Computer Science and Information Systems 16(2):565–595 https://doi.org/10.2298/CSIS181001010C

Outlier Detection in Graphs: A Study on the Impact of Multiple Graph Models

Guilherme Oliveira Campos[1,2], Edré Moreira[1], Wagner Meira Jr.[1], and Arthur Zimek[2]

[1] Federal University of Minas Gerais
Belo Horizonte, Minas Gerais, Brazil
{gocampos,edre,meira}@dcc.ufmg.br
[2] University of Southern Denmark
Odense, Denmark
zimek@imada.sdu.dk

Abstract. Several previous works proposed techniques to detect outliers in graph data. Usually, some complex dataset is modeled as a graph and a technique for detecting outliers in graphs is applied. The impact of the graph model on the outlier detection capabilities of any method has been ignored. Here we assess the impact of the graph model on the outlier detection performance and the gains that may be achieved by using multiple graph models and combining the results obtained by these models. We show that assessing the similarity between graphs may be a guidance to determine effective combinations, as less similar graphs are complementary with respect to outlier information they provide and lead to better outlier detection.

Keywords: outlier detection, multiple graph models, ensemble.

1. Introduction

Outlier detection is a challenging problem, since the concept of outlier is problem-dependent and it is hard to capture the relevant dimensions in a single metric. The inherent subjectivity related to this task just intensifies its degree of difficulty. The increasing complexity of the datasets as well as the fact that we are deriving new datasets through the integration of existing ones is creating even more complex datasets. Consequently, methods have been specialized in various ways, e.g., for high-dimensional data [45,18,26], for sequence data [8], or for spatial data [38]. Graphs, which are able to express a variety of rich data relationships [2], reinforce the trend towards more complex concepts of outliers.

Ensemble techniques have been used in outlier detection (including outliers in graphs [32]). Even though ensemble techniques are not yet well studied nor established for outlier detection in graphs, this powerful approach becomes more and more frequent on relational datasets, mainly because of the benefits of combining individual outlier detection results, tackling the problem from multiple perspectives [42]. The focus of existing techniques is on the design of diverse ensemble members and on the combination strategies to put individual results together in an ensemble, but they are always relying on the information available in a given graph model of some target data. However, different kinds of outliers may be easier or harder to detect in different graph models derived from the same original raw data.

Normally, the graph representations or models found in the literature to represent real-world relations were created from decisions made by the user, defining which are the nodes and which are the edges. Complex web sites or databases rich in information such as Facebook, DBLP, and Twitter, among others, can be modeled as graphs in numerous possible ways. For example, Facebook user data could be represented as a graph where each node represents a person and each edge represents the existence of a friendship connection between nodes. But we could also connect people that have common interests, the same workplace, university, etc. Or we could be interested in modeling events as nodes and connect two events with an edge if the same people attended both. Thus, these graph models contain a bias built in by the user when generating such graphs. Obviously this bias can be of great value to some users but counter-productive to other users, depending on the task that will be performed in such graphs. Users may create the graphs according to the properties that they define necessary to facilitate the task of extracting information in such graphs. However, the choice of such a bias, which is often overlooked, is striking in the final outcome of the task that will be executed on the graphs.

Generating multiple graph models for some raw data and combining results obtained on those different models is a strategy to tackle more complex and diverse outliers. However, the generation of a graph for some given data is problem dependent. Although the model dimensions are usually intuitive and the analyst is able to enumerate them, it is hard to assess which dimensions effectively improve the outlier detection. We, therefore, make our assessment using different graph models, following different intuitions about which data aspects we want to model, but also diversified by some random parameters. Note that we are not proposing a way to automatically generate multiple graph models given a database. One could debate whether it is even possible as just the data analyst may know different dimensions of a particular graph. We assess the similarity of graphs as a possible guiding principle in assembling ensembles in the absence of ground truth, i.e., in the unsupervised learning task of outlier detection.

Let us summarize in the following list the contributions of this work:

1. We devise a methodology for the quantitative assessment of the gains of multiple graph models for some given database.
2. We explore two similarity-based strategies for selecting ensemble members from multiple graph models.
3. We present a quantitative experimental evaluation of outlier detection ensembles based on multiple graph models using synthetic data, including comparison to single graph models.
4. We evaluate the gains of employing multiple graph models using two outlier detection algorithms and four combination techniques.
5. We perform three case studies on data derived from DBLP, from Citation Network, and from Facebook and provide quantitative and qualitative insights regarding multiple graph models.
6. We also present support to advise the practical data analyst to employ multiple graph models with a preference for more diverse graphs over just more graphs.

This paper is an extended version of our previous conference paper [6]. The main extensions are the use of an additional similarity measure (DeltaCon [17]), an additional base outlier detection method (Radar [22]), additional combination techniques (mean, max and borda), and an additional dataset (Citation Network).

The remainder of this study is organized as follows. We discuss related work in Section 2. We describe our methodology to characterize the gains of multiple graph models and to compose an ensemble based on a set of multiple graph models in Section 3 and its implementation in case studies in Section 4. We analyze and discuss the results in Section 5. We conclude the paper in Section 6.

2. Related Work

To the best of our knowledge, there is no previous work that analyzes the impact of using multiple graph models on the quality of outlier detection. However, Rotabi et al. [34] use multiple overlapping networks to solve the strong tie detection task. They combine information provided by two different graphs (a dense and a sparse graph) from the same dataset (e.g., Twitter) to predict strong ties. They made experiments with one dense graph and four sparse graphs from their Twitter dataset: mutual follow, phone book, email address book, and direct message, respectively. To generate multiple graph models and to combine information extracted from them to improve the prediction is related to our approach. However, the tasks of outlier detection and of building ensembles for outlier detection are different and come with different challenges.

Two particular areas of outlier detection relate to our work: methods to detect outliers in (single) graphs and methods to combine outlier results (ensembles). In the following subsections we survey some methods for outlier detection in graphs, highlighting the principles of the method employed in our experiments, and sketch some relevant research in ensemble methods for outlier detection.

2.1. Outlier Detection in Static Graphs

Various methods have been proposed to detect outliers in static graphs [2]. On plain graphs, the outliers may be identified based on structural behavior [1,13] or community behavior [7,40,41]. On attributed graphs, outliers may also be identified by structural behavior, looking for unusual substructures [29,23]. We focus on detecting community (or contextual) outliers on static attributed graphs.

The CODA algorithm [12] detects contexts and, consequently, the nodes that have not been assigned to any context as outliers. CODA results are binary: a node is either an outlier or an inlier, that is, CODA does not rank the outliers. To perform a ranking and also to identify outliers in subspaces, ConSub [36] statistically selects congruent subspaces. GOutRank [27] is based on clustering algorithms in subspaces and scores nodes according to their membership in multiple subspace clusters. FocusCO [30] detects clusters and corresponding outliers in a user-driven manner.

ConOut [35] assigns each node to a single context (subgraph) and its statistically relevant subset of attributes. As we use this method in the experiments in this work, we describe its central ideas in more detail. The context selection step aims to find local neighborhoods that are similar with respect to the graph structure. For example, if two nodes share a large number of neighbors, they should belong to the same context. Overall, the context of a node is the reflexive transitive closure of adjacent nodes that are significantly similar to the first. The next step is to select attributes for each context through the comparison of the distribution of all attribute values in the local context to the whole dataset distribution. A statistical test (F-test or Kolmogorov-Smirnov test) checks whether the context values present a significantly smaller variance compared to the entire dataset distribution. The attributes are locally chosen according to this test. The anomaly score is finally based on the multiplication of a local graph density and a local attribute deviation. These two values represent structural deviations (e.g., a node that does not present the same structural pattern as its context members) and attribute deviations (e.g., a node that has different attribute values than its context members).

A recently proposed approach named Radar [22] detects outliers by modeling attribute and network information from a residual analysis perspective. The idea behind Radar is that the attributes of every instance can be reconstructed by a linear combination of some representative instances, when considering only the information perspective. Additionaly, when link information between instances is added to the model, the framework relies in a homophily property, such that similar instances are more likely to be linked together. The authors propose an optimization problem where the objective function should be minimized to find the best coefficient matrix $\mathbf{W}$ and residuals $\mathbf{R}$ for the reconstruction of attributes whithin the network context. At the end of the optimization process, the anomaly score for each instance i is computed as the ℓ_2-norm of the i-th row of the residual matrix $\mathbf{R}$.

All techniques of anomaly detection in static attributed graphs are performed on a single graph representation derived from a given dataset. Normally, there are no works that assess the impact of the graph modeling process to a single graph for real-world databases. There are also no works that discuss the pros and cons of employing multiple graphs models to represent a given database.

2.2. Ensemble Techniques in Outlier Detection

Outlier detection in general has been improved by using ensemble methods, i.e., combining the findings or results of individual learners to an integrated, typically more reliable and better result. An ensemble is expected to improve over its components if these components deliver results with a certain minimum accuracy while being diverse [42]. The two main challenges for creating good ensembles are, therefore, (i) the generation of diverse (potential) ensemble members and (ii) the combination (or selection) of members to an ensemble.

Some strategies to achieve diversity among ensemble members are feature bagging (i.e., combining outlier scores learned on different subsets of attributes) [20], different parameter choices for some base method [11], the combination of actually different base methods [28,19,37], the introduction of a random component in a given learner [24], the use of different subsamples of the data objects [44], adding some random noise component to the data ("perturbation") [43], or using approximate neighborhoods for density estimates [15]. In a sequential setting, the first top outliers detected are removed from the data set before it is handed over to other learners [33].

