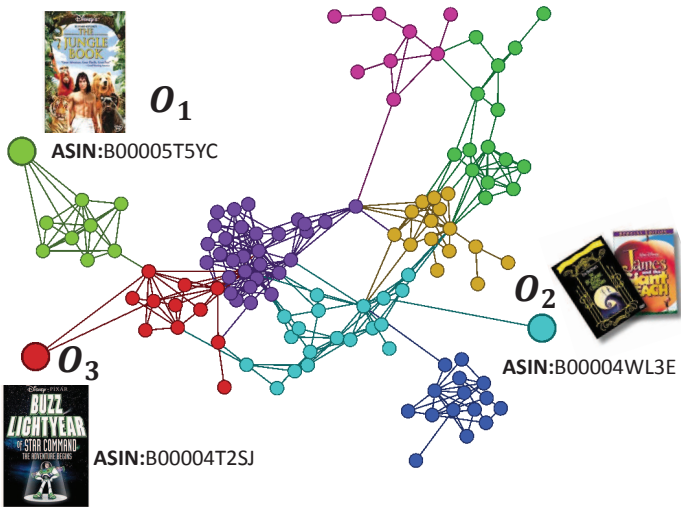


Figure 5.4.: Visualization of 3 hidden outliers on the Disney network and their graph connectivity to their neighborhood

Table 5.3 gives an overview of quality results. Considering only one source of information – only attributes or graph structure – clearly misses some of the outliers. In particular, the full space technique (LOF) is hindered by the high dimensionality of the product features. On the other hand, subspace analysis (SOF) allows the detection of subspace outliers (e.g., O_2), which is a structural outlier found by graph-based techniques (SCAN) as well. However, none of the paradigms is able to reveal contextual outliers such as O_1 and O_3 (cf. Figure 5.4). For example, product O_1 is one of the contextual outliers that corresponds to the overpriced film *The Jungle Book (1994)* of

[3] `http://www.amazon.com/dp/ASIN_VALUE`

Rudyard Kipling's hidden in a group of *Read-Along Disney* films. Its local context is not only characterized by the strong connectivity between the nodes in its graph context, but it is also has following relevant attributes: *number of reviews* and *price private seller*. These outliers can only be detected if graph structure and attributes are combined. CODA considers both data types, but it fails due to the existence of irrelevant attributes. Regarding approaches doing a selection of the attributes, the subspace selection techniques (*GOutRank* and *ConSub*) obtain high quality results, but at much higher runtimes.

In contrast, *ConOut* selects a projection of relevant attributes in the local graph neighborhoods. Thus, it allows to identify highly deviating values. It is the most efficient approach in these graph and attribute contexts. As shown in Table 5.3, *ConOut* shows a 6.5% decrease w.r.t. the best algorithm (*GOutRank*) while being 134 times faster in the runtime. Therefore, it shows the best performance considering both quality and runtime results. It invests some extra runtime compared to traditional approaches for a significant quality improvement. On the other hand, it loses some quality compared to subspace techniques [ISML$^+$13, MISMB13], but is more efficient. Thus, it can be applicable for larger networks.

In the following, we discuss the ranking positions between these outliers considering its graph connectivity. These have been ranked at top positions by *ConOut*. Our approach assigns the fourth position to O_1, which is a local outlier with highly deviating attribute values in a strongly connected neighborhood. Second is object O_3 in the ranking, which is weakly connected to its neighborhood and deviates strongly in the rating values from the other co-purchased products. As our ranking function combines the graph and attribute information (cf. Def. 5.7), O_1 and O_3 have higher scores than the isolated co-purchased product O_2. Regarding the ranking functions, Table 5.4 shows the outlier detection quality for each of them.

Ranking Function	AUC[%]
$LAD \cdot LGD$	**81,21**
$LAD + LGD$	79,66
LAD	78,10
LGD	50,28
$Max(LGD, LAD)$	75,14
$Min(LGD, LAD)$	78,81

Table 5.4.: AUC results for the different ranking functions on the Disney network

The best AUC values are highlighted in bold, and high quality results that are within 2% are not grayed out. Results show that the unification of both information sources: local graph density and the attribute deviation obtains the highest results. However, the proposed ranking function (cf. Def. 5.7) outperforms the others. Overall, the evaluation

on this real data set demonstrates the existence of local outliers hidden in combinations of the graph structure and the attribute values. We have shown that *ConOut* is more effective than existing algorithms and ranks local outliers accurately according to their degree of deviation in attributed graphs.

Algorithm	O_1	O_2	O_3
	ASIN: B00005T5YC	**ASIN:** B00004WL3E	**ASIN:** B00004T2SJ
ConOut	**3**	**8**	**7**
CODA	×	×	×
SCAN	×	√	×
LOF	96	89	57
SOF	77	**2**	86
ConSub	8	**3**	**2**
GOutRank	12	29	20

Table 5.5.: Ranking results from the top ranked outliers on the Disney network

DBLP Network

In our second evaluation we use a larger database. We have extracted a part of the DBLP graph with authors represented as nodes and co-authorship as edges. In addition, we describe each author by a scientific profile containing 46 numeric attributes. These attributes provide information on the author's publication ratio at major database, data mining, artificial intelligence, and statistics conferences. The extracted graph consists of 44808 nodes with 119053 edges. In this graph, *ConOut* achieves a runtime of 11.26 seconds. We discuss the outlierness of individual authors w.r.t. their local context in DBLP. Note that we are not looking for truly extraordinary individuals, e.g., with an exceptionally high number of publications in DBLP. Hidden outliers are more local exceptions, e.g., deviating significantly from their co-authors. Let us discuss some of the top-ranked authors found by ConOut.

Pavan Vatturi: He is a structural outlier as Pavan has published only together with one author. He has also high deviating attribute values. His local context is identical to the one of his advisors' *Weng-Keen Wong*. Weng-Keen has a local context with high publishing ratios in *IJCAI*, *AAAI*, and *ICML*, but Pavan has never published in these conferences in contrast to the other authors in his advisors context (e.g., *Ugur Kuter, Santiago Ontañón, Victor R. Lesser*).

Christoph Heinz is a strong connected node in his context consisting of 18 authors (e.g., *Martin Schneider*, *Jens-Peter Dittrich*, *Dieter Korus*). In this context, authors publish frequently on database conferences (e.g., *VLDB, EDBT*, and several more) but

they have never publish on the *CIKM* conference. In contrast to his context, Christoph has not publish in database conferences, which are relevant for his context, but he is the only one that has published on *CIKM*.

Ina Müller-Gormann belongs to a highly connected local context (31 authors) with several relevant attributes (*SIGMOD, KDD, ICDE, ICDM*, and several more). She has published with many authors (e.g., *Arthur Zimek*, *Hans-Peter Kriegel*, ...) of this context, however, she has a clear deviation in the relevant attributes. She has not published in the relevant conferences of her local context.

All these authors are clearly local outliers. The strong connectivity in the graph structure and the highly deviating attribute values in the relevant attributes of their contexts cause their high ranks. Thus, they would not have been found without the local context selection provided by *ConOut*.

5.7. Summary

In this Chapter, we have proposed *ConOut*, a context selection for outlier ranking in graphs with numeric node attributes. Our approach computes locally graph and attribute contexts for each object in the database. For each context, it selects a set of relevant attributes. Relevance of attributes is measured by a statistical test which compares the local and the global variance of each attribute. Thus, outlier ranking relies on a high contrast between outliers and their local context. Overall, *ConOut* computes a high quality outlier ranking that scales well with the number of attributes. Our thorough evaluation on synthetic and real world data shows that it finds local contexts, in contrast to other approaches.

ConOut balances quality with efficiency when joining attribute information with the graph structure. In contrast to approaches based on subspace selection, the runtimes of *ConOut* are significantly lower. To achieve this, we assume that attributes are independent. We do so to give way to an efficient selection of relevant attributes, in linear time. Efficiency is important when it comes to larger attributed graphs. As future work, we aim to design local efficient approaches without assuming the independence of the attributes.

Our approach focuses on numerical node attributes. Thus, a mixture of attribute types such as binary, categorical, and continues values is not explicitly considered in this work. The statistical test would require additional unification of the relevance measure to be applicable in the presence of such heterogeneity. Finally, we also aim at other graph definitions, e.g., considering edge attributes or directed graphs. Such data provides even more information for data mining, however, it also poses novel challenges regarding attribute selection.

Part III.

Generic Context Selection

6. Congruent Subspaces

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.

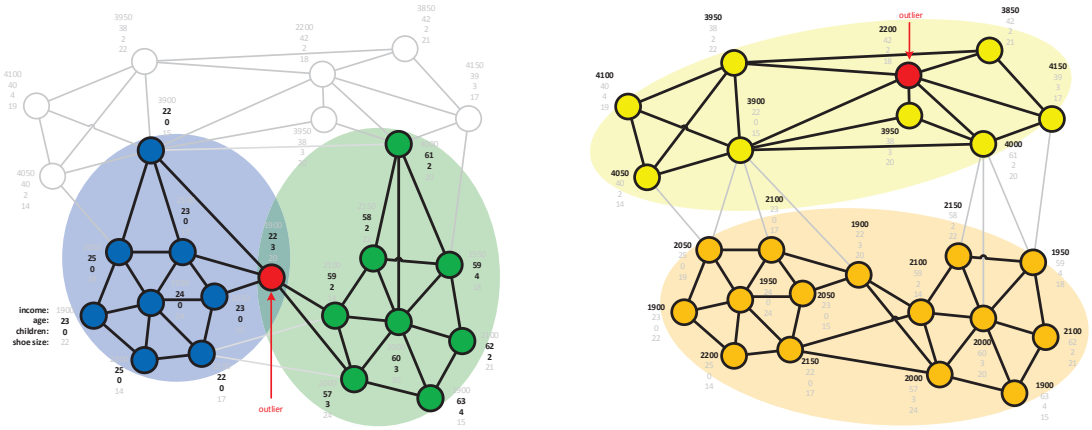
In this Chapter[1], we propose a method for the statistical selection of such congruent subspaces. More specifically, we define a measure which assesses the degree of congruence between a set of attributes and the entire graph. We use it as the core of a statistical test, which congruent subspaces must 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.

6.1. Motivation

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 [STM07, ZCY09, DLMR11, GLF+10] have shown that exploiting existing dependencies is beneficial, e.g., for cluster and outlier detection. However, the techniques proposed in these articles highly rely on this dependency assumption. In particular, *community outlier mining* [GLF+10] is able to detect an outlier node if connected nodes have similar values in all attributes. Such assumptions are known as homophily [MSLC01] 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

[1] This chapter is an extension of the published work in the Proceedings of the IEEE International Conference on Data Mining (ICDM 2013) [ISML+13]

graph structure of social networks [MSLC01]. 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.



(a) *congruent subspace* {*age, children*}

(b) *congruent subspace* {*income*}

Figure 6.1.: Toy example for multiple views of a social network depending on the selected congruent subspace.

Let us illustrate this with a toy example in Figure 6.1. 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 6.1(a)), we observe communities of young persons without children, old people with several children, and a deviating outlier. Considering another subspace (cf. Figure 6.1(b)), we observe a different community/outlier structure w.r.t. income. In this thesis, we call this phenomena *multiple views* of an attributed graph.

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 [STM07, ZCY09, DLMR11, GLF+10].

Regarding outlier mining, one can identify community outliers merely based on these subspace neighborhoods having identified congruent subspaces. In this chapter, 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.

6.2. Comparison to Related Work

We distinguish the method presented in this Chapter 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. Table 6.1 summarizes the main characteristics of the related approaches according to the context selection schemes proposed in Chapter 2.

Specialized subspace techniques Recent methods have observed the lack of dependency in the full attribute space and have proposed local subspace selection schemes for specific subgraphs [ATMF12, LM12, GFBS10, SMJZ12, GBFS13, GBS11]. 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 model-dependent solutions to the problem of subspace selection, but they lack generality and are not designed as pre-processing step for other graph mining models.