Different combination procedures have been proposed based on outlier scores or on outlier rankings [20,11,19,42,32]. Some methods have also been proposed to select the more diverse or (in a semi-supervised setting) the more accurate ensemble members [37,25,33,32].

All ensemble methods in outlier detection aim at the reduction of bias and variance inherent to some learner. Assessing the impact of the bias inherent to the input data has not been addressed so far.

Considering the standard KDD process model [10], our focus is on the data transformation and its impact on what we can learn from the data. The ensemble approach is thus effectively moved to an earlier step in the KDD process, since we employ different graph models to represent the same raw data.

3. Methodology

The classical approach for detecting outliers in a dataset is to model the data as a single graph and to apply a single outlier detection method, as sketched in Figure 1(a). By using this approach, the identification of outliers is biased by the given model and the selected algorithm. Alternatively, one could use an ensemble approach to apply a set of complementary outlier detection methods on a single graph and combine their results, such that the algorithm bias is reduced. This approach is sketched in Figure 1(b). Existing work for outlier detection in graphs follows the methodologies in Figures 1(a) and 1(b). As a consequence the built-in bias from the graph model selection is not adressed.

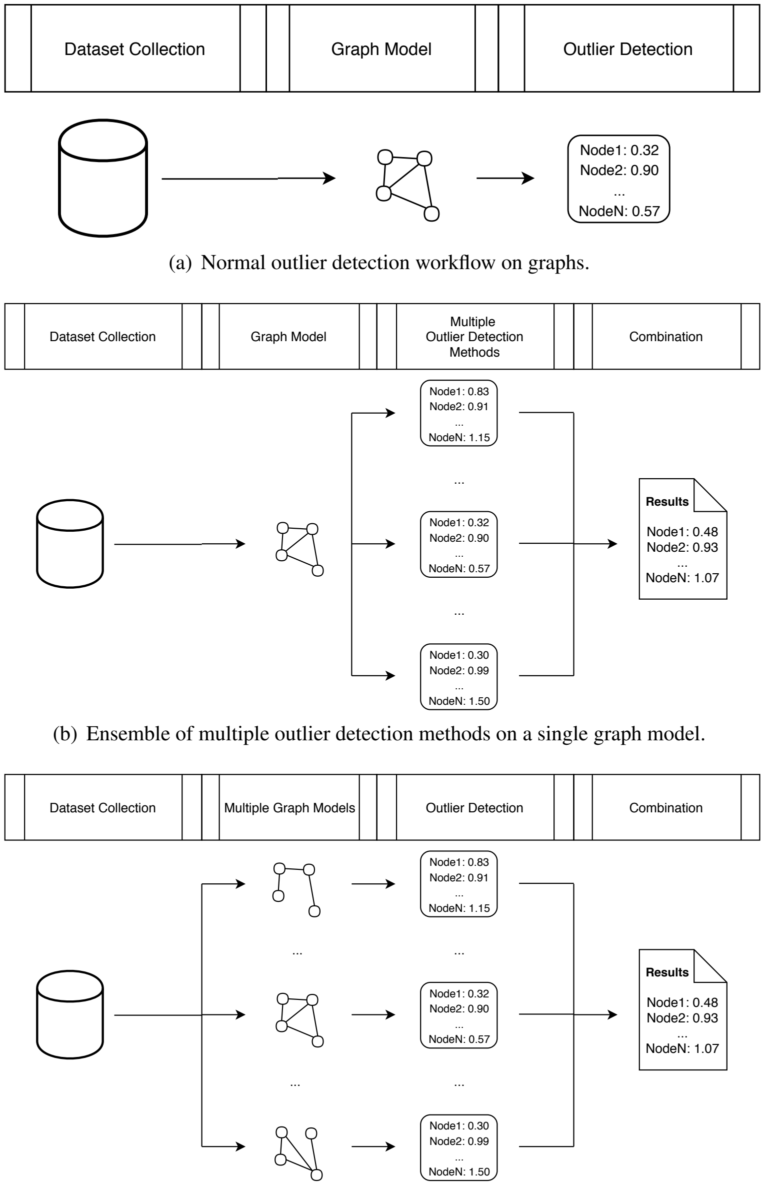
Here we propose a new methodology that tackles the reduction of graph model bias towards outlier detection by generating multiple graph models to represent the same data. The overall workflow for an ensemble method combining outlier detection results from multiple graphs is depicted in Figure 1(c). First, multiple graph models represent the same dataset, possibly taking different aspects of the dataset into account for deriving different graph models. We assume, though, that the nodes in different graphs represent the same entities. Only their relations change from model to model. Next, some algorithm to detect (node) outliers in graphs are applied to each graph model. In the last step, results from the outlier detection on the different graph representations are combined. Through the ensemble of different graphs modeling the same data, we can expect an increasing precision and robustness of the outlier detection.

For this general approach, various questions could be studied, for example, which outlier detection methods are more suitable and how the results should be combined. For this study, however, we take a fixed decision on these questions, using two methods for outlier detection and using four consensus functions to combine the results, as we are studying the impact of the design and choice of graph models.

We describe two methodologies employed in the experimental assessment presented in this work. The first methodology aims at assessing the potential gains of combining *multiple* graph models compared to using a *single* graph model. The second methodology aims at assessing the improvements for outlier detection through the combination of *complementary* multiple graph models that capture preferably different aspects of the data.

3.1. Characterizing Multiple Graph Models

We use multiple graph models here to represent different aspects of a data set, i.e., to take different perspectives. For all perspectives, i.e., derived graph models, for some dataset, the entities that may be outliers remain the same and are vertices in the graph. The different perspectives are expressed by different edges describing relations between the nodes. How different graph models express different aspects of a dataset depends on the dataset and its semantic. We will discuss specific graph models for different example datasets later. Here we characterize the notion of multiple graph models more formally.



(a) Normal outlier detection workflow on graphs.

(b) Ensemble of multiple outlier detection methods on a single graph model.

(c) Multiple graph models for outlier detection.

Fig. 1. Different workflows for outlier detection in graphs.

Definition 1. ***(Graph Model)***
A graph model G_p is described by $G_p = \{V, E_p, A\}$, where

- *p is the perspective of the graph G_p.*
- *V represents the node set and each node $v \in V$ represents an entity.*
- *E_p represents the edge set, where $E_p = \{(v_i, v_j) \mid \forall i \neq j, v_i, v_j, \in V \wedge \mathcal{F}_p(v_i, v_j) \geq t_p\}$. The function $\mathcal{F}(.)$ returns a score $\in [0, 1]$ that defines for a given perspective p the correlation or similarity between two nodes, and t is a user-defined threshold.*
- *A is the attribute set and each attribute $a \in A$ represents an attribute.*

Multiple graph models as outlined above can then be characterized as follows:

Definition 2. ***(Multiple Graph Models)***
A set of multiple graph models $\mathcal{G} = \{G_1, G_2, ..., G_p, ..., G_M\}$, where M is the total amount of perspectives, we have:

$$V_i = V_j \forall i, j \in [1, 2, ..., M].$$

Our definition assumes that different graph models have the same node set V. We thus avoid the node-to-node mapping problem, since the set of vertices, related to the dataset entities that may turn out being outliers, remains the same across the different graph models.

Given a dataset D, we generate multiple graph models $\mathcal{G} = \{G_1, G_2, ..., G_p, ..., G_M\}$ with respect to different perspectives p of M. Suppose we have two graphs $G_1 = \{V, E_1, A\}$ and $G_2 = \{V, E_2, A\}$. A single perspective p relates to a specific procedure to generate the edges of G_p, though possibly with different parametrization. For example, if D represents a citation network dataset, p is co-authorship, G_p will represent D as a co-authorship graph, in which nodes represent authors and edges represent the existence of co-authorship. In this example, an edge $e_{ij} = (v_i, v_j)$ will exist iff $\mathcal{F}(v_i, v_j) = 1$, where $\mathcal{F}(.)$ returns 1 if v_i and v_j have at least one publication together and 0 otherwise.

The perspective p may also represent correlations or similarities between nodes. For example, in a citation network dataset, a graph G_p may be defined as the correlation between publications in conferences based on co-occurrence of words in the title or in the abstract, where two nodes (authors) are connected if they have large correlation values. Or a perspective may be the defined as the similarity between authors in their research topics. In these scenarios, an edge $e_{ij} = (v_i, v_j)$ will exist iff $\mathcal{F}(v_i, v_j) > t_p$, where t_p denotes the correlation or similarity threshold in perspective p.

As these examples demonstrated, the definition of a perspective depends on the given data for a given problem. We will introduce tailored perspectives on the various experimental datasets in the case studies (Section 4).

3.2. Characterizing the Gains of Using Multiple Graph Models

To characterize the gains of using multiple graph models, we propose the following steps:

1. **Generate multiple graph models according to the problem being addressed.** For each raw dataset, we design multiple graph models that describe the same entities as nodes but differ quantitatively (how dense they are) and qualitatively (which relationships are expressed in the graph structure). These multiple graph models aim to

materialize the various perspectives that one can take on the data. As a result, they can make different kind of outliers detectable.
The design of graph models for some raw data set remains problem-dependent, though. We explore different variants for synthetic datasets and for real datasets.

2. **Assess experimentally the combined usage of multiple graph models.** We quantify experimentally the gains of multiple graph models using both synthetic and real-world datasets.

3.3. Selective Composition of Multiple Graph Models

Orthogonal to the generation of as many graph models as practical is the design of graph models that are as complementary as possible, i.e., they should model different aspects of the raw data. The less similar two graphs are, the more likely they are complementary regarding the outlier information they provide.