	Context Selection		
	Graph Perspective	**Attribute Perspective**	**Generic scheme?**
Full Space			
clustering [STM07, ZCY09, ZCY10, XKW+12, Vie12] outlier mining [GLF+10]	global	✗	✗
Specialized Subspace Techniques			
subspace clustering [GFBS10, GBS11, GBFS13]	local	multiple	✗
subspace outlier *GOutRank* [Chapter 7]	local	multiple	✗
projected clustering [ATMF12, GFRS13]	local	single	✗
local outlier mining *ConOut* [Chapter 5]	local	single	✗
General Feature Selection			
feature selection [TL12]	global	single	✓
ConSub	**global**	**multiple**	✓

Table 6.1.: Comparison of ConSub with related unsupervised approaches for mining attributed graphs according to their context selection schemes

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 [New03]. 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 [TL12]. 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 [TL12] and full space outlier detection [GLF+10].

6.3. Problem Overview

Overall in this chapter, we use the notation described in Chapter 2. In the following, we describe first the challenges and basic definitions for the selection of subspaces on attributed graphs. Then, we introduce the concept of *subspace community outlier* based on the definition previously introduced of congruent subspaces.

Selection Scheme Existing algorithms exploit the dependencies between both graph G and attributes D for knowledge discovery. In particular, they exploit the assumption of homophily [MSLC01] that connected nodes tend to have similar characteristics. This effect is also known as *assortative mixing* [New03]. However, this assumption may not be fulfilled for the full space of attributes D. 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* [New03].

For example, *community outlier mining* needs to capture a group of similar objects w.r.t. the graph structure and the attribute values [GLF+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 [New03] is not enough. We expect outliers or clusters to hide in combinations of attribute values, as has already been shown for multivariate vector data [Agg13, CBK09]. 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 $S \subseteq D$ 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 $S \subseteq D$ where the attribute values are consistent with the graph structure.

Definition 6.1:

Congruent Subspaces
Given an attributed graph $G = (V, E, \alpha)$ and subspace $S \subseteq D$,

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

$\forall\, V' \subseteq V$ with high **mutual similarity** between the attribute values in subspace S:
Subgraph $G' = (V', E', \alpha')$ 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:

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

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 6.1, three main questions remain: (1) how to define mutual similarity of objects within a subgraph (V', E', α') 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 6.4 and describe the algorithm for the selection of CS in Section 6.5.

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 [GLF+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 [GLF+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 D does not follow the assumption that all the attributes are congruent with the graph structure. In case of CODA [GLF+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 [GLF+10], CODA can only deal with dimensions that are correlated with the graph (i.e., in our notion: D is congruent with the graph). However, not all given dimensions are congruent on the graph structure in real world networks [MSLC01, New03]. Furthermore, different subsets of attributes correspond to different community/outlier structures (cf. Figure 6.1). 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 6.2:

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

a node $o \in N$ that shows a **high deviation** 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 6.3:

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 6.3. It is the major difference to traditional outlier scores using the entire set of attributes $score(o, D)$. Please note that other aggregation functions or even ensemble techniques might be of interest as well [MSS11, Agg12]. However, this is research orthogonal to our current work and will not be addressed in this chapter. 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 6.6.

6.4. ConSub Model

In order to assess the congruence of a subspace $S \subseteq D$ 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 $d_j \in S$ in a Monte Carlo processing. For each interval, we consider the subgraph formed by the nodes that have attribute values within these intervals as shown in Figure 6.2. Thus, we ensure similar attribute values within the subgraphs as it is a requirement for congruent subspaces (cf. Definition 6.1). 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 6.4, we introduce the *ConSub* measure for the assessment of congruence. We describe the estimation of the number of expected edges in Section 6.4 and propose a statistical test for the comparison of observed and expected edge counts in Section 6.4.

Congruence Assessment

In *ConSub*, we consider intervals of the attribute values for the retrieval of subgraphs where nodes have similar values. These attribute regions $[low_j, high_j]\ \forall d_j \in S$ restrict the graph structure to subgraphs (cf. Definition 6.4). 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*:

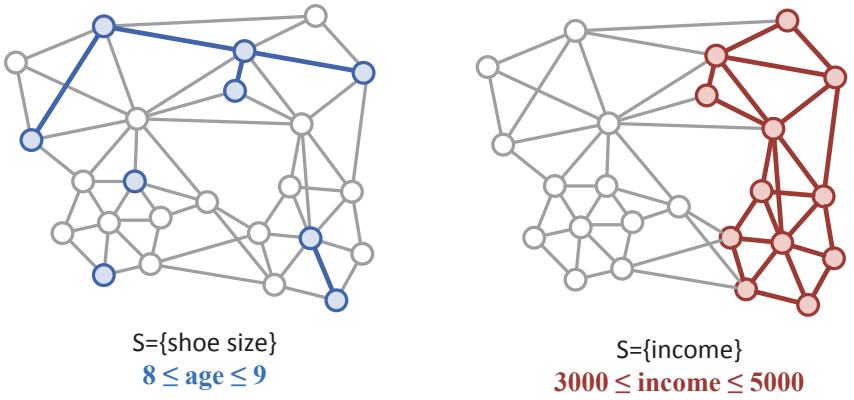
Definition 6.4:

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

$$V_{C,S} = \{o \in V \mid \alpha(o) \in \mathbb{R}^d \wedge \forall d_j \in S : \alpha_j(o) \in I_j\}$$

and

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



(a) *Few number of edges* (b) *High number of edges (congruent)*

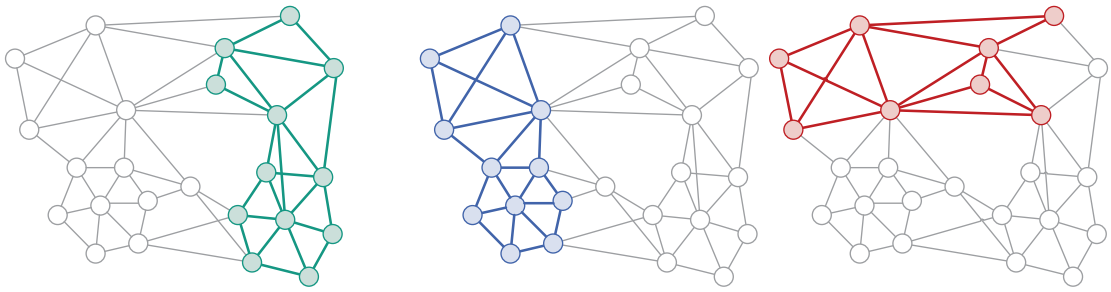
Figure 6.2.: Example of different constraint subgraphs

Continuing our running example from Figure 6.1, we consider the constraint subgraph $G_{\{([3000,5000],income)\},\{income\}}$. We observe an unexpectedly high number of edges $|E_{\{([3000,5000],income)\},\{income\}}|$ since this subgraph is congruent with the given constraints (cf. Figure 6.2(b)).

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 6.4.

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. Figure 6.3 shows an example of such a 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.



(a) $G^1_{\{([3000,5000],S)\},S}$ (b) $G^1_{\{([1000,2000],S)\},S}$ (c) $G^1_{\{([1500,3000],S)\},S}$

Figure 6.3.: Random generation of constraint subgraphs for iterations: $m = \{1, 2, 3\}$, $S = \{income\}$

The definition of our congruency measure is given by:

Definition 6.5:

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 6.4. 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.

Expected Edge Count Estimation

Definition 6.1 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 [ER60, New06]. 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 [New06]. Therefore, we employ a null model that preserves the degree distribution of the given graph.

(3) If a lower dimensional subspace $S'_1 = S \setminus \{d_j\}$ 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* $G_{C \setminus \{(I_j, d_j)\}, S \setminus \{d_j\}}$ where d_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 6.6:

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

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

where $V' = V_{C \setminus \{(I_j, d_j)\}, S \setminus \{d_j\}}$ 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 $deg(G_{C,S}, d_j, o)$ edges to other nodes without a constraint on d_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: $\sum_{p \in V' \setminus \{o\}} deg(G_{C,S}, d_j, p)$. 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 6.7:

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

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

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 [New06] 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.

Statistical Test

In order to find congruent subspaces with significantly more edges than expected (cf. Definition 6.1), 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 6.5 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 6.4. 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 (cf. Figure 6.4).

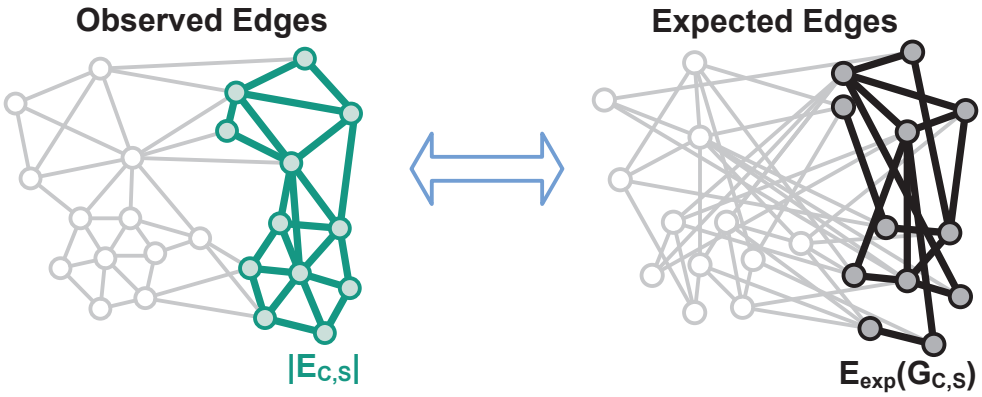


Figure 6.4.: Visual example of observed and expected edge count from a null model

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 (cf. Figure 6.4).

For $ConSub$, we use the Wilcoxon signed-rank test [Wil45], 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 [Wil45]. We use this as the instantiation of $deviation(|E_{C,S}|, E_{exp}(G_{C,S}))$ in each Monte Carlo iteration (cf. Definition 6.5). Thus, the congruence measure is the average *p-value* for all the Monte Carlo iterations.

6.5. Algorithm

The selection of congruent subspaces $S \subseteq D$ is computationally expensive due to the exponential number of subspaces. Overall, one must analyze 2^d 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 [CFZ99, MSS11, KMB12]. This is a heuristic based on the well-known Apriori processing paradigm. Given n-dimensional congruent subspaces $\{S_1, S_2, \ldots\}$, we derive the $(n+1)$-dimensional candidate subspaces with a bottom up procedure similarly to the Apriori algorithm [AIS93]. 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 6.7, 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 6 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 d_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 [KMB12]. 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 $\mathcal{O}(M \cdot (|S| \cdot |V| + |E| + k \cdot \log(k)))$. 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.

Algorithm 6 Selection of Congruent Subspaces *ConSub*

Input: $G = (V, E, \alpha)$, 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 observed and expected edge count
8:              for the current constraint subgraph G_{C,S}        ▷ cf. Def. 6.7
9:          end for
10:         calculate test variable W
11:         deviation = p-value corresponding to W
12:         update congruence(S)                                  ▷ cf. Def. 6.5
13:     end for
14:     if congruence(S) ≤ α then
15:         CS = CS ∪ {S}
16:     end if
17: end for
18: create new candidates Cand* using CS
19: return CS ∪ SubspaceSelection(G, M, α, k, Cand*)
```

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 6.7, we study the impact of this parameter on the results and the runtimes.

6.6. Community Outlier Detection

Given the selection of congruent subspaces in Section 6.4, we can already enhance the quality of existing community outlier detection models such as CODA [GLF+10] or of other graph mining tasks [STM07, ZCY09, DLMR11] 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 6.1 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 [GLF+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 [GLF+10] appears when it has a high local deviation. However, *ConSub* selects a set of subspaces (cf. Definition 6.1) and each object may belong to different homogeneous neighborhoods depending on the congruent subspace (cf. Figure 6.1). Thus, an outlier score (cf. Definition 6.3) 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 $N_1 \subseteq V$ and $N_2 \subseteq V$

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 6.8:

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:

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

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

where $dist_S(\alpha_S(o), \alpha_S(p)) \in [0,1]$ is any normalized distance function.

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 [GLF+10], we finally compute the outlier score w.r.t. its homogeneous neighborhood.

We define the homogeneity of a neighborhood N as follows:

Definition 6.9:

Neighborhood Homogeneity
Given a congruent subspace S, the homogeneity of a neighborhood $N \subseteq V$ 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 $dist_S(\alpha_S(o), \alpha_S(q))$ between connected nodes $o, q \in N$ of the neighborhood in subspace S. $interdist_S(N)$ is the average distance $dist_S(\alpha_S(o), \alpha_S(p))$ 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 6.1(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 6.10:

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

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

Following [BKNS00], 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.

6.7. 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[2]. 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*) [TL12], and (3) our congruent subspace selection (*ConSub*). For each of these pre-processing methods we apply community outlier detection (*CODA*) [GLF+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 distance-based outlier model *DistOut* (cf. Definition 6.10), 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 [Agg13].

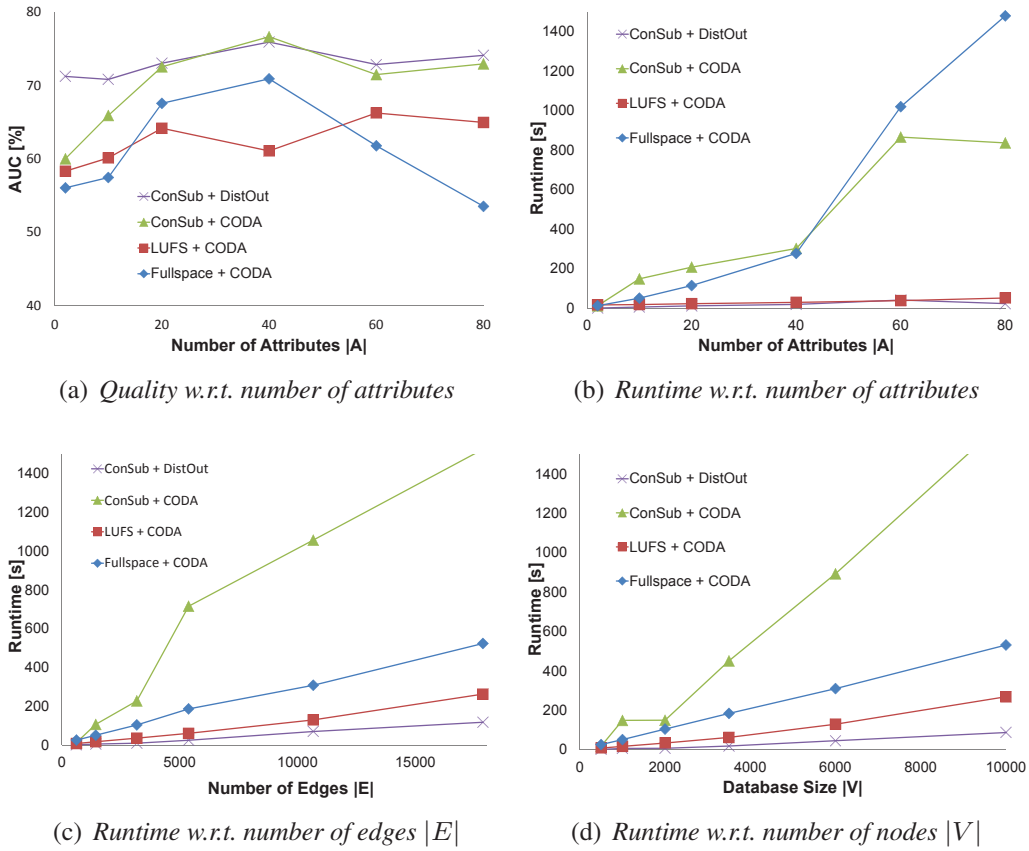
Synthetic Data

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 [LFR08]. 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 6.5(a). 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

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

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.



(a) *Quality w.r.t. number of attributes*

(b) *Runtime w.r.t. number of attributes*

(c) *Runtime w.r.t. number of edges* $|E|$

(d) *Runtime w.r.t. number of nodes* $|V|$

Figure 6.5.: ConSub results on synthetic data: quality and runtime scalability

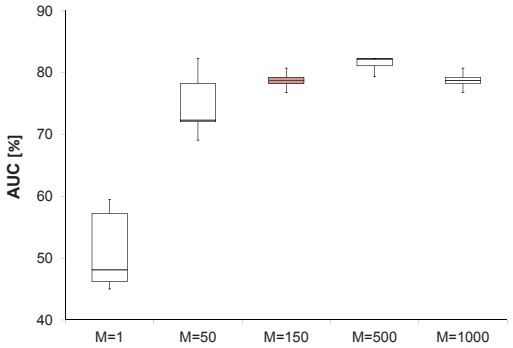
Runtime Figure 6.5(b) shows the runtimes with increasing number of attributes. *ConSub + DistOut* and *LUFS + CODA* show best scalability w.r.t. $|A|$. However, *LUFS* selects a single subspace only, while *ConSub* outputs multiple subspaces. With *ConSub + DistOut*, we can analyze more subspaces within 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 6.5(c) and Figure 6.5(d). 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*.

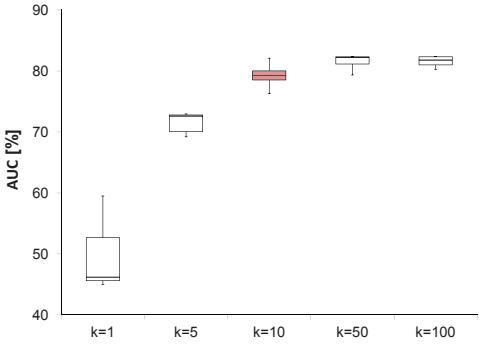
Parameter Settings The box plots in Figure 6.6 show an overview of quality results achieved on all synthetic datasets and highlights the robustness of our method w.r.t. parameterization. *ConSub* has three parameters: the significance level (α), the number of intervals (k) and the number of Monte Carlo 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 6.6). 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 6.6(a)). However, more iterations result in higher runtimes (cf. Figure 6.6(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 6.6(b). Again, a larger sample size increases the runtime (cf. Figure 6.6(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 6.5. High values $\alpha \geq 10\%$ induce considerably higher runtimes (cf. Figure 6.6(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.

Real Data

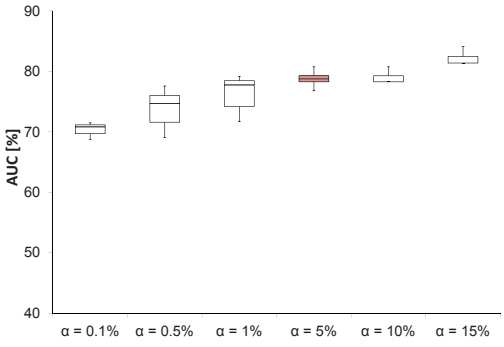
We use four attributed graphs obtained from real world networks for the evaluation of our approach. We use the *Amazon* co-purchase network [LAH07] consisting of product nodes with 28 attributes such as product prices, ratings, number of reviews, etc. Further, we use the *Enron* communication network 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. In order to present quality assessment we use the benchmarks described in Chapter 3.



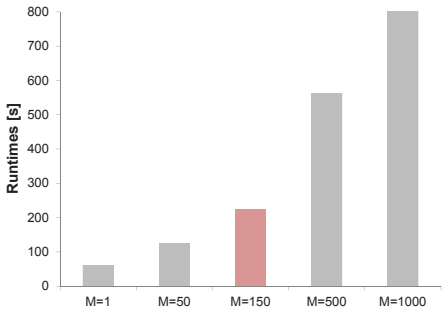
(a) *Quality w.r.t. number of Monte Carlo iterations (M)*



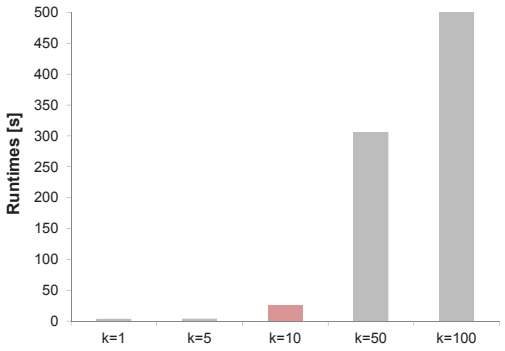
(b) *Quality w.r.t. number of intervals (k)*



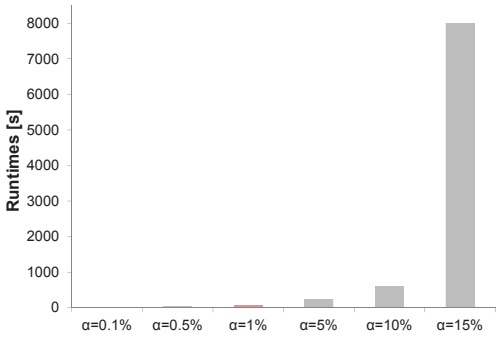
(c) *Quality w.r.t. the significance level (α)*



(d) *Average runtime w.r.t. number of Monte Carlo iterations (M)*



(e) *Average runtime w.r.t. number of intervals (k)*



(f) *Average runtime w.r.t. significance level (α)*

Figure 6.6.: ConSub parameter settings w.r.t. quality and runtimes. Evaluation on all 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 [GLF$^+$10]	67.97 ± 5.8	152.66
Edges:333	LUFS [TL12] + CODA [GLF$^+$10]	44.44 ± 13.5	3.46
Attributes:28	*Fullspace* + CODA [GLF$^+$10]	50 ± 0	6.05
Books	ConSub + DistOut	60.02 ± 0.49	2.15
Nodes:1,418	ConSub + CODA [GLF$^+$10]	53.52 ± 2.25	14.81
Edges: 3,695	LUFS [TL12] + CODA [GLF$^+$10]	-	
Attributes:28	*Fullspace* + CODA [GLF$^+$10]	53.35 ± 0	36.14
Enron	ConSub + DistOut	74.8 ± 0.08	840.54
Nodes: 13,533	ConSub + CODA [GLF$^+$10]	60.8 ± 6.98	1130.78
Edges:176,987	LUFS [TL12] + CODA [GLF$^+$10]	48.3 ± 5.48	472.6
Attributes:20	*Fullspace* + CODA [GLF$^+$10]	45.7126 ± 0	397.33

Table 6.2.: Quality and runtime of ConSub on real world networks

Since most of the approaches are non deterministic, we have executed each algorithm 20 times in order to reduce possible random effects. Table 6.2 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 6.3. *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 6.3.: Congruent subspaces in Amazon co-purchase network

6.8. Summary

In this Chapter, 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 preprocessing step for these and other graph mining paradigms.

7. Subspace Analysis for Outlier Ranking

Outlier ranking is an important task for finding anomalous objects. In practice, however, there is not always a clear distinction between outliers and regular objects as nodes have different roles w.r.t. different subgraphs and their relevant subspaces in an attributed graph. An object may deviate in a context formed by a local subgraph of the attributed graph and its relevant subspace while this same node might appear perfectly regular in other subspaces and subgraphs. One can think of these different local contexts as multiple views on one database. Traditional outlier ranking techniques search for multiple views considering only vector data. Thus, they miss complex outliers that are hidden in different local contexts that result from the combination of the graph structure and node attributes.

In this Chapter[1], we propose *GOutRank*, a flexible framework for outlier ranking in subspaces of attributed graphs. We rank graph nodes according to their degree of deviation in both graph and attribute properties. Subspace clustering provides a selected subset of nodes and its relevant attributes in which deviation of nodes can be observed. Our graph outlier ranking introduces scoring functions based on these selected subgraphs and subspaces.

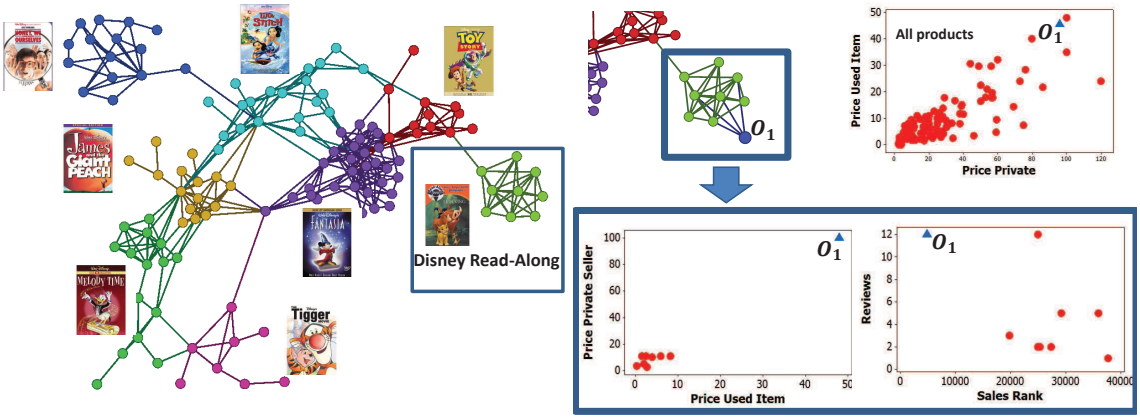
7.1. Introduction

Outlier analysis is an important data mining task for fraud detection, network intrusion analysis, anomaly detection in e-commerce, and many more. In these applications one looks for highly deviating objects that show-up in contrast to the regular objects. Outlier ranking techniques score each object based on its degree of deviation. Hence, they overcome traditional outlier detection techniques [RL87, KN98], which rely on a binary decision boundary. Outlier rankings enable a user-driven exploration of outliers

[1] Parts of this chapter has been published in the Proceedings of the IEEE International Conference on Data Mining (ICDM 2013) [MAIS+12] and in the Proceedings the International Workshop on Graph Data Management (GDM 2013) in Conjunction with IEEE International Conference on Data Engineering (ICDE 2013) [MISMB13]

by looking at the most promising objects first. They allow users to choose the decision boundary between outliers and regular objects in a flexible way.

In the past, outlier ranking techniques have focused on homogeneous vector data [CBK09] or graph data [SQCF05]. However, in many of today's applications, information of both types is available. For instance, heterogeneous data can be found on e-commerce marketplaces such as Amazon. Their product databases store a large number of attributes for each product, e.g., prices, different rating ratios, product reviews. In addition, co-purchased products are stored as a graph structure. In this scenario, exceptional objects correspond to outstanding, fake, suspicious, or overpriced products. Not all of these outliers can be detected by a traditional outlier analysis restricted to attribute values or to graph structures only. For example, overpriced products might appear quite regular if one looks at the overall price distribution of the database. However, if one combines both price and co-purchases one might reveal its high deviation in price w.r.t. to this local subgroup of co-purchased products.



(a) *Subgraph of Disney DVD products* (b) *Different subspaces of the given attributes*

Figure 7.1.: An outlier example in a subgraph of the Amazon co-purchased network

Our main hypothesis is that such complex outliers can only be detected by a combination of all available information. To this end, outlier mining techniques for heterogeneous databases have to be developed. They have to cope with information on relations between products, but also with a large number of attributes. Out of this large set of heterogeneous data, outlier ranking techniques have to automatically detect relevant data: (1) *subgraphs* as the relevant graph context of an outlier and (2) *subspaces* as the relevant attribute set in which an outlier is deviating. This is required as complex outliers deviate from their local context. For the attribute space, deviation might not be visible if one considers irrelevant attributes, e.g., randomly distributed attributes. Exceptional object deviation is also not recognized if one considers all given attributes simultaneously. Overall, outlier ranking has to measure the deviation of objects w.r.t. a subgroup of the data objects and a subset of the attributes.

Let us illustrate this in a real world example. Figure 7.1(a) shows a part of the Amazon co-purchase network. In particular, we have selected *Disney DVD* products, which have been reviewed by a group of high school students in a user experiment at our university (cf. Chapter3). Product O_1 is one of the outliers, showing up due to its high price in attributes *price offered by private sellers* and *price for used products*. This object shows high deviation w.r.t. these prices compared to its co-purchased products.