Two different graphs $G_1 = (V, E_1, A)$ and $G_2 = (V, E_2, A)$, which differ only in their set of edges E_1 and E_2, can have a degree of similarity measured by Jaccard [14] on their sets of edges:

$$Jaccard(G_1, G_2) = \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|}. \tag{1}$$

The similarity of a set $\mathcal{G}$ of more than two graphs is assessed as the average pairwise similarity:

$$Jaccard(\mathcal{G}) = \frac{1}{|\mathcal{G}|} \sum_{\substack{G_1, G_2 \in \mathcal{G} \\ G_1 \neq G_2}} Jaccard(G_1, G_2). \tag{2}$$

DeltaCon [17] is another algorithm to compute similarity between graphs. Given two input graphs $G_1 = (V, E_1, A)$ and $G_2 = (V, E_2, A)$, with different edge sets E_1 and E_2, DeltaCon computes the similarity score $DeltaCon(G_1, G_2) \in [0, 1]$, such that a score of 1 means identical graphs.

DeltaCon applies the Fast Belief Propagation (FaBP) [16] method to measure node affinity in the same graph and to build the $n \times n$ similarity matrix **S**, based on the $n \times n$ identity matrix **I**, the $n \times n$ diagonal degree matrix **D**, and the $n \times n$ adjacency matrix **A**:

$$\mathbf{S} = [s_{ij}] = [\mathbf{I} + \epsilon^2 \mathbf{D} - \epsilon \mathbf{A}]^{-1} \tag{3}$$

ϵ is a small constant to regulate influence of the neighboring nodes.

The main idea is to analyze graph differences from the information flow viewpoint. For instance, a missing edge in a clique subgraph is not as important as a missing edge connecting two dense subgraphs.

In its fastest version, DeltaCon divides the node set into g groups and computes the affinity score of each node to each group. The $n \times g$ similarity matrix S_1 and S_2 is then built for each graph G_1 and G_2, respectively, and the similarity score is computed as follows:

$$d(G_1, G_2) = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{g} \left(\sqrt{s_{1,ij}} - \sqrt{s_{2,ij}}\right)^2} \tag{4}$$

$$DeltaCon(G_1, G_2) = \frac{1}{1 + d(G_1, G_2)} \tag{5}$$

The similarity of a set $\mathcal{G}$ of more than two graphs is assessed as the average pairwise similarity:

$$DeltaCon(\mathcal{G}) = \frac{1}{|\mathcal{G}|} \sum_{\substack{G_1, G_2 \in \mathcal{G} \\ G_1 \neq G_2}} DeltaCon(G_1, G_2). \tag{6}$$

While previous methods to select members for outlier detection ensembles (trained on the same dataset) rely on some sort of estimate of the quality in the unsupervised scenario [37,32], the similarity of graph models is completely agnostic of (supposed) outliers and can therefore serve as a general guidance to the selection of ensemble members.

4. Implementation of the Methodology: Case Studies on Synthetic and Real Data

We perform case studies on synthetic data, data from DBLP, data from Citation Network, and data from Facebook. We focus on quantitatively different graphs for synthetic data, and on semantically different graphs for DBLP, Citation Network, and Facebook data. These real-world databases were chosen to represent realistic scenarios where multiple graph models are viable ways to help detecting outliers. By considering different areas we demonstrate the reproducibility of our results. It should be noted that it is straightforward to derive quantitatively different graphs in an automated way, while semantically different graphs need to be designed manually for each problem, taking into account the semantics of the data and considering which information may be possibly interesting.

4.1. Synthetic Datasets

For a quantitative experiment, we generate three families of synthetic datasets (A, B, and C) to evaluate the benefits of combining multiple graph models from the same source: (A) graphs following a power-law degree distribution, in particular Zipf's law according to Gao et al. [12], (B) graphs generated by following a stochastic algorithm proposed by Barabási and Albert [3], and (C) graphs following the Erdős and Rényi model [9], in which every possible edge is created with the same constant probability. For these dataset families, we start with a graph comprising 4950 nodes generated according to the corresponding distribution or algorithm, respectively, and include 50 outliers that follow a uniform degree distribution. We repeat this procedure 10 times, so we have 10 base graphs for each experiment with 5000 nodes each, where the inliers follow a known degree distribution and outliers follow a uniform degree distribution. We also add 20 attributes and set a random percentage of those to be relevant for the outlier detection task. Relevant attributes follow different Gaussian distributions for outliers and inliers with $\mu = [-10, 10]$ and $\sigma = [1, 5]$. Irrelevant attributes follows the same Gaussian distribution for outliers and inliers.

We use two parameters to induce diversity in the graphs, determining the percentage of edges that we remove from the graph or that we add to the graph, respectively. This way we achieve diverse graph structures without changing the number of nodes or their attribute values, by randomly removing and adding edges. We derive 10 graphs of different density from each of the 10 base graphs in graph families A, B, and C, resulting in 100 graphs for each experiment.

Table 1. Number of authors per publications at 23 conferences.

Number of Publications	Number of Authors
1	44905
2	11940
3	5398
4	3144
5	2078
6-10	4421
11-20	2282
> 20	1488
78350	**75656**

4.2. DBLP Data

The bibliography dataset provided by DBLP (`http://dblp.uni-trier.de/db/`) is a rich dataset comprising several informations about publications in the computer science area. For our assessment, we sample a dataset containing just the publications from 23 related conferences, namely: AAAI, IAAI, CIKM, CVPR, ECIR, ECML, PKDD, EDBT, ICDT, ICDE, ICDM, ICML, IJCAI, KDD, PAKDD, PODS, SDM, SIGIR, SIGMOD, SSDBM, VLDB, WSDM, and WWW. We select authors with at least 5 publications to compose the node set of our graph models, leading to 10269 nodes. In addition, we describe each author through 47 different attributes that represent the ratio of the author's publication in each conference over the total amount of the author's publication (23 attributes), the ratio of the author's publication in each conference over the total amount of conference publications (23 attributes), and the normalized amount of publications (1 attribute).

Table 1 shows a relation between number of publications and number of authors. Since we selected only authors that have at least 5 publications, this amounts to 13.57% of the DBLP dataset regarding the 23 conferences mentioned. Figure 2 shows for each conference the number of publications and unique authors. CVPR conference has the highest amount of publications, but AAAI has the highest amount of unique authors. On the other side, IAAI has the lowest amount of publications and ICDT the lowest amount of unique authors.

As we select authors that have at least 5 publications in the 23 conferences, our modeled graphs may present scenarios in which a vertex has no connections, i.e., it is an isolated node in the graph. Consider, for example, that one author has 5 publications and no co-authorship. In this case, the author would be represented by an isolated node in the co-authorship graph. We consider such isolated nodes as ranked last since the outlier detection algorithms used do not handle such scenarios with isolated nodes.

While we focused on quantitatively different graphs in the synthetic data, here we derive four *semantically different* graph models from the DBLP data:

1. Co-authorship: If two authors have at least one publication together, they will have an edge connecting them.
2. Correlation between publications in each conference: Two authors are connected if they have a high correlation of their distribution of publications over the conferences (i.e., they tend to publish at the same venues).

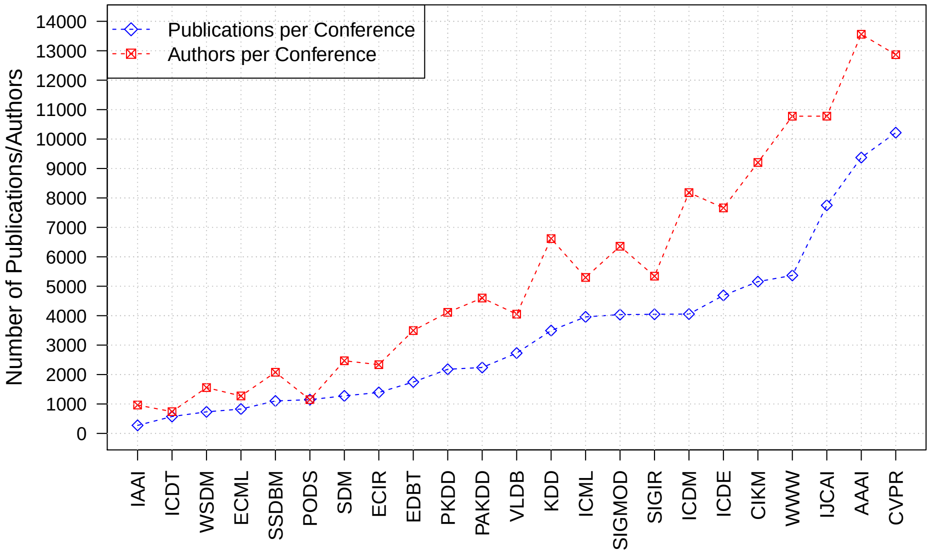


Fig. 2. Authors and publications per conference.

3. Correlation between publications in each area of knowledge: In order to model knowledge areas, we group conferences into five areas: Artificial Intelligence, Data Mining, Databases, Information Retrieval, and Web. In this model, two authors are connected if they have a high correlation between publications in venues from similar areas of knowledge.

   A correlation threshold may be seen as a parameter of graph density: the lower the threshold, the denser the graph. In models 2 and 3, we used 0.9 as correlation threshold to determine the existence of an edge.

4. Similarity in topics: Two authors are connected if they have publications with similar title words. We determine the set of unique words used in all publications of each author, removing all stop words, applying Porter's stemming algorithm [31], and selecting all unique words for each author. The edge between two authors exists if the overlap between their vocabularies is at least 40%, that is, if the intersection between their title words is at least 40% of the smaller of both vocabularies.