However, traditional outlier mining techniques can not detect this deviation. If we consider the graph structure only, the product is densely connected to other products. Based on the graph structure it seems to be regular. If we take only the attributes into account (cf. product prices in Fig. 7.1(b)), we observe many objects with high prices for new articles offered by private sellers and high prices for used articles. This seems to be quite regular over all products. Thus, graph structure or attributes alone can not reveal the deviation of object O_1. Nevertheless, O_1 is highly deviating in the densely connected group of *Disney Read-Along* products. All products of this subgraph have highly similar attribute values w.r.t. both prices, except for O_1. Note that this is only the case for this subspace. Other subsets of the attributes (e.g., Sales Rank and Reviews) form a very sparse subspace and do not indicate any high deviation of O_1. Overall, one can claim O_1 to be a true outlier w.r.t. to the *Disney Read-Along* products and the price attributes.

A recent research direction has focused subspace graph clustering that focuses on the selection of a local subgraph and its subspace (e.g., the *Disney Read-Along* products and the price attributes). Several algorithms has been proposed [MCRE09, ZWZK06, GFBS10, GBS11, GBFS13]. They provide various clustering models taking different application demands into account. In general, these approaches are able to detect multiple views on the same database, and groups each object accordingly to multiple subspace graph clusters. However, all of these techniques focus on object groupings and are not able to assess the deviation of individual outliers.

We see huge potential in utilizing established subspace analysis models from the domain of subspace graph clustering for subspace outlier mining on attributed graphs. Both efficiency and quality improvements in clustering could be exploited for subspace outliers in a general framework. However, subspace graph clustering poses two main challenges for outlier detection: first, each object, even if it deviates substantially in some subspaces, is very likely to be part of at least some clusters in other projections. Thus, outliers are not simply non-clustered objects. Second, assessing the degree of deviation is not straightforward. Subspace graph clusters represent groups of data in very many different (or similar) views, which makes the assessment of deviation a non-trivial task. An outlierness score for meaningful ranking requires a principled integration of these multiple views.

In this Chapter, we focus on the detection of such outliers that deviate w.r.t. a subgraph of highly connected nodes. The individual outlier shows high similarity to these nodes in the graph structure, but there exists a selection of attributes in which it deviates.

We call this selection of attributes a relevant subspace. For the automatic selection of subgraphs and subspaces we rely on recent subspace analysis and graph clustering techniques. Based on the results of these techniques, we propose several ranking functions that exploit the characteristics of the obtained subspace clusters.

7.2. Comparison to Related Work

Table 7.1 summarizes existing outlier mining techniques for attributed graphs and highlights the improvements of *GOutRank*. Traditional approaches for outlier mining have focused either on relational [VW09, RL87, KN98, BKNS00, AY01, LK05, MSS11, KMB12] or graph data [NC03, EH07, Cha04, XYFS07, SHH+10, AMF10].

Although general approaches have been proposed as pre-processing step for the selection of relevant attributes [TL12, ISML+13] (cf. Chapter 6), they do not consider a local selection of the attributes w.r.t. a local subgraph. In contrast to this, we have proposed *ConOut* in Chapter 5 that selects a single subset of attributes for a given subgraph. Although we show the scalability of *ConOut* compared to algorithms based on multiple views, a unique view of the data results in information loss.

To avoid this, different techniques have focused on subspace graph clustering on attributed graphs [MCRE09, ZWZK06, GFBS10, GBS11, GBFS13]. They address the selection of multiple subset of attributes on the graph cluster level. In this Chapter, we exploit the potential of these methods. Since they do not have been designed for outlier analysis, we introduce several ranking functions that analyze their subspace cluster for outlier ranking.

	Data Type		**Context Selection**	
	Graph	**Attributes**	**Graph Perspective**	**Attribute Perspective**
Traditional Approaches				
relational data (full dimensional) [VW09, RL87, KN98, BKNS00]	✗	✓	✗	✗
relational data (multiple views) [AY01, LK05, MSS11, KMB12]	✗	✓	✗	multiple
anomalous edges [Cha04]	✓	✗	global	✗
anomalous subgraphs [NC03, EH07]	✓	✗	local	✗
outlier nodes as by-product clustering [XYFS07, SHH+10]	✓	✗	local	✗
node neighborhood analysis [AMF10]	✓	✗	local	✗
anomalous labelled subgraphs [DLMR11]	✓	✓	local	✗
Attributed Graphs				
semi-supervised [CPF06]	✓	✓	global	✗
community outlier mining [GLF+10]	✓	✓	global	✗
local context selection *ConOut* [Chapter 5]	✓	✓	local	single
global subspace selection *ConSub* [Chapter 6]	✓	✓	global	multiple
subspace cluster analysis *GOutRank*	✓	✓	**local**	**multiple**

Table 7.1.: Overview of outlier mining techniques

7.3. Problem Overview

GOutRank generalizes our previous outlier ranking method *OutRank* [MAIS+12], which has focused on high dimensional vector data without considering graph structures. Both techniques share the idea of computing a subspace clustering as pre-processing for outlier ranking. In the following, we first introduce the require notation for this chapter before we describe the challenges of our framework.

A subspace clustering result in an attributed graph is a set of subspace clusters:

$$Res = \{(C_1, S_1) \dots (C_n, S_n)\}$$

where $C_i \subset V$ is a densely connected subgraph with high attribute similarity in the subspace $S_i \subset A$. A node $v \in V$ can be part of multiple clusters in different subspaces. We denote by $|S_i|$ the cardinality of a cluster subspace (C_i, S_i). Let $|C_i|$ be the cardinality of cluster C_i and $|S_i|$ the dimensionality of subspace S_i for $(C_i, S_i) \in Res$. max_C is the maximal cluster size in Res and max_S the maximal dimensionality of Res. An outlier ranking is a sorted list of all $o \in V$, in ascending order of a scoring function:

$$score(o) : V \rightarrow \mathbb{R}$$

The score represents a measure for the objects' regularity, and it considers both graph structure and attribute values. In this chapter, outliers have low scores, and regular objects have high scores.

GOutRank has been designed for complex outliers, which deviate only w.r.t. a local subgraph and a subset of relevant attributes. In order to assess complex deviations, subspace clusters are analyzed, and the results are integrated into a score for each object. To compute the score, we have to formalize the degree of regularity (or deviation) of an object in the subspace. Subspace clustering provides groups of regular objects, and potential outliers in the respective subspace. An important consideration for the ranking is to avoid bias associated with similar views that do not carry new information regarding regularity (or deviation) of an object.

Figure 7.2 illustrates some examples of the complex outliers that *GOutRank* focus on. For instance, node O_1 has highly deviating attribute values w.r.t. the selected subgraph and subspace $S_1 = \{d_1, d_2\}$ (subspace cluster result (C_1, S_1)). On the other hand, it appears to be a regular w.r.t. other selected subgraphs and subspaces. It is clustered in (C_3, S_2) and (C_5, S_5). *GOutRank* tackles the challenges with these outliers hidden in combination of subspaces and subgraphs in attributed graphs and it detects outliers that can not be detected by traditional techniques, single view techniques such as *ConOut* or global approaches like *ConSub*.

Different subspace clustering algorithms for attributed graphs have been proposed. Each technique provides a different cluster definitions (e.g. grid-based [GFBS10] or

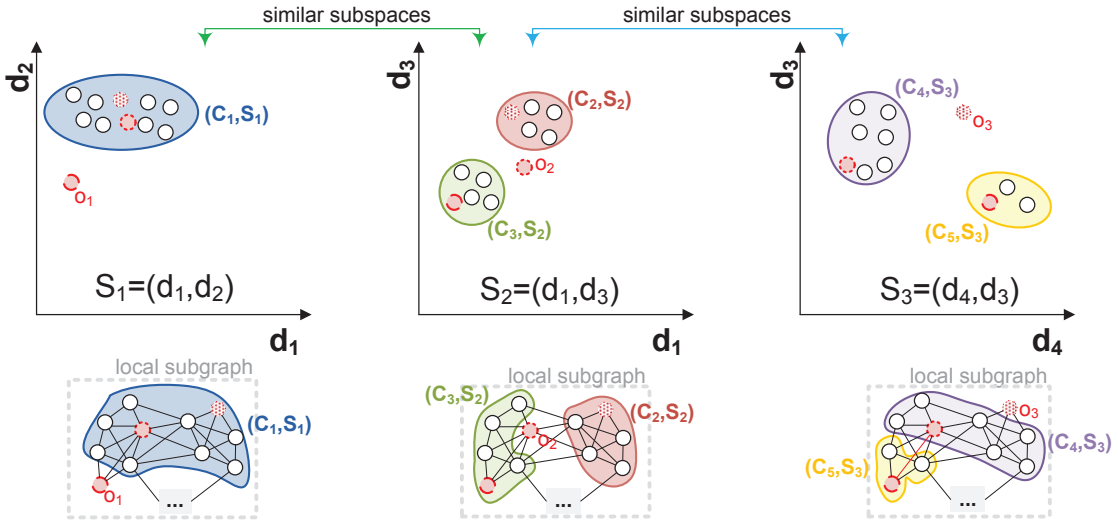


Figure 7.2.: Outliers w.r.t. multiple subspace views and local subgraphs. Example of a subspace clustering result: $Res = \{(C_1, S_1), (C_2, S_2), (C_3, S_2), (C_4, S_3), (C_5, S_3)\}$

density-based [GBS11]), which fulfills a certain application demand. We abstract from these individual definitions and use a general clustering result as input for our outlier ranking. Overall, *GOutRank* does not require a particular instantiation, and can therefore be adapted to new developments in scoring or subspace analysis. We consider this decoupling of *scoring* and *subspace analysis* as a major contribution to the development of future outlier ranking techniques. However, the development of such scoring functions in attributed graphs induces three main challenges: the selection of subgraphs with their individual subspaces, redundancy in the cluster results and the scoring of objects in these multiple subspaces. In the following, we discuss the main challenges before presenting the *GOutRank* solution in detail.

Selection of subgraphs and subspaces We deem the selection of subgraphs and subspaces the main challenge for outlier ranking in attributed graphs. In graph data, densely connected subgraphs stand for clusters with high intra-cluster similarity. Many relations between these clustered objects are a clear indicator for a homogeneous subgroup. Considering the attributes of each clustered node, we observe a correlation between the graph structure and some attribute values. Hence, a group of clustered nodes may only show high attribute similarity for a subset of relevant attributes. As illustrated in Figure 7.1(b), some subspaces show high correlation with the selected subgraph, while other attributes may tend to be irrelevant for this subgraph and show scattered attribute values.

As mentioned in Section 7.2, recent techniques [MCRE09, ZWZK06, GFBS10, GBS11, GBFS13] set about solving this challenge. These approaches detect subspace clusters in attributed graphs. For instance, Figure 7.2 shows an example of a posible subspace clustering result. We have consciously decided to take their results as input to our

scoring functions in order to solve this first challenge.

Redundancy of Subspaces A subspace clustering result Res is usually redundant, i.e., a subspace cluster (C_i, S_i) often overlaps (with respect to the clustered objects) with other subspace clusters (C_j, S_j) (e.g., $\{(C_1, S_1)\}$ and $\{(C_2, S_2), (C_3, S_3)\}$ from example shown in Figure 7.2). Typically, these overlaps occur when the subspace projections share many attributes. In the extreme case, a subspace cluster is reflected in all its lower dimensional projections as stated by the following monotonicity property:

$$(C_i, S_i) \in Res \Rightarrow (C_k, S_k) \in Res \, \forall S_k \subseteq S$$

Most subspace clustering models obey this monotonicity property [GFBS10, GBS11]. The inverse property is often used to prune subspaces for efficient subspace processing.

As a consequence, each object $o \in (C_i, S_i)$ is clustered in all $2^{|S_i|} - 1$ many lower dimensional subspace projections. Even worse, is the fact that subspace clusters are expected to re-occur in very similar subspaces that share dimensions:

$$o \in (C_i, S_i) \wedge o \in (C_j, S_j) \text{with} |S_i \cap S_j| \neq 0$$

Following our example of Figure 7.2, S_1 and S_2 share the dimension d_1. Outlier scores should be aware of the similarity between subspaces, which captures the increasing expectation of shared cluster structures.

Scoring of objects in multiple subspace clusters A naive outlier score would assign $score(o) = 1$ to all objects that occur in at least one cluster and $score(o) = 0$ to all objects that are not clustered. However, current subspace graph clustering techniques in attributed graphs allow to obtain multiple views of an object w.r.t. the graph structure and the relevant subset of attributes. Such a function does not consider that an object might belong to several subspace clusters (cf. Figure 7.2), and it misses thereby essential information about each object given by its different cluster assignments. This information should be included for outlier ranking, and scoring should also depend on the occurrence of objects in different subspace clusters.

7.4. Ranking Functions

In this work, we analyze two different type of scoring functions. First, we use the properties derived from the graph subspace clustering results such as subspace dimensionality or similarity. Specifically, we propose three different functions that use different characteristics from these results. However, nodes in a graph are also characterized by centrality measures which can provide meaningful information for outlier detection as shown in [TPT+10, DKB+12].