For sake of quantitative evaluation, we add 50 artificial outliers by randomly selecting 50 times 2 to 5 nodes that we merge to generate one outlier. Such merging imitates the effect of having two (or more) authors with the same name in the database without name-disambiguation such that their publication profiles are merged in DBLP, which is a quite realistic problem. These merged nodes are the same across all graph models. We repeat the artificial outlier generation 10 times, resulting in 10 DBLP datasets with different outliers.

4.3. Citation Network Data

The Citation Network data was extracted from ACM and provided at `https://aminer.org/citation` [39]. It comprises information about publications in many venues. For each paper, it provides id, title, abstract, year of publication, authors, venue, and references. The main difference between DBLP and Citation Network is that in the first one the node set represents authors, while here nodes are papers.

We preprocessed the original Citation Network ACM version 09 dataset, which was collected until 2017 by removing publications with missing information, selecting authors with at least 10 publications, and conferences with at least 500 papers to compose the node set of our graph models. After this pre-processing step, the remaining 16,197 papers represent our Citation Network dataset.

In addition, we describe each paper through 6 different attributes derived from the information given on the original dataset: year of publication, average authors per publications, average authors per publications in the conference where the paper was published, number of conference papers accepted in that year, and total amount of papers in the conference.

We derive four *semantically different* graph models from the Citation Network data:

1. Reference: Connect two papers if at least one of them references the other.
2. Co-authorship: If two papers share at least one author, they are connected by an edge.
3. Similarity in abstract: Similar to the DBLP model 4, we determined the set of unique words used in each paper's abstract, removed all stop words, applied Porter's stemming algorithm [31], and selected all unique words for each paper. If two papers share at least 40% of their abstract vocabulary, we connect them with an edge.
4. Similarity in topics: This is the same as DBLP model 4. We perform the same approach for word processing and connect two papers if they share at least 40% of their title words.

Here we include 50 realistic outliers to simulate effects of plagiarism. We selected 50 different papers and copied their title and abstract information, changing the authors, the venue and the year. To select the authors and the venue, we choose by random from the pool of possible choices that is present in our preprocessed dataset. The year we select randomly from the orginal publication until 2017. We repeat the artificial outlier generation 10 times, resulting in 10 Citation Network datasets with different outliers.

Each paper has attributes that are derived from the year, the authors, and the venue. The values of these attributes are suspicious for outliers as we generate them by selecting authors, venue, and year randomly. We expect that different graph models change the vicinity of outlier nodes in different aspects, so that the 'plagiarized' papers are considered a more significant outlier in some models. For example, in a graph where papers are connected by similar title words, the authors that plagiarized a paper should have a different behavior of publications than other authors that published in similar topics.

4.4. Facebook Data

We use the Facebook dataset provided by Leskovec and Krevl [21]. This dataset consists of 10 ego nodes and their friends from Facebook. Each ego and its friend list have a set of attributes related to, for example, age, gender, education, or work. We selected the 3 largest ego nodes (ids 107, 1684, and 1912) and their friends to generate the dataset used in this experiment. This was done in order to group nodes that have similar sets of attributes, since not all 10 ego nodes have the same attribute set. This dataset has 2,573 nodes in total. The IDs and feature vectors of this dataset have been anonymized. 88 attributes describe aspects such as education, work, gender, location, language, and birthday.

We derived 3 semantically different graphs to represent this dataset:

1. connecting people according to their friendship,
2. connecting people that have correlated education features, and
3. connecting people that have correlated work-related features.

As for the DBLP models 2 and 3, we use 0.9 as correlation threshold.

The outlier generation process was performed the same way as for the DBLP dataset, randomly selecting 2 to 5 nodes to merge and to generate an outlier. We added 50 outliers to the dataset, repeating the random procedure 10 times, resulting in 10 Facebook datasets with different outliers.

4.5. Outlier Detection Methods

To perform outlier detection on each individual graph, we use ConOut [35] and Radar [22]. On ConOut, we use similarity parameter 0.5, and a significance level parameter 0.1 as suggested by Sánchez et al. [35]. The similarity parameter indicates the threshold to consider two nodes similar or not. The significance parameter indicates whether we accept the statistical F-test or reject it (as described on Section 2.1). In other words, if the p-value is below the significance parameter, the attribute has lower variance inside the context comparing to the whole dataset, thus we add it to the set of relevant attributes.

On Radar, we use $\alpha = 0.5$, $\beta = 0.2$ and $\gamma = 0.2$ as suggested by the authors [22]. These parameters control the row sparsity of the coefficient matrix, row sparsity of the residual matrix and the contribution of the network modeling, respectively.

ConOut and Radar are examples of outlier detection methods in static graphs that output scores of outlierness for each node. This type of methods is suitable to our scenario as we, in a posterior step, combine these outputs into a single score. Any other outlier detection method may be used after adjusting the combination technique.

4.6. Combination

To combine the results obtained from different graph models derived from the same dataset into a single outlier score for each node we use two rank and two score combination procedures. In this particular scenario, we do not need a normalization step for score combination since we apply the same algorithm with same parameters on different graphs. We combine scores by taking the average value and the maximum value.

We also transform the scores into rankings and aggregate them by using the median and the Borda Count algorithm [4]. The Borda Count method is a classic voting system that can be applied to combine rankings from different sources. This method gives a score to each member of the list according to its relative position. The final aggregated ranking is a simple sorted list based on the sum of scores of each member. This is equivalent to combining the rankings by their mean rank.

Table 2. Average ROC AUC values on Power Law (Exp. A) synthetic data experiments when we select the most dissimilar models to combine in each iteration. For single graph models (ensemble size 1), we take the average of all possibilities.

Ens. Size	ConOut				Radar			
	Mean	Max	Median	Borda	Mean	Max	Median	Borda
	Jaccard Similarity							
1	0.633				**0.655**			
2	0.671	0.634	0.668	0.668	0.652	0.651	0.654	0.654
3	0.707	**0.638**	0.681	0.707	0.651	0.651	**0.657**	0.655
4	0.728	0.634	0.719	0.732	0.651	0.651	**0.657**	**0.656**
5	0.746	0.627	0.729	0.754	0.651	0.651	**0.657**	0.655
6	0.749	0.612	0.744	0.759	0.651	0.651	**0.657**	0.655
7	0.756	0.612	0.742	0.766	0.652	0.651	0.656	0.655
8	**0.757**	0.600	**0.753**	**0.767**	0.652	0.651	0.656	0.655
9	**0.757**	0.590	0.750	**0.767**	0.652	0.651	0.656	0.655
10	0.748	0.580	0.742	0.761	0.652	0.651	0.655	0.654
Ens. Size	DeltaCon Similarity							
1	**0.633**				**0.655**			
2	0.672	0.630	0.681	0.681	0.652	0.651	0.653	0.653
3	0.688	0.580	0.668	0.699	0.650	0.651	0.651	0.652
4	0.705	0.577	0.701	0.717	0.650	0.651	0.654	0.654
5	0.714	0.577	0.703	0.724	0.651	0.651	**0.655**	0.654
6	0.721	0.580	0.716	0.733	0.651	0.651	**0.655**	0.654
7	0.728	0.576	0.718	0.740	0.651	0.651	**0.655**	0.654
8	0.736	0.581	0.729	0.748	0.651	0.651	**0.655**	0.654
9	0.741	0.580	0.725	0.752	0.651	0.651	**0.655**	0.654
10	**0.748**	0.580	**0.742**	**0.761**	0.652	0.651	**0.655**	0.654

To assess the impact of graph similarity on the quality of the combined results we use Jaccard similarity (Equation 1) and DeltaCon similarity (Equation 5). Combinations of more than two graph models are ranked by the average pairwise Jaccard similarity (Equation 2) and average pairwised DeltaCon similarity (Equation 6).

5. Results and Discussion

5.1. Synthetic Data

The results for the synthetic datasets are shown in Tables 2, 3, and 4 for experiments A, B and C respectively. These tables shows quantitatively the impact on outlier detection performance if we take the most dissimilar models to combine the results obtained on those. The ensemble size is the number of combined graph models, where 1 relates to a single graph model (i.e., no combination but individual performances on all graphs) and 2 to 10 relate to the corresponding number of combined multiple graph models. The performance is quantified in terms of the average Area Under the Curve of the Receiver Operating Characteristic (ROC AUC), a standard measure for outlier detection performance [5].

In all three experimental scenarios, using multiple graph models generally improves over using individual graphs only. Radar seems to be a more robust method than ConOut

Table 3. Average ROC AUC values on Barabási and Albert (Exp. B) synthetic data experiments when we select the most dissimilar models to combine in each iteration. For single graph models (ensemble size 1), we take the average of all possibilities.