Thus, we also introduce some scoring functions including the information of such measures. In the following, we describe first the ranking functions based on subspace clustering results and, then, we define the functions including centrality measures.

Scoring based on Subspace Clustering Results

The size of the clusters and their dimensionality is a relevant property for ranking outliers. For instance, outlier O_2 from our example frequently appears in the biggest clusters (cf. Figure 7.2). On the other hand, O_1 is always clustered in smaller clusters. Thus, O_1 seems to be more deviating as O_2 which is clustered in larger clusters. In the following, we introduce our first function based on these two properties:

Definition 7.1:

Baseline Score BS
Given a node $o \in V$, we define the baseline score BS as:

$$score_{BS}(o) = \frac{1}{2} \cdot \sum_{\{(C_i,S_i) \in Res \mid o \in C_i\}} \frac{|C_i|}{max_C} + \frac{|S_i|}{max_S}$$

This function defines outliers as objects that are found in abnormally few and low dimensional subspace clusters. Its core idea is that regular objects tend to cluster with many other similar objects. This is used as a first indication of the regularity of objects. The dimensionality of clusters is used as the second indication. Objects that are part of clusters with many attributes have strong dependencies in several properties. Hence, these regular objects get high scores.

However, this simple measure has clear drawbacks if outliers are not reflected by small and low dimensional clusters. Our first measure does not include a comparison of neither subspaces nor the detected set of clustered objects. As depicted in our previous example (cf. Figure 7.2), outliers might be detected only due to their unexpected deviation in similar subspaces. Comparing two similar subspaces and the contained clusters leads to a more enhanced scoring. Essentially, redundant clusters do not provide any knowledge for outlier scoring. They simply count each object multiple times and introduce a bias to the overall scoring function. Thus, our more enhanced evidence measures incorporate the similarity of cluster and subspace sets derived from the Jaccard Index: $simObj(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|}$ and $simDim(S_i, S_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$ respectively.

Definition 7.2:

Subspace Similarity Score SS
For each object $o \in V$, we compare each $(C_i, S_i) \in Res$ with all other subspaces $S^* \in Res$:

$$score_{SS}(o) = \frac{1}{|Res|} \cdot \sum_{\{(C_i,S_i) \in Res \mid o \in C_i\}} mean\{subDif(o, S_i, S^*)\}$$

with $S_i \neq S^*$ and

$$subDif(o, S_i, S^*) = \begin{cases} 1 - simDim(S_i, S^*) & \text{, if } \forall (C^*, S^*) \in Res \Rightarrow o \notin C^* \\ 1, \text{ else.} \end{cases}$$

In the extreme case, an object gets the highest $score_{SS}(o) = 1$ if it is clustered in all subspaces. This is the best evidence of being regular. If o is clustered in (C_i, S_i) but not in S^* then it depends on the (dis-)similarity of S_i and S^*. For very similar subspaces one expects that clustered structures reoccur. For redundant subspace clustering models this is true due to the monotonicity property. As depicted in Figure 7.2 it is usually the case that clusters reoccur due to correlated attributes. Definition 7.2 is aware of this property and expects this situation. In contrast to this expectation, it highlights outliers that show unexpected behavior in such similar subspaces. Lowest scores are assigned to objects $o \in (C_i, S_i)$ but not clustered in any of its similar subspaces $simDim(S_i, S^*) \approx 1$.

Overall, the score aggregates the behavior of o in multiple views, comparing cluster with the residual subspaces in Res. For each cluster (C_i, S_i) we use the harmonic mean of the subspace difference $subDif(o, S_i, S^*)$ such that strong deviation in one subspace does not dominate the overall score. It enforces low scores only for outliers that show high deviation in many of their similar subspace projections S^*.

In our third scoring function we go even further and consider the possible split of (C_i, S_i) in a set of clusters $\{(C_1, S^*), \ldots, (C_j, S^*)\}$ in a similar subspace S^*. A simple example of a split is given in Figure 7.2: The cluster (C_1, S_1) is split-up and covered by the two cluster C_2 and C_3 in subspace S_2. As given in the following definition, this comparison heavily involves more and more possible reasons for the deviation of the object o:

Definition 7.3:

Cluster Coverage Score CC
For each object $o \in V$, we compare each $(C_i, S_i) \in Res$ and $o \in C_i$ with all other subspace cluster sets $C^* \in \{(C_1, S^*), \ldots, (C_j, S^*)\}$ with high coverage of (C_i, S_i):

$$score_{CC}(o) = \frac{1}{|Res|} \cdot \sum_{\{(C_i,S_i) \in Res \mid o \in C_i\}} mean\{covClust(o, Cov, S^*)\}$$

with $S \neq S^*$ and $covClust(o, Cov, S^*) =$

$$\begin{cases} (1 - simDim(S_i, S^*)) \cdot \quad mean\{simObj(C_i, C^*) \mid C^* \in Cov\} \\ \quad \text{, if } \exists (C^*, S^*) \in Res \wedge o \in C^* \\ \\ simDim(S_i, S^*) \cdot \quad mean\{(1 - simObj(C_i, C^*) \mid C^* \in Cov\} \quad \text{, else.} \end{cases}$$

, and Cov a set of clusters that covers the objects in C_i best w.r.t. $simObj(C_i, C^*)$.

In contrast to the previous definitions, $score_{CC}$ includes the possibility of clusters splitting up in multiple clusters. This can happen as similar subspaces S^* might reveal sub-structures Cov that cover the original subspace cluster (C_i, S_i). We utilize the same notion as before and match the "evidence of regularity" to the similarity of subspaces and its contained subspace clusters. In the first case of cluster coverage $covClust(o, Cov, S^*)$ the object is clustered in subspace S^*. Thus, it gets high scores if S_i and S^* are dissimilar while the detected clusters C_i and C^* are very similar. This is a good indication for a regular object as it is similarly clustered in different projections. In contrast, the object gets very low scores if it is not clustered in a dissimilar subspace with very similar clusters. The later situation indicates an unexpected outlier which does not follow a similar clustering.

Clearly, Definition 7.3 requires some additional processing in finding the optimal cluster coverage Cov in each subspace. However, it is also the most complex scoring, and we would like to evaluate the quality enhancement by including more and more information. Let us briefly summarize the increase of used information in our three scoring functions:

- Baseline scoring (BS): only size and dimensionality of individual clusters are used
- Subspace Similarity scoring (SS): comparison of multiple subspaces weighted by their similarity
- Cluster Coverage scoring (CC): comparison of multiple sets of clusters that cover (C_i, S_i) weighted by the similarity of subspaces and the similarity of clusters.

Scoring considering Graph Information

However, previously explained ranking functions clearly misses some graph properties. To overcome this drawback, *GOutRank* defines other additional properties which can be extracted from the graph. They utilize the centrality of a node in the graph structure.

First, we consider the local edge density to be a valuable criterion for our scoring. We search for isolated nodes in a strong connected graph structure. On the one hand, outliers are characterized by their low edge density. While on the other hand, highly connected subgraphs should be rated as indication for regular objects. In our example, co-purchases with many other products indicates the regularity of a product as a central hub from which other products are purchased. Outliers show only very few purchases and are clustered in sparsely connected subgraphs. Furthermore, this criterion can distinguish between nodes in multiple clusters with different edge densities. Overall, highly connected subgraphs are rated as better indication for regular objects than sparsely connected graphs.

Definition 7.4:

Node Degree Score

$$score_{DEG}(o) = \frac{1}{3} \cdot \sum_{\{(C,S) \in Res \mid o \in C\}} \frac{|C|}{max_C} + \frac{|S|}{max_S} + \frac{deg(o)}{deg_{max}}$$

with $deg(o) = |\{(o,p) \in E\}|$ and $\frac{deg(o)}{deg_{max}} \in [0,1]$

as the normalized edge degree of node o.

As second indication for regularity, we observe the centrality measure obtained by the Eigenvalues [TPT+10]. This measure has been used to immunize the most vulnerable node in a graph (e.g., to make it as robust as possible against a computer virus attack). It is based on a recent development in terms of graph centrality and provides an interesting indication for our regularity measure. The indicator is based on the observation that central nodes such as hubs form the core of the regular subgraph. Thus, high scores are assigned to these nodes.

Definition 7.5:

Eigenvalue Score

$$score_{EIGEN}(o) = \frac{1}{3} \cdot \sum_{\{(C_i,S_i) \in Res \mid o \in C_i\}} \frac{|C|}{max_C} + \frac{|S|}{max_S} + \frac{|EV(o)|}{|EV|_{max}}$$

with $\frac{|EV(o)|}{|EV|_{max}} \in [0,1]$ the normalized eigenvalue of node o.

Clearly, there are further centrality measures that could be used as instantiations of our model. In addition to these both centrality measures, we also consider closeness $score_{CLOS}$ and betweenness $score_{BET}$ for the scoring functions in Section 7.5. Incorporating these basic graph properties shows significant quality improvement in our evaluation. But even more important, it highlights the potential for future regularity criteria in this scoring framework.

Finally, let us discuss the effects of the scoring functions and their intrinsic properties. They are designed as a conjunction of different indicators. Clear outliers are not part of any cluster, or they are part of clusters which only consist of nodes in very small, low dimensional, and sparsely connected subgraphs. All of these properties indicate a high deviation and lead to top ranking positions. Intermediate positions in the ranking are assigned to objects that show up in either large, high dimensional, or densely connected subgraphs. Finally, clear regular objects are clustered in large, high dimensional, and densely connected subgraphs, and thus, will be ranked at the bottom. For the graph-based components of $score_{DEG}$, $score_{BET}$, $score_{CLOS}$ and $score_{EIGEN}$ we expect centrality measures to provide an enhanced distinction between individual objects. In this respect, *GOutRank* can be considered as a general framework. It enhance its detection quality by novel future developments in both centrality measures and subspace clustering.

7.5. Experiments

In our empirical evaluation, we show the potential and the capabilities of our *GOutRank* method on a real world database. In particular, we consider the *Disney* network which is depicted in Figure 7.1(a). The existing graph clusters correspond to similar Disney films such as *Disney Pixar* Films or *Disney* classics. Product O_1 from Figure 7.1(b) is one of the real world outliers that corresponds to the overpriced film[2] *The Jungle Book (1994)* of *Rudyard Kipling's* hidden in the cluster of *Read-Along Disney* films. For our quality assessment we use the benchmark explained in Chapter 3.

[2] `http://www.amazon.com/dp/B00005T5YC`

In our evaluation on real world data, we compare *GOutRank* to the following competitors: LOF (only attributes, without subspace analysis) [BKNS00], SOF and RPLOF (only attributes with subspace analysis) [AY01, LK05], SCAN (graph clustering that detects structural outliers) [XYFS07], and CODA (graph and attribute outlier mining, without subspace analysis) [GLF+10]. In addition to this, we have compared *GOutRank* to other paradigms for outlier ranking on attributed graphs that have been presented in this thesis: *ConOut* (cf. Chapter 5) and *ConSub* (cf. Chapter 6).

Furthermore, we compare our ranking functions based only on the subspace clustering results: *Baseline* (cf. Def. 7.1), *Subspace Similarity* (cf. Def. 7.2), and *Cluster Coverage* (cf. Def. 7.3) with the ranking functions including different centrality measures: *Node Degree* (cf. Def. 7.4), *Eigenvalue* (cf. Def. 7.5), "Betweenness" and "Closeness". Additionally, we analyze our ranking functions with different multiple subspace graph clustering approaches: CoPaM [MCRE09], GAMer [GFBS10], an extension of Cocain [ZWZK06] and the recently proposed density-based approach DB-SC [GBS11]. All of these clustering techniques are publicly available in [GFBS10, GBS11].

Used data	Paradigm	Algorithm	AUC	Runtime
attributes	full space	LOF [BKNS00]	56.85	41
	subspace selection	SOF [AY01]	65.88	825
	subspace selection	RPLOF [LK05]	62.5	7
graph	graph clustering	SCAN [XYFS07]	52.68	4
both	full space	CODA [GLF+10]	50.56	2596
	local context selection (single)	*ConOut* [Chapter 5]	81.21	199