Ens. Size	ConOut				Radar			
	Mean	Max	Median	Borda	Mean	Max	Median	Borda
	Jaccard Similarity							
1	0.647				**0.762**			
2	0.792	0.757	0.751	0.751	0.753	0.753	0.758	0.758
3	0.813	0.749	0.762	0.778	0.754	0.753	0.762	0.760
4	0.835	**0.760**	0.793	0.806	0.754	0.753	0.762	0.760
5	0.842	0.757	0.786	0.815	0.754	0.753	**0.763**	0.761
6	0.849	0.755	0.807	0.823	0.755	0.753	**0.763**	0.761
7	0.855	0.754	0.802	0.830	0.755	0.753	**0.763**	0.761
8	0.858	0.754	0.813	0.832	0.755	0.753	**0.763**	**0.762**
9	0.861	0.755	0.811	0.835	0.755	0.753	**0.763**	**0.762**
10	**0.863**	0.756	**0.814**	**0.837**	0.755	0.753	**0.763**	**0.762**
Ens. Size	DeltaCon Similarity							
1	0.647				**0.762**			
2	0.784	0.747	0.752	0.752	0.753	0.753	0.758	0.758
3	0.825	0.762	0.769	0.797	0.753	0.753	0.761	0.759
4	0.830	0.758	0.784	0.797	0.754	0.753	0.762	0.760
5	0.845	**0.760**	0.786	0.817	0.754	0.753	0.762	0.761
6	0.850	0.758	0.806	0.825	0.754	0.753	0.762	0.761
7	0.852	0.759	0.793	0.825	0.755	0.753	**0.763**	0.761
8	0.855	0.757	0.806	0.827	0.755	0.753	**0.763**	**0.762**
9	0.858	0.757	0.801	0.831	0.755	0.753	**0.763**	**0.762**
10	**0.863**	0.756	**0.814**	**0.837**	0.755	0.753	**0.763**	**0.762**

as its results have lower variation. It is out of our scope to analyze different parameter values for the outlier detection algorithms, as we are interested on the effects of applying multiple graph models on the same raw dataset. As mentioned before, the γ parameter on Radar balances the contribution of attribute and network information. On our experiments, we fix γ as suggested by the authors and, as a consequence, the results show robustness towards multiple graph models, as the variations on the edges do not have a high impact on the final Radar output.

ConOut is largely benefited by the usage of multiple graph models. Using ConOut on single graphs, average ROC AUC reaches 0.633, 0.647, and 0.637 on experiments A, B and C respectively. In comparison with multiple graph models, these numbers go as high as 0.767, 0.863, and 0.814. On most cases, combining all 10 outputs from different graphs achieves the highest ROC AUC value, which indicates that using multiple graph models has large potential to be applied in many different scenarios, as we have 3 different synthetic experiments that express real-world graph behaviors.

Regardless of the algorithm applied, combine scores using the maximum value do not perform well in general. Experiment C using Radar algorithm is the only specific scenario where combining by maximum has the best average ROC AUC value, 0.712. Mean, median and borda have a very similar behavior when used in conjunction with the

Table 4. Average ROC AUC values on Erdős and Rényi (Exp. C) synthetic data experiments when we select the most dissimilar models to combine in each iteration. For single graph models (ensemble size 1), we take the average of all possibilities.

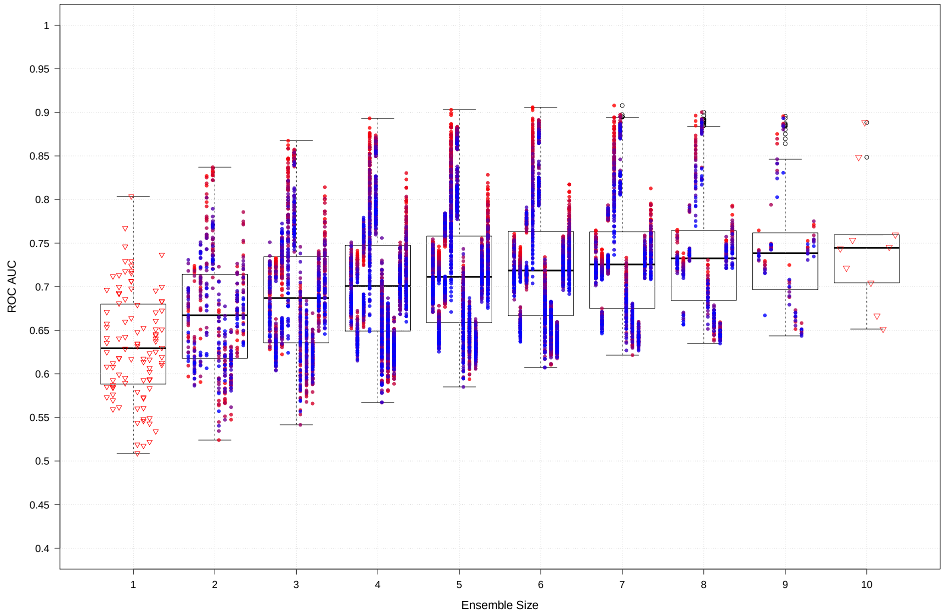
Ens. Size	ConOut				Radar			
	Mean	Max	Median	Borda	Mean	Max	Median	Borda
	Jaccard Similarity							
1	0.637				**0.701**			
2	0.753	0.720	0.727	0.727	0.705	0.707	0.700	0.700
3	0.750	0.696	0.713	0.726	0.703	0.704	0.695	0.698
4	0.783	0.717	0.749	0.758	0.703	0.705	0.697	0.699
5	0.798	0.726	0.740	0.770	0.703	0.705	0.697	0.698
6	0.805	0.726	0.760	0.778	0.703	0.705	0.697	0.698
7	0.807	0.734	0.759	0.780	0.704	0.707	0.697	0.699
8	0.808	0.739	**0.766**	**0.781**	0.704	0.707	0.698	0.699
9	**0.814**	0.750	0.760	**0.781**	0.706	0.711	0.698	0.699
10	0.807	**0.758**	0.758	0.774	**0.707**	**0.712**	0.699	**0.701**
Ens. Size	DeltaCon Similarity							
1	0.637				**0.701**			
2	0.741	0.712	0.718	0.718	0.705	0.706	**0.701**	0.701
3	0.778	0.731	0.727	0.751	0.707	0.709	**0.701**	**0.702**
4	0.791	0.732	0.754	0.761	**0.708**	0.710	0.699	0.701
5	0.807	0.748	0.754	0.778	**0.708**	0.711	0.700	**0.702**
6	0.806	0.740	0.762	0.776	0.707	0.711	0.699	0.701
7	0.812	0.748	0.758	**0.782**	0.707	0.711	0.698	0.700
8	0.809	0.752	**0.765**	0.779	**0.708**	**0.712**	0.698	0.701
9	**0.813**	0.756	0.758	**0.782**	**0.708**	**0.712**	0.698	0.701
10	0.807	**0.758**	0.758	0.774	0.707	**0.712**	0.699	0.701

ConOut algorithm. These combination approaches yield their highest values using similar ensemble sizes on each experiment, which again supports the argument for using multiple graph models rather than single graphs.

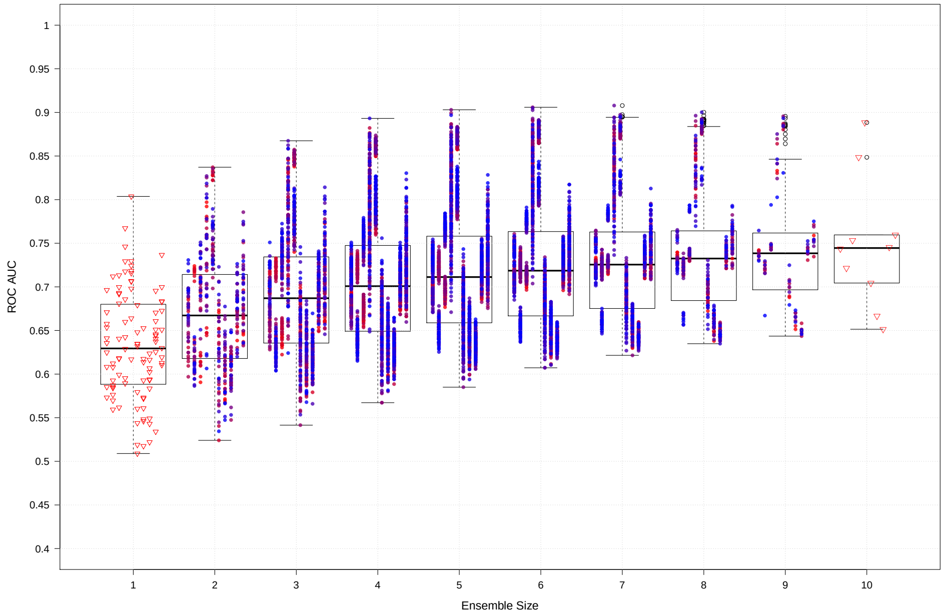
The results do not change much between using Jaccard similarity and DeltaCon similarity.

From Tables 2, 3, and 4 we can infer that using the most dissimilar models according to Jaccard and DeltaCon to represent the same dataset in general improves the performance on detecting outliers. Figures 3, 4 and 5 shows results using ConOut on synthetic experiments. The colors of the points represent how similar the combined graph models are, ranging from red to blue, where blue is more similar and red is less similar. Each experiment is represented by one plot using Jaccard similarity and one plot using DeltaCon similarity. We expect more red points on the top of each boxplot, which means that the combination of dissimilar graphs delivers better results than combining similar graphs. Since we have 10 independent subsets for each graph family, each column of points inside the boxplots shows the results for one of these subsets.