	global subspace selection	*ConSub* [Chapter 6]	81.77	8930
	subspace cluster analysis	*GOutRank*	**86.86**	**26648**

Table 7.2.: AUC[%] values and Runtime[ms] results w.r.t all competitors on the Amazon database [Disney DVD selection].

Comparison to competing approaches Table 7.2 shows AUC (area under the ROC curve) measures and the runtimes for all approaches. The loss of information is clearly visible for both paradigms: (1) approaches using only attributes and (2) approaches using only the graph structure. For the first paradigm, we observe a higher quality of subspace outlier mining [AY01, LK05] compared to the full space method [BKNS00]. This is due to the selection of relevant attributes for each individual outlier. However, they miss several outliers, hidden in combination of both data types, due to the loss of graph information. On the other hand, graph-based approaches [XYFS07, GLF+10] show very low AUC. Although CODA has both graph and attribute information available, it fails due to the curse of dimensionality in the full attribute space. *ConOut* focuses on a single projection of the attributes. This results in efficient runtimes, but it also causes an information loss since multiple views are not considered for outlier detection. On the other hand, *ConSub* avoids this selecting all subsets of the attributes. However, it does a

global selection and, thus, it is not able to detect local deviations. Overall, *GOutRank* is not as fast as the competing approaches. However, the runtimes depend heavily on the used subspace clustering technique (cf. Figure 7.3) and *GOutRank* clearly outperforms all competitors with a quality enhancement. It is able to cope with both attribute and graph information and with large numbers of given attributes. It is a successful synthesis of both graph and attribute information with high quality due to its outlier detection in selected subspaces.

	CoPaM [MCRE09]	**Cocain** [ZWZK06]	**GAMer** [GFBS10]	**DB-SC** [GBS11]
Only subspace clustering results				
$score_{BS}$	58.61	75.85	75.28	82,48
$score_{CC}$	68,07	-	78,24	63,27
$score_{SS}$	67,51	79.09	83,33	82,76
Including Graph Information				
$score_{DEG}$	69,49	76.97	82,91	82,48
$score_{BET}$	69,07	78.10	84,46	82,9
$score_{CLOS}$	69,06	78.67	85,17	83,74
$score_{EIGEN}$	72.45	77.96	86,86	82,48

Table 7.3.: AUC[%] values w.r.t. different graph clustering techniques and scoring functions on the Amazon database [Disney DVD selection].

Considering different clustering techniques and ranking functions *GOutRank* allows to use any subspace graph clustering as pre-processing step. Thus, we compare *GOutRank* with the different scoring functions and four clustering inputs [GFBS10, GBS11, MCRE09, ZWZK06] in Table 7.3. Regarding the different clustering schemes, we observe best results for *GAMer* and *DB-SC*, the most recent subspace graph clustering approaches. *GOutRank* finds most of the hidden outliers due to their high quality clustering. In particular, *GAMer* achieves better results due to the high redundancy in the cluster results compared to *DB-SC*. As shown in our experimental evaluation on relational data [MAIS$^+$12], redundancy in the results is a good feature for some of our ranking functions,i.e., the cluster covering function. However, it requires more runtimes for analyzing the clustering results. Figure 7.3 shows also the runtimes of the pre-processing step and the calculation of each of the scores. In most of the cases, the overhead for scoring is negligible in comparison to the runtime of the subspace clustering algorithms. However, the runtimes also depends on the number of obtained cluster results. Thus, the ranking functions show higher runtimes for those subspace clustering techniques with a large number of cluster results,i.e., *Cocain* and *DB-SC*. In particular, the ranking function $score_{CC}$ does not provide any result within an hour due to the high redundancy of the cluster results.

In comparison with $score_{BS}$, we observe a clear benefit of the enhanced scoring functions $score_{DEG}$, $score_{CLOS}$, $score_{BET}$ and $score_{EIGEN}$ which take the centrality of the nodes into account. Both other clustering approaches (i.e. the extension of Cocain and CoPaM) have AUC values that are worse. The results also highlight the high outlier ranking quality of *GOutRank* for the most recent graph clustering techniques. This indicates that improving the graph clustering techniques can lead to an increased outlier detection quality of *GOutRank*.

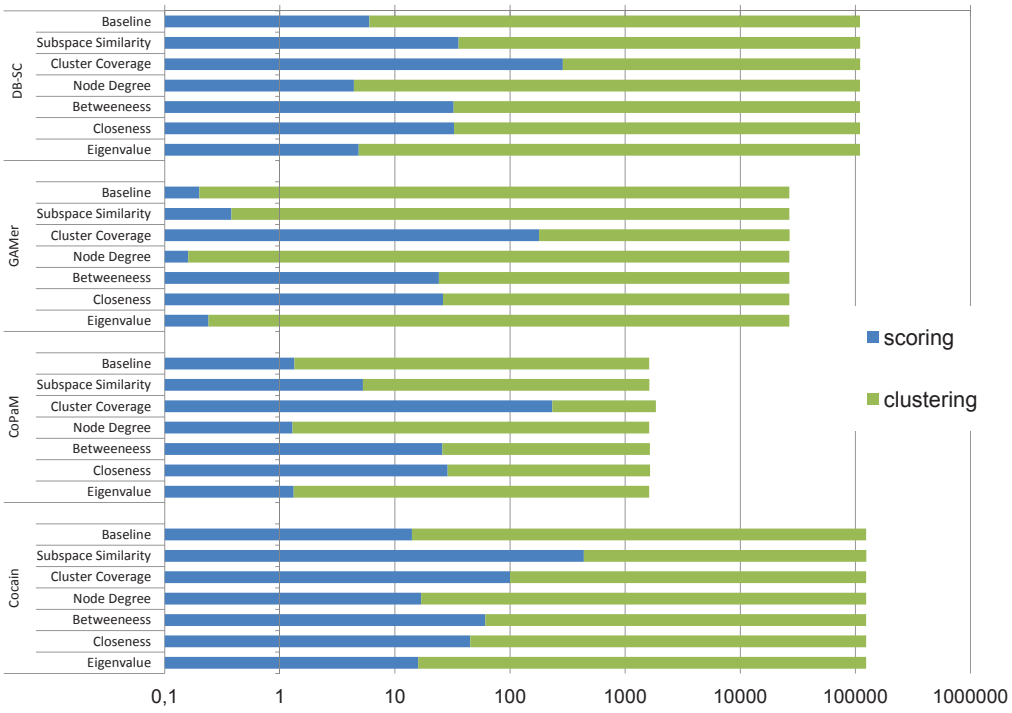


Figure 7.3.: Comparison between runtimes on the Amazon database: runtimes scoring function vs. runtimes subspace clustering. The x-axis plots the runtimes [ms] in logarithmic scale

7.6. Summary

With *GOutRank* we have proposed a first solution for outlier ranking in local subspaces of attributed graphs. Graph nodes are ranked according to their outlierness regarding both graph and attribute properties. We build upon graph clustering and subspace analysis as pre-processing steps to our outlier scoring. Both contribute to the high quality result in our evaluation. In all other cases we observe a significant decrease of the AUC values due to information loss w.r.t. graph data, or because there is no subspace analysis. We have made similar observations for our outlier scoring functions. They capture outlierness w.r.t. both subgraphs and subspaces. They are able to detect high quality

outliers in attributed graphs. Our evaluation with the Amazon network highlights that *GOutRank* has assigned high ranking positions to most of the user-labeled outliers.

Our future work in this area will focus on several open challenges. In the following, we describe the most promising ones, which have been derived out of our case study on the Amazon network.

As first open challenge, we see high potential in the integration of outlier ranking into the actual graph clustering process. This would allow an interactive exploration of outliers during the cluster computation. Top-k results could be computed directly out of the clustering task without computing scores for all objects in the database as a post-processing.

Our current two step processing has clear drawbacks w.r.t. scalability. It has to mine all subspace clusters first, before computing scores for each object in a second step. Integration of these two steps might lead to first efficiency improvements. However, further heuristics and approximations will be required to reduce the complexity in main bottlenecks such as subspace selection and complex scoring functions.

Finally, we observe an open challenge in the extraction of further node properties as indicators for our scoring. There is a variety of centrality measures available that could be used for the structural outlierness of a node. However, we see even more potential in enhanced selection methods for individual subspaces. Current clustering techniques compute one subspace for the entire cluster. Individual sets of attributes for each node might provide even better outlier scores.

Overall, there are several directions that have been opened by the basic idea of *GOutRank* for future research. We are looking forward to exploit this potential for future improvements in graph outlier ranking.

Part IV.

Summary

8. Summary

The work presented in this thesis combines the information of the graph structure with the information of the node attributes for mining attributed graphs. Overall, our research ranges from novel concepts and models for attributed graphs up to the development of benchmarks for the evaluation of the results. In the following, we summarize the main research results and we describe an outlook of possible future research directions.

8.1. Conclusion

Attributed graphs are a widely used representation of real world networks, where the information of both graph structure and node attributes is combined. Nevertheless, not all the information attached to each node is relevant and it may even be contradicting with the graph structure. Hence, combining such irrelevant attributes with the graph structure for mining simultaneously both information sources deteriorates the quality of the results. In order to solve this main challenge, we propose *context selection* for community and outlier detection on attributed graphs. In general, we present several formalizations of *contexts* that enable to be robust w.r.t. contradicting effects in the combination of attributes and the graph structure.

In this thesis, we specify a context in multiple ways depending on the application demands or the underlying data mining task. First, we focus on the selection of the relevant attributes that show dependencies with the graph. This is our first requirement for being robust w.r.t. contradicting effects of the attributes. We define this as the *attribute perspective* of a context. On the other hand, node attributes may be either only relevant for a local subgraph or they show dependencies with the entire graph. We call these different views the *graph perspective* of a context. As a result of these two possible perspectives, we propose a taxonomy of possible context selection schemes for mining attributed graphs in Part I. Based on this, the first part of the thesis compares our work with existing schemes for mining attributed graphs. Specifically, we show that context selection schemes for outlier mining on attributed graphs have not been addressed yet in the literature. Further, Chapter 3 contributes to the community providing novel use cases and benchmarks for the evaluation of outlier mining techniques on attributes

graphs. In particular, our work presents a novel use case for outlier mining in electronic platforms.

Regarding the formalization of context selection schemes, we introduce different models which we map onto different steps of the KDD process for attributed graphs (cf. Figure 1.2). Accordingly, we have categorized our techniques in two main groups: *model-dependent* and *generic approaches*. Our *model-dependent* approaches are focused on the extraction of specific patterns out of the data that corresponds to the data mining step of the KDD process. In other words, the context selection is done based on an underlying cluster or outlier definition. Alternatively, *generic approaches* provide more flexibility due to their independence on the model. This enables to use them as pre-processing or post-processing steps for the enhancement of other techniques.

Part II presents model-dependent techniques. First, we propose efficient algorithms for a modularity-driven clustering on attributed graphs. The robustness w.r.t. both outlier nodes and irrelevant attributes is one major requirement for clustering attributed graphs. To solve this, we introduce a novel measure called *attribute compactness* that only considers the relevant attribute information. Then, we incorporate this attribute information into the well-established quality measure modularity and define a novel quality measure called *attribute-aware modularity*. We prove that its maximization is NP-hard. Thus, we ensure the incremental and numerically stable calculation of our measure for the development of efficient algorithms. Finally, we empirically show that our modularity-driven approach is not only robust w.r.t. irrelevant attributes and outlier nodes, but it is also efficient.

Regarding a model-dependent approach for outlier detection, we introduce *ConOut*, a local approach for outlier ranking on attributed graphs. Its goal is to compute the outlierness of each object in the database and, then, sort the objects based on their deviation w.r.t. their own neighborhood. For each node, we specify a local context consisting of a set of nodes (local subgraph) and its relevant attributes that are determined by a statistical test. As we ensure high contrast in the attribute values due the selection of the relevant ones, we can accurately compute the local deviation of each object w.r.t. its own neighborhood. Furthermore, we include the information of the graph structure in the ranking function. This means that, for instance, highly connected nodes in the graph with highly deviating attribute values are ranked first. Overall, our approach *ConOut* proposes an efficient context selection that scales w.r.t. both the graph size and the number of attributes.

In summary, Part II focuses on the development of efficient and model-dependent algorithms that generalize well-known models to attributed graphs. Specifically, the efficiency of both of these approaches relies on the single projection of the attributes that enables to be linear w.r.t. the dimensionality of the database.

In contrast to this, Part III focuses on generic approaches that tackle the challenge of multiple views. In other words, the algorithms in the following chapters select multiple

contexts. First, we propose the pre-processing technique *ConSub* that selects all subsets of attributes showing dependencies with the entire graph structure. Its goal is to improve full dimensional techniques ensuring the *homophily* assumption for multiple subset of attributes. To achieve this, we introduce a novel measure for the degree of *congruence* between the graph and the attribute values. The core idea of this measure is to use a statistical test, which congruent subspaces must pass. An additional challenge arises when considering the exponential number of possible subspaces to be analyzed for the selection of multiple contexts. We solve this introducing an efficient heuristic. In our experiments, we do not only show the enhancement of existing full dimensional techniques with *ConSub*, but we also demonstrate the efficiency of our algorithm with a large graph.

In addition to the variety of existing full dimensional approaches, several approaches for subspace clustering for attributed graphs have been also proposed in the literature. In contrast to *ConSub*, they select locally different subspaces for multiple subgraphs. With *GOutRank*, we introduce a flexible framework that enables to apply any subspace clustering technique for outlier ranking on attributed graphs. We propose several ranking functions that analyze the results of these subspace clustering techniques for calculating the deviation of each object. In addition to this, we include the information of centrality measures to enrich the ranking functions with more information of the graph structure. A main advantage of *GOutRank* is that the presented functions do not depend on the underlying cluster model. As a consequence, different techniques can be applied and, thus, it benefits from further developments in the research area of graph subspace clustering due to its flexibility.

Overall in Part III, we propose general approaches that leverage or improve other research efforts. They do so by having generic context selection schemes. Furthermore, all previously explained techniques tackle the challenge of multiple views on attributed graphs for both community and outlier detection on attributed graphs.

8.2. Future Work

In our work, we have focused on approaches for continuous node attributes and static attributed graphs. Considering our use case of electronic platforms, more research questions arise due to the volume and variety of the data in these databases. Hence, several research questions are still open regarding the analysis of attributed graphs. In the following, we summarize the most interesting ones:

Mixed Attribute Type Our presented approaches assume that all node attributes are continuous variables. However, node attributes in real world networks consist of different attribute types. For instance, a product in a co-purchased network belongs to a category which is a categorical variable or it may be characterized by a binary attribute

indicating if it contains a photo. Further, a product may include several reviews which are store in text form. In this thesis, we propose several context selections schemes based on statistical tests or we provide flexible frameworks where the techniques can be interchanged. Thus, we can easily adapt our techniques to other specific variable types. For instance, we can change the statistical test to one that considers categorical variables in *ConOut* (cf. Chapter 5) or we can use a subspace clustering algorithm for binary attributes in *GOutRank* (cf. Chapter 7). However, to consider only one attribute type (continuous, categorical or text) results in a significant information loss of information. Therefore, the algorithms have to be able to handle with different attribute types simultaneously. In other words, correlations between distinct types such as a categorical variable (e.g., *product category*) and a continuous variable like *price* have to be quantified . While traditional approaches for vector data have already considered this research question, it is still an open challenge to analyze the dependencies between two different variable types and the graph structure.

Imperfect Data The approaches proposed in this thesis focus on the robustness w.r.t. irrelevant attributes and outliers, but the treatment of missing values is also an important issue to be considered. Following our use case of electronic platforms, not all the attributes are known for all products. For instance, products may be not offered by private sellers and, thus, the price value, when the product has been used, is sometimes unknown. Further, errors during the data collection may cause missing attribute values or even missing edges in the graph structure. Setting a default value in these cases distorts the results. On the other hand, deleting these missing values results in an information loss. To avoid this, the information of both the graph and the attributes can be used to predict these values. For instance, the graph neighborhood can be considered to infer the missing attribute values of a node. Multiple type of techniques can be designed for missing values which can assist different steps of the KDD process similar to the models proposed in this thesis. They can be designed as pre-processing steps that provide previously a interpolation of the missing values (attributes or edges). On the other hand, new model-dependent approaches being robust with these missing values can also be designed. In both cases, the main challenge is to unify the attribute information with the graph structure for the inference while being simultaneously aware of contradicting effects from this combination. Otherwise, irrelevant attributes or outlier nodes may deteriorate the results.

Dynamic In this thesis, we have considered static attributed graphs, but real world networks evolve over the time. Considering dynamic attributed graphs, both the graph structure and the node attributes change continuously. For instance, the price of a product offered by a private seller may change simultaneously when this product is bought with other products together. In other words, changes may occur together in both sources of information: graph and attributes. As a consequence, dynamic attributed graphs require to consider simultaneously the changes caused over the time according to both sources of information. This can not be done with traditional approaches for time series or dynamic plain graphs since they neglect one source. Thus,

novel definitions have to be first introduced when handling with dynamic attributed graphs. Further, not all changes are relevant and these dynamic networks also require novel context selection schemes which are still an open challenge.

Distributed Algorithms Scalability is a a major requirement for mining real world networks. Today‘s applications such as social networks or electronic platforms have a huge amount of users. Although we propose efficient algorithms for attributed graphs in this thesis, our approaches has been designed for a sequential computation. Thus, they are not able to take advantage of parallel and distributed systems for improving performance. Currently, traditional approaches for vector or graph data have been enhanced in order to execute them in parallel or distributed environments due to the concern of this huge amount of data. However, this issue has still not been considered for attributed graphs. In the development of distributed algorithms for attributed graphs, the classical challenges of parallelization arise. First, the sequential version of these methods cannot be parallelized in a straightforward way (e.g., optimization of attribute-aware modularity presented in Chapter 4). Thus, one has to develop new algorithms guaranteeing their correctness w.r.t. its sequential version. Second, the parallelization of algorithms does not ensure scalability on huge volumes of data. Parallel algorithms can also have a worse performance w.r.t. their sequential versions if they are not carefully designed. For example, the communication of the results between the machines or the fusion of the results can produce an additional overload. Finally, one has to ensure a balanced distribution. In particular, the properties of scale-free networks require a careful load balancing w.r.t. the selection of nodes to be treated in each machine. Since the node connectivity within social networks from social media is highly skewed presenting few nodes with extremely high degrees, a bad distribution of the graph components hinders the scalability of these approaches as has been already shown for several parallel graph mining techniques.

Evaluation Finally, one of the most relevant steps in the development of data mining techniques is their evaluation. Usually, data mining techniques are validated with both: synthetic and real world networks. In contrast to the plethora of traditional graph generators, few generative models have been proposed for attributed graphs and, all of the existing ones focus on categorical attribute values. In this work, our approaches have been evaluated with synthetic attributed graphs following some basic structural properties from well-established graph generators. Nevertheless, we have generated the distribution of the attribute values according the our underlying assumptions. A more objective evaluation requires more realistic data generation as done traditionally for plain graphs. However, it is still an open research question how continuous attributes correlate with the graph structure in order to formalize generative models.

Regarding the validation on real world networks, a quantitative assessment of their quality is only possible if the ground truth is known. In this thesis, we have have put many efforts to provide benchmarks on real world networks. Nevertheless, benchmarks on larger networks are still an open challenge. First, to label such a large and complex

dataset is challenging and expensive. Crowdsourcing or user experiments to label a large attributed graph are a possible solution to this problem, but a good and unbiased design of such experiments arises several challenges. First, a deep understanding of the data and some basic notions (e.g., the general outlier concept) are required in order to have accurate labels. Otherwise, the provided labels may be the result of random assignments. Another challenge is to ensure an uniform distribution of the labels. In other words, it is not not meaningful that only a small ratio of the entire dataset is analyzed and labeled. Overall, to specify the ground truth on real world networks is problematic, but is is essential for the quality assessment of further or existing approaches on attributed graphs.

In summary, our work contributes to different aspects of data analysis on attributed graphs. It introduces novel concepts and a variety of approaches for mining them as well as use cases and benchmarks for their evaluation. However, there are still open research directions for the enhancement of the models proposed, their performance on massive networks and the design of benchmarks for their evaluation.

Bibliography

[ACL06] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2006.

[Agg12] Charu C Aggarwal. Outlier ensembles: Position paper. *ACM SIGKDD Explorations Newsletters*, 2012.

[Agg13] Charu C Aggarwal. *Outlier Analysis*. Springer, 2013.

[AIS93] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 1993.

[AMF10] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *Advances in Knowledge Discovery and Data Mining*. Springer, 2010.

[ATK14] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph-based anomaly detection and description: A survey. *Data Mining and Knowledge Discovery Journal (DMKD)*, 2014.

[ATMF12] Leman Akoglu, Hanghang. Tong, Brendan Meeder, and Christos. Faloutsos. PICS: Parameter-free identification of cohesive subgroups in large attributed graphs. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2012.

[AW10] Charu C Aggarwal and Haixun Wang. *Managing and mining graph data*. Springer, 2010.

[AWY+99] Charu C Aggarwal, Joel L. Wolf, Philip S Yu, Cecilia Procopiuc, and Jong Soo Park. Fast algorithms for projected clustering. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 1999.

[AY01] Charu C Aggarwal and Philip S Yu. Outlier detection for high dimensional data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2001.

[BDG+08] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2008.

[Ber06] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*. Springer, 2006.