All three experiments show more red dots on the top of boxplots, especially when Jaccard is selected. The behavior of boxplots reinforces our claim on using multiple graph

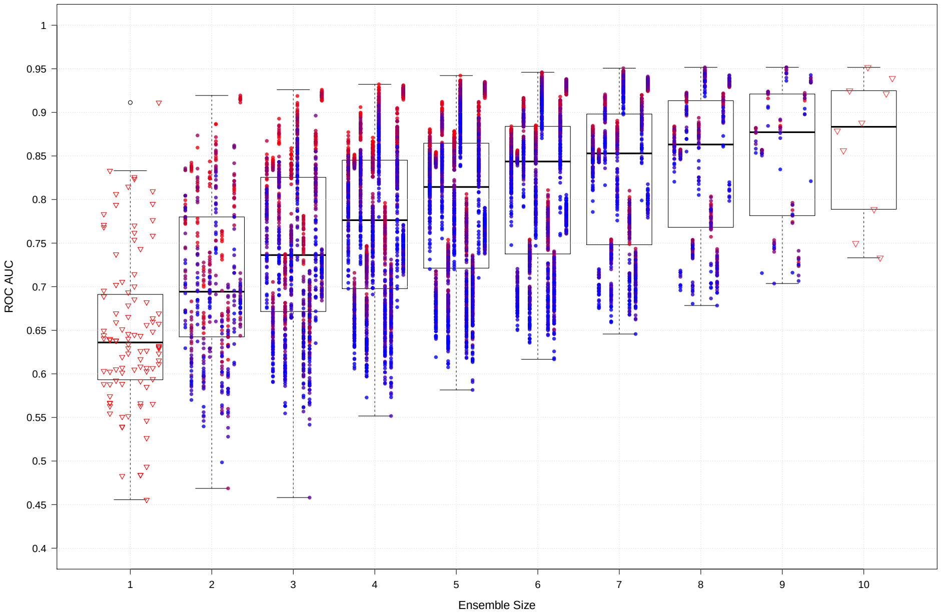


(a) Experiment A: Power-law degree distribution with ConOut using Jaccard similarity measure

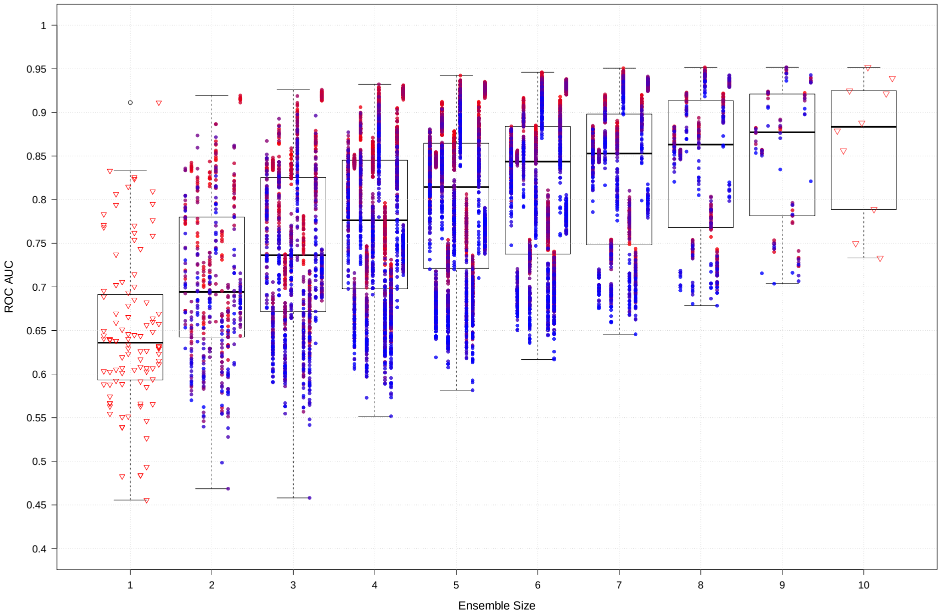


(b) Experiment A: Power-law degree distribution with ConOut using DeltaCon similarity measure

Fig. 3. ROC AUC on single and multiple graph models on experiment A synthetic datasets as we vary the ensemble size (number of combined graph models, where 1 represents single graph models). The colors of the points reflect the (average) similarity of the graph models selected for combination: blue (more similar) to red (less similar). Each column of points inside each boxplot presents the results for an independent subset of graphs.

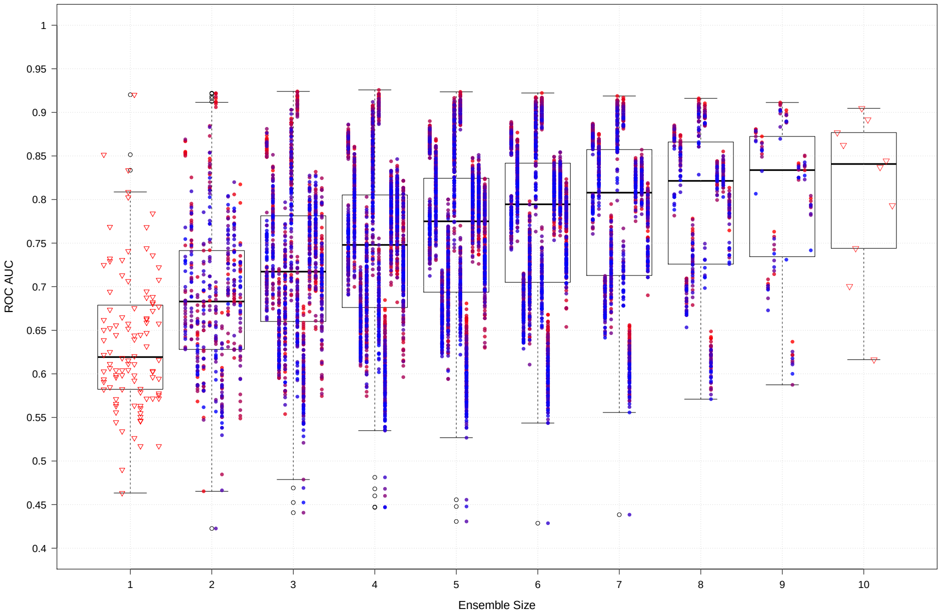


(a) Experiment B: Barabási and Albert stochastic model with ConOut using Jaccard similarity measure

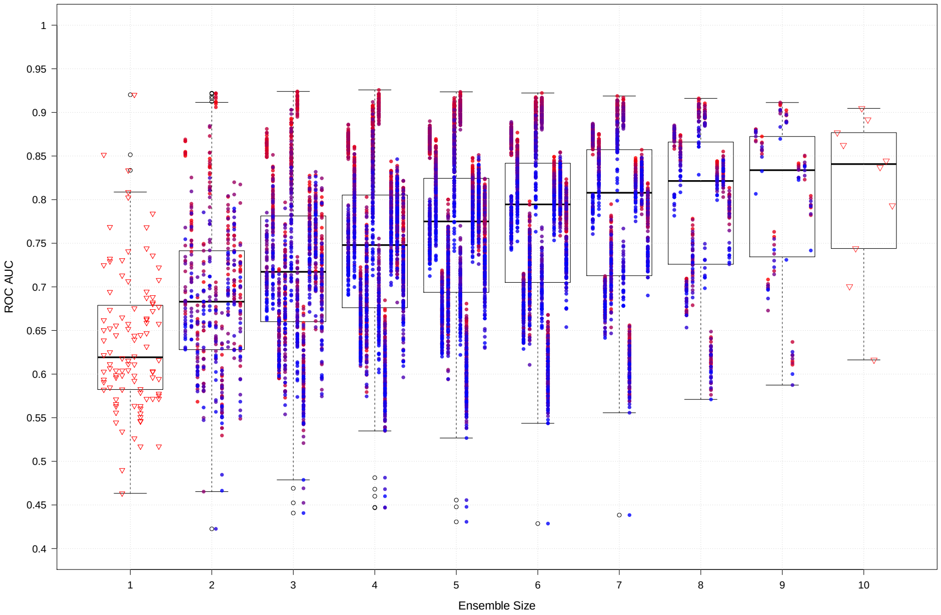


(b) Experiment B: Barabási and Albert stochastic model with ConOut using DeltaCon similarity measure

Fig. 4. ROC AUC on single and multiple graph models on experiment B synthetic datasets as we vary the ensemble size (number of combined graph models, where 1 represents single graph models). The colors of the points reflect the (average) similarity of the graph models selected for combination: blue (more similar) to red (less similar). Each column of points inside each boxplot presents the results for an independent subset of graphs.



(a) Experiment C: Erdős and Rényi model with ConOut using Jaccard similarity measure



(b) Experiment C: Erdős and Rényi model with ConOut using DeltaCon similarity measure

Fig. 5. ROC AUC on single and multiple graph models on experiment C synthetic datasets as we vary the ensemble size (number of combined graph models, where 1 represents single graph models). The colors of the points reflect the (average) similarity of the graph models selected for combination: blue (more similar) to red (less similar). Each column of points inside each boxplot presents the results for an independent subset of graphs.

Table 5. Average ROC AUC values on DBLP data experiments when we select the most dissimilar models to combine in each iteration. For single graph models (ensemble size 1), we take the average of all possibilities.

Ens. Size	ConOut				Radar			
	Mean	Max	Median	Borda	Mean	Max	Median	Borda
	Jaccard Similarity							
1	0.840				**0.506**			
2	0.890	0.874	0.896	0.896	0.501	**0.510**	0.511	0.511
3	0.898	0.877	0.888	0.903	0.505	0.505	0.497	0.504
4	**0.953**	**0.934**	**0.939**	**0.943**	0.495	0.499	**0.527**	**0.527**
	DeltaCon Similarity							
1	0.840				**0.506**			
2	0.890	0.874	0.896	0.896	0.501	**0.510**	0.511	0.511
3	0.950	**0.934**	0.927	0.933	0.502	0.494	**0.528**	**0.534**
4	**0.953**	**0.934**	**0.939**	**0.943**	0.495	0.499	0.527	0.527

models for outlier detection, as the ROC AUC increases when we select a larger ensemble size.

Overall, using multiple graph models achieves better performance than using single graph models, regardless of the similarity parameter threshold, of the number of models, and of which models are combined: the tendency of the results shows that the performance on multiple graph models is correlated to the performance of the individual models we combine.

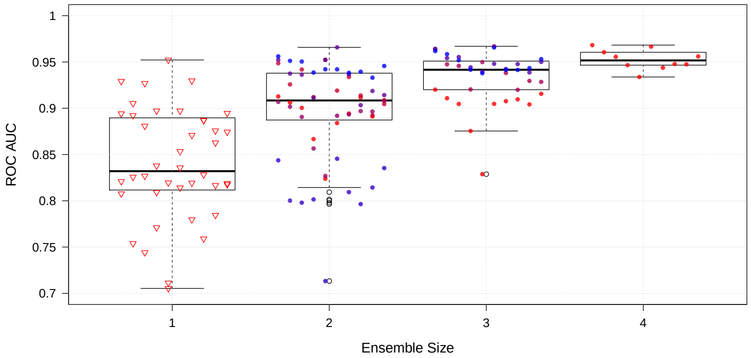
Multiple graph models represent different aspects of the data, where each model potentially highlights different outliers. When we combine the outlier detection results obtained over multiple graph models we benefit from this diversity.

5.2. DBLP Data

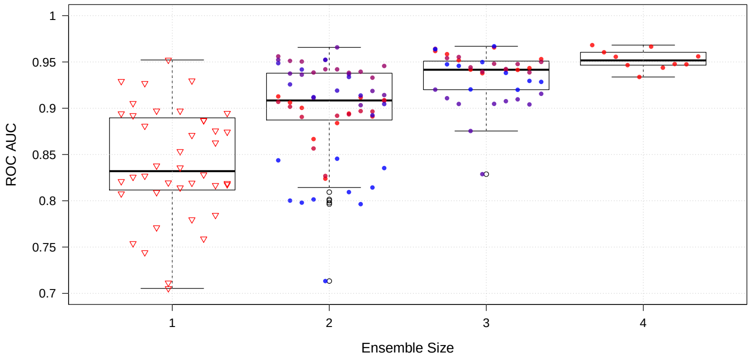
Table 5 shows the average ROC AUC results for ConOut and Radar on single and multiple graph models. We again see the robustness of the Radar algorithm. Even though Radar performs poorly on DBLP data, using multiple graph models for outlier detection improves over single graphs. ConOut reaches average ROC AUC of 0.953 in the best scenario combining all graph models outputs using mean, in comparison to 0.840 when using only single graphs to represent the data.

Figure 6 shows the results when we apply ConOut to single graph models (ensemble size 1) and to multiple graph models, using the mean as consensus function.

Although the quantitative results for the DBLP dataset are quite good in terms of precision, there are still some authors that consistently get high outlier scores together with the generated outliers (merged nodes). For the quantitative evaluation based on the artificially constructed outliers, these are considered inliers, but they might actually qualify as real outliers in some sense. Thus we inspect some examples in Figure 7, depicting the degree of outlierness for 9 authors that consistently present high scores. These plots measure the degree of outlierness among 4 single graph models (DBLP 1 to 4) and for multiple graph models, represented by the most dissimilar models (Ensemble 2 to 4). We average the rankings of 10 iterations and transform the final ranking into bins. There are



(a) DBLP with ConOut using Jaccard similarity measure



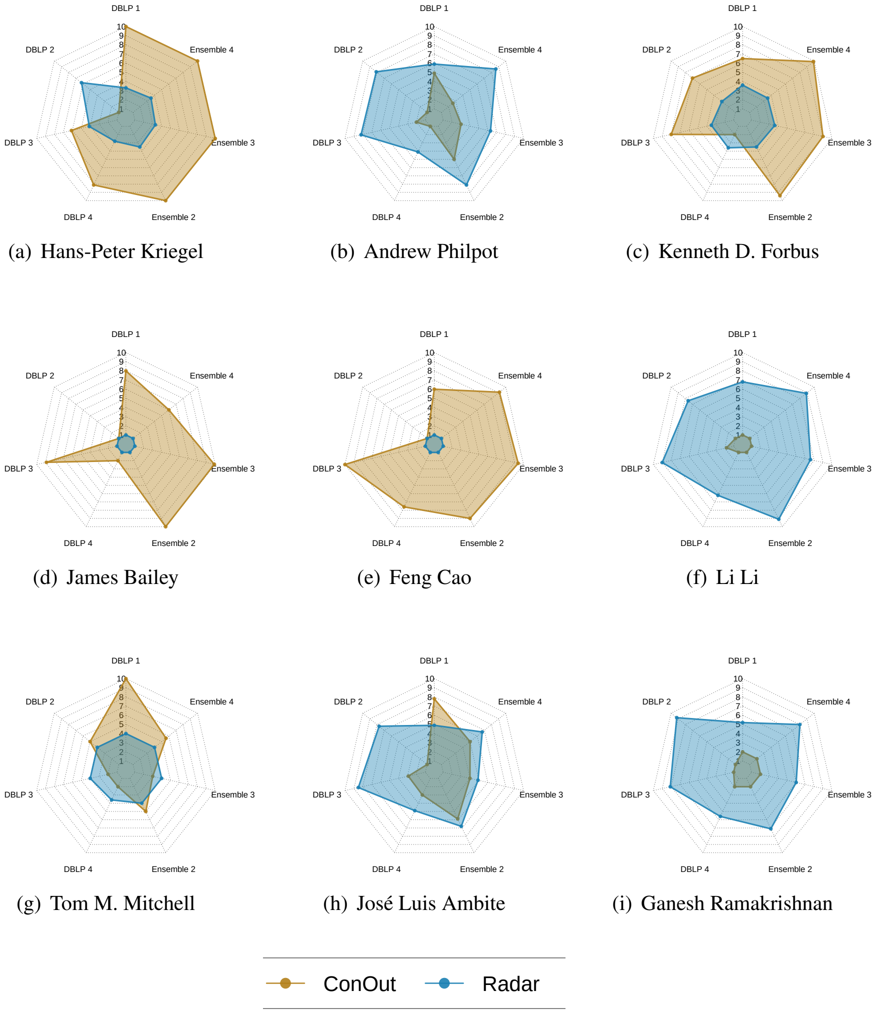
(b) DBLP with ConOut using DeltaCon similarity measure

Fig. 6. ROC AUC on single and multiple graph models on DBLP dataset as we vary the ensemble size (number of combined graph models, where 1 represents single graph models). The colors of the points reflect the (average) similarity of the graph models selected for combination: blue (more similar) to red (less similar). Each column of points inside each boxplot presents the results for an independent subset of graphs.

10 bins, ranging from 10 (outlier) to 1 (inlier). The bins compress the rankings of outlierness and they increase by a factor of 2 starting from bin 10 covering ranks from 1 to 5, bin 9 covering ranks 6 to 15, bin 8 covering ranks 16 to 35 etc.

Hans-Peter Kriegel is consistently the most prominent outlier node according to ConOut in all DBLP models, except for model 2, in which Radar gives his largest score of outlierness among all models. His pattern is shown in Figure 7(a), where he tends to be an outlier in almost all scenarios according to ConOut.

The same pattern is also present for Feng Cao's and Kenneth D. Forbus's plots, Figures 7(e) and 7(c) respectively. Feng is a top outlier in the DBLP 3 model according to ConOut and exhibits a high degree of outlierness in other single and multiple graph models, but is not a prominent outlier in the DBLP 2 model. Kenneth is not among top 5 of any single graph model, but when we combine multiple graph models, he become top outlier



(a) Hans-Peter Kriegel

(b) Andrew Philpot

(c) Kenneth D. Forbus

(d) James Bailey

(e) Feng Cao

(f) Li Li

(g) Tom M. Mitchell

(h) José Luis Ambite

(i) Ganesh Ramakrishnan

Fig. 7. These plots depict the degree of outlierness for some natural outliers in the DBLP dataset using ConOut and Radar algorithms. The rankings were averaged over 10 iterations and multiple graph models comprise the most dissimilar graphs according to Jaccard. We transform the final ranking into bins, ranging from 10 to 1, where 10 refers to top outlier scores and 1 to inliers scores. We increase the bins range by a factor of 2, starting from bin 10 covering rankings [1 - 5] and ending on bin 1 covering rankings [2556 - Maximum]. Ensembles 2 to 4 correspond to multiple graph models.

Table 6. Average ROC AUC values on Citation Network data experiments when we select the most dissimilar models to combine in each iteration. For single graph models (ensemble size 1), we take the average of all possibilities.

Ens. Size	ConOut				Radar			
	Mean	Max	Median	Borda	Mean	Max	Median	Borda
	Jaccard Similarity							
1	**0.590**				**0.659**			
2	0.580	0.577	0.545	0.545	0.658	**0.659**	**0.659**	0.658
3	**0.673**	**0.691**	0.570	**0.608**	0.658	**0.659**	**0.659**	**0.659**
4	0.632	0.642	0.517	0.526	**0.659**	**0.659**	**0.659**	**0.659**
	DeltaCon Similarity							
1	**0.590**				**0.659**			
2	**0.646**	**0.723**	0.581	0.581	0.658	0.658	0.658	0.658
3	0.610	0.662	0.510	0.557	**0.659**	**0.659**	**0.659**	**0.659**
4	0.632	0.642	0.517	0.526	**0.659**	**0.659**	**0.659**	**0.659**

according to ConOut. Radar labels Feng Cao as inlier in each model and gives a slightly higher degree of outlierness to Kenneth D. Forbus.