[BGLL08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008.

[BGRS99] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is nearest neighbors meaningful. In *Proceedings of the International Conference on Database Theory (ICDT)*, 1999.

[BGW07] Ulrik Brandes, Marco Gaertler, and Dorothea Wagner. Engineering graph clustering: Models and experimental evaluation. *ACM Journal of Experimental Algorithmics (JEA)*, 2007.

[BKNS00] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2000.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 2009.

[CFZ99] Chun-Hung Cheng, Ada Waichee Fu, and Yi Zhang. Entropy-based subspace clustering for mining numerical data. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 1999.

[CGL83] T.F. Chan, G.H. Golub, and R.J. LeVeque. Algorithms for computing the sample variance: analysis and recommendations. *The American Statistician*, 1983.

[Cha04] Deepayan Chakrabarti. Autopart: Parameter-free graph partitioning and outlier detection. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2004.

[Cla05] Aaron Clauset. Finding local community structure in networks. *Physical review E*, 2005.

[CNM04] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 2004.

[CPF06] Duen Horng Chau, Shashank Pandit, and Christos Faloutsos. Detecting fraudulent personalities in networks of online auctioneers. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2006.

[CZG09] Jiyang Chen, Osmar Zaïane, and Randy Goebel. Local community identification in social networks. In *Proceedings of the IEEE International Conference on Advances in Social Network Analysis and Mining (ASONAM)*, 2009.

[DKB+12] Qi Ding, Natallia Katenka, Paul Barford, Eric Kolaczyk, and Mark Crovella. Intrusion as (anti) social communication: characterization and detection. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2012.

[DLMR11] Michael Davis, Weiru Liu, Paul Miller, and George Redpath. Detecting anomalies in graphs with numeric labels. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*, 2011.

[EH07] William Eberle and Lawrence Holder. Discovering structural anomalies in graph-based data. In *Proceedings of Data Mining Workshops in conjunction with the IEEE International Conference on Data Mining (ICDM)*, 2007.

[EK10] David Easley and Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 1996.

[ER60] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl*, 1960.

[FB07] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 2007.

[For10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 2010.

[Fre79] Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1979.

[GBFS13] 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*. Springer, 2013.

[GBS11] Stephan Günnemann, Brigitte Boden, and Thomas Seidl. Db-csc: a density-based approach for subspace clustering in graphs with feature vectors. *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2011.

[GFBS10] Stephan Günnemann, Ines Färber, Brigitte Boden, and Thomas Seidl. Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2010.

[GFRS13] Stephan Günnemann, Ines Färber, Sebastian Raubach, and Thomas Seidl. Spectral subspace clustering for graphs with feature vectors. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2013.

[GLF+10] Jing Gao, Feng Liang, Wei Fan, Chi Wang, Yizhou Sun, and Jiawei Han. On community outliers and their efficient detection in information networks. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2010.

[HKP11] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 3rd edition, 2011.

[HZZL02] Daniel Hanisch, Alexander Zien, Ralf Zimmer, and Thomas Lengauer. Co-clustering of biological networks and gene expression data. *Bioinformatics*, 2002.

[ISMIB14] Patricia Iglesias Sánchez, Emmanuel Müller, Oretta Irmler, and Klemens Böhm. Local context selection for outlier ranking in graphs with multiple numeric node attributes. In *Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM)*, 2014.

[ISMK+ar] Patricia Iglesias Sánchez, Emmanuel Müller, Uwe L. Korn, Klemens Böhm, Andrea Kappes, Tanja Hartmann, and Dorothea Wagner. Efficient algorithms for a robust modularity-driven clustering of attributed graphs. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2015 (to appear).

[ISML+13] Patricia Iglesias Sánchez, Emmanuel Müller, Fabian Laforet, Fabian Keller, and Klemens Böhm. Statistical selection of congruent subspaces for mining attributed graphs. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2013.

[Jol86] Ian Jolliffe. *Principal Component Analysis*. Springer, New York, 1986.

[KKK04] Karin Kailing, Hans-Peter Kriegel, and Peer Kröger. Density-connected subspace clustering for high-dimensional data. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2004.

[KKZ09] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering High-Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2009.

[KMB12] Fabian Keller, Emmanuel Müller, and Klemens Böhm. HiCS: High contrast subspaces for density-based outlier ranking. In *Proceedings of the IEEE International Conference on Data Engineering(ICDE)*, 2012.

[KN98] Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, 1998.

[LAH07] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 2007.

[LFR08] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 2008.

[LK05] Aleksandar Lazarevic and Vipin Kumar. Feature bagging for outlier detection. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2005.

[LKSF10] Andrea Lancichinetti, Mikko Kivelä, Jari Saramäki, and Santo Fortunato. Characterizing the community structure of complex networks. *PLoS One*, 2010.

[LM12] Jure Leskovec and Julian J Mcauley. Learning to discover social circles in ego networks. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)*, 2012.

[LN08] Elizabeth A Leicht and Mark EJ Newman. Community structure in directed networks. *Physical review letters*, 2008.

[MAIS+12] Emmanuel Müller, Ira Assent, Patricia Iglesias Sánchez, Yvonne Mülle, and Klemens Böhm. Outlier ranking via subspace analysis in multiple views of the data. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2012.

[MAP06] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes-which naive bayes. In *Proceedings of the Conference on Email and Anti-Spam (CEAS)*, 2006.

[MCRE09] 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.

[MGAS09] Emmanuel Müller, Stephan Günnemann, Ira Assent, and Thomas Seidl. Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the International Conference on Very Large Data Bases (VLDB))*, 2009.

[MISMB13] Emmanuel Müller, Patricia Iglesias Sánchez, Yvonne Mülle, and Klemens Böhm. Ranking outlier nodes in subspaces of attributed graphs. In *Proceedings of the Workshop on Graph Data Management in Conjunction with IEEE International Conference on Data Engineering (ICDE)*, 2013.

[MSLC01] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 2001.

[MSS11] Emmanuel Müller, Matthias Schiffer, and Thomas Seidl. Statistical selection of relevant subspace projections for outlier ranking. In *Proceedings of the IEEE International Conference on Data Engineering(ICDE)*, 2011.

[MZK+09] Gabriela Moise, Arthur Zimek, Peer Kröger, Hans-Peter Kriegel, and Jörg Sander. Subspace and projected clustering: Experimental evaluation and analysis. *Knowledge and Information Systems*, 2009.

[NC03] Caleb C Noble and Diane J Cook. Graph-based anomaly detection. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2003.

[New03] Mark EJ Newman. Mixing patterns in networks. *Physical Review E*, 2003.

[New04a] Mark EJ Newman. Analysis of weighted networks. *Physical Review E*, 2004.

[New04b] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 2004.

[New06] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 2006.

[NG04] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 2004.

[NMB13] Hoang Vu Nguyen, Emmanuel Müller, and Klemens Bohm. 4s: Scalable subspace search scheme overcoming traditional apriori processing. In *Proceedings of the IEEE International Conference on Big Data*, 2013.

[NMV+13] Hoang Vu Nguyen, Emmanuel Müller, Jilles Vreeken, Fabian Keller, and Klemens Böhm. CMI: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2013.

[PAISM14] Bryan Perozzi, Leman Akoglu, Patricia Iglesias Sánchez, and Emmanuel Müller. Focused clustering and outlier detection in large attributed graphs. In *Proceedings of the ACM SIGKDD international conference on Knowledge Discovery and Data Mining (KDD)*, 2014.

[Ree01] David G Rees. *Essential statistics*. Chapman & Hall/CRC, 2001.

[RL87] Peter J Rousseeuw and Annick M Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, 1987.

[RN11] Randolf Rotta and Andreas Noack. Multilevel local search algorithms for modularity clustering. *ACM Journal of Experimental Algorithmics (JEA)*, 2011.

[SHH+10] Heli Sun, Jianbin Huang, Jiawei Han, Hongbo Deng, Peixiang Zhao, and Boqin Feng. gskeletonclu: Density-based network clustering via structure-connected tree division or agglomeration. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2010.

[SM13] Christian Staudt and Henning Meyerhenke. Engineering high-performance community detection heuristics for massive graphs. In *Proceedings of the International Conference on Parallel Processing (ICPP)*, 2013.

[SMJZ12] Arlei Silva, Wagner Meira Jr, and Mohammed J Zaki. Mining attribute-structure correlated patterns in large attributed graphs. *Proceedings of the VLDB Endowment (PVLDB)*, 2012.

[SQCF05] Jimeng Sun, Huiming Qu, Deepayan Chakrabarti, and Christos Faloutsos. Relevance search and anomaly detection in bipartite graphs. *ACM SIGKDD Explorations Newsletter*, 2005.

[Ste70] Michael A Stephens. Use of the Kolmogorov-Smirnov, Cramer-von Mises and related statistics without extensive tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1970.

[STM07] Motoki Shiga, Ichigaku Takigawa, and Hiroshi Mamitsuka. A spectral clustering approach to optimally combining numerical vectors with a modular network. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2007.

[SVR09] S. Sen, J. Vig, and J. Riedl. Learning to recognize valuable tags. In *Proceedings of the ACM International conference on Intelligent User Interfaces*. ACM, 2009.

[SWJR07] Xiuyao Song, Mingxi Wu, Christopher M. Jermaine, and Sanjay Ranka. Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2007.

[TL11] Hanghang Tong and Ching-Yung Lin. Non-negative residual matrix factorization with application to graph anomaly detection. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2011.

[TL12] Jiliang Tang and Huan Liu. Unsupervised feature selection for linked social media data. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2012.

[TPT+10] Hanghang Tong, B. Aditya Prakash, Charalampos E. Tsourakakis, Tina Eliassi-Rad, Christos Faloutsos, and Duen Horng Chau. On the vulnerability of large graphs. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2010.

[Vie12] Emmanuel Viennet. Community detection based on structural and attribute similarities. In *Proceedings of the International Conference on Digital Society (ICDS)*, 2012.

[VW09] Pavan Vatturi and Weng-Keen Wong. Category detection using hierarchical mean shift. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.

[Was94] Stanley Wasserman. *Social network analysis: Methods and applications*. Cambridge university press, 1994.

[WD09] Xiang Wang and Ian Davidson. Discovering contexts and contextual outliers using random walks in graphs. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2009.

[Wil45] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1945.

[XKW+12] Zhiqiang Xu, Yiping Ke, Yi Wang, Hong Cheng, and James Cheng. A model-based approach to attributed graph clustering. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2012.

[XYFS07] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2007.

[YC71] Edward A Youngs and Elliot M Cramer. Some results relevant to choice of sum and sum-of-product algorithms. *Technometrics*, 1971.

[YCN04] Kevin Y Yip, David W Cheung, and Michael K Ng. Harp: A practical projected clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2004.

[YJCZ09] Tianbao Yang, Rong Jin, Yun Chi, and Shenghuo Zhu. Combining link and content for community detection: a discriminative approach. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2009.

[YML13] Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2013.

[ZCY09] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment (PVLDB)*, 2009.

[ZCY10] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Clustering large attributed graphs: An efficient incremental approach. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2010.

[ZWZK06] Zhiping Zeng, Jianyong Wang, Lizhu Zhou, and George Karypis. Coherent closed quasi-clique discovery from large dense graph databases. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.