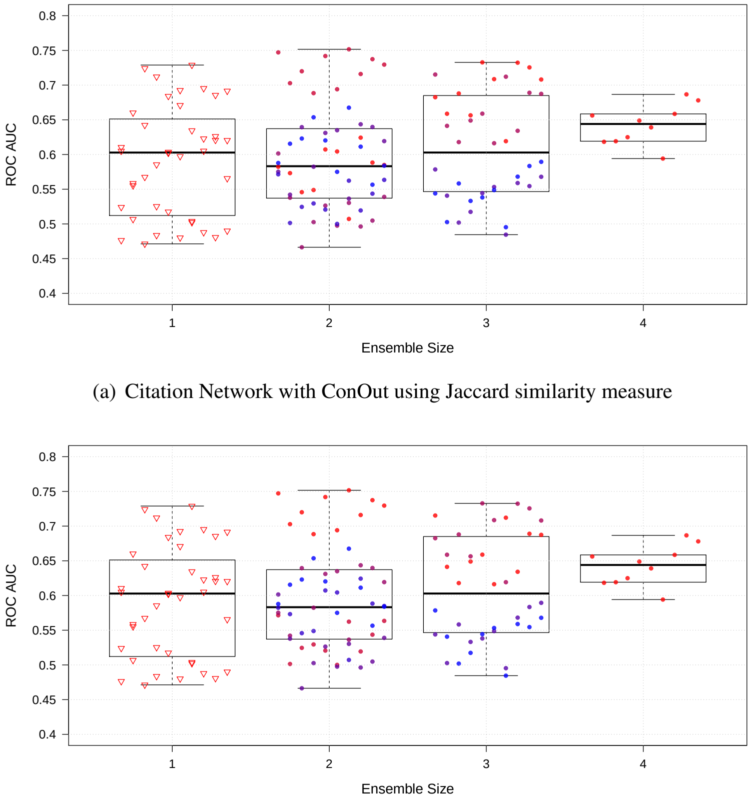
James Bailey and Feng Cao have opossite behavior when compared to Li Li and Ganesh Ramakrishnan regarding the algorithm used to output their degree of outlierness. On the first two, ConOut labels them as outliers and Radar as inliers. On the other hand, Li Li and Ganesh are outliers in Radar's perspective and inliers in ConOut's perspective.

José Luis Ambite and Tom M. Mitchell are clear examples of nodes that are outliers only in a single graph model (DBLP 1) according to ConOut. When we combine dissimilar models, they do not appear unusual anymore. Radar seems very consistent on these two cases, specially on José Luis Ambite, where it outputs high degree of outlierness on models 2 and 3, but slightly reduce his score when we combine multiple graph models.

5.3. Citation Network Data

Table 6 shows the average ROC AUC results for ConOut and Radar on single and multiple graph models. We again see the robustness of the Radar algorithm towards different models to represent the same data. DeltaCon similarity measure selects better dissimilar models to combine their outputs than Jaccard on Citation Network when selecting two graph models. With Jaccard best results are achieved when selecting three models. The highest value using multiple graph models is 0.723 average ROC AUC combining 2 most dissimilar models according to Jaccard using ConOut algorithm and combine by maximum score.

Figure 8 shows the results when we apply ConOut to single graph models (ensemble size 1) and to multiple graph models, using borda as combination function. We repeat the same procedure to generate Figure 6, where the colors of the points are set as for the experiments on synthetic data (blue: combination of more similar graph models, red: less similar). Each column within each boxplot relates to one of the 10 dataset variants (different outliers). In Figures 8(a) and 8(b) the boxplots shows the benefits of using multiple graph models for outlier detection task. On Citation Network, the red dots are clearly on top of each experiment, which holds our claim towards selecting more diverse



(a) Citation Network with ConOut using Jaccard similarity measure

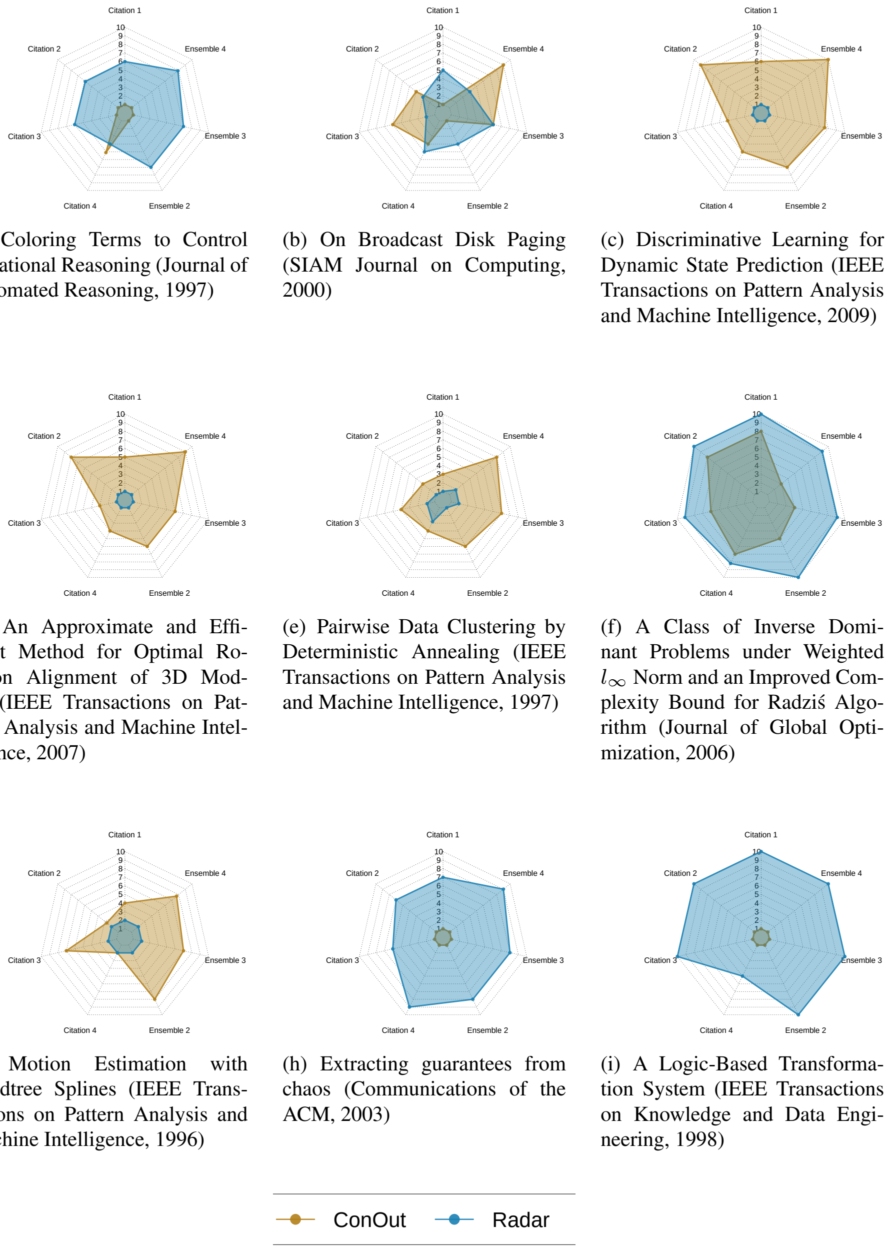
(b) Citation Network with ConOut using DeltaCon similarity measure

Fig. 8. ROC AUC on single and multiple graph models on Citation Network dataset as we vary the ensemble size (number of combined graph models, where 1 represents single graph models). The colors of the points reflect the (average) similarity of the graph models selected for combination: blue (more similar) to red (less similar). Each column of points inside each boxplot presents the results for an independent subset of graphs.

graphs to combine their outputs. Even though when selecting ensemble size equal to 2 in Figure 8(a) the red dots are in the middle of the boxplot, on the top of each experiment are the second most dissimilar models. This effect is shown in Table 6, where Jaccard best results are selecting ensemble size 3.

Together with true outliers (‘plagiarized’ papers) there are some papers that consistently get high outlier scores. Figure 9 measure the degree of outlierness among 4 single graph models (Citation Network 1 to 4) and for multiple graph models, represented by the most dissimilar models (Ensembles 2 to 4). We average the rankings of 10 iterations and transform the final ranking into bins. There are 10 bins, ranging from 10 (outlier) to 1 (inlier).

The behavior present in Figures 9(c) and 9(d) is quite similar. Radar labels both as true inliers and ConOut shows a high degree of outlierness on ensemble size 4. Papers



(a) Coloring Terms to Control Equational Reasoning (Journal of Automated Reasoning, 1997)

(b) On Broadcast Disk Paging (SIAM Journal on Computing, 2000)

(c) Discriminative Learning for Dynamic State Prediction (IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009)

(d) An Approximate and Efficient Method for Optimal Rotation Alignment of 3D Models (IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007)

(e) Pairwise Data Clustering by Deterministic Annealing (IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997)

(f) A Class of Inverse Dominant Problems under Weighted l_∞ Norm and an Improved Complexity Bound for Radziś Algorithm (Journal of Global Optimization, 2006)

(g) Motion Estimation with Quadtree Splines (IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996)

(h) Extracting guarantees from chaos (Communications of the ACM, 2003)

(i) A Logic-Based Transformation System (IEEE Transactions on Knowledge and Data Engineering, 1998)

Fig. 9. These plots depict the degree of outlierness for some natural outliers in the Citation Network dataset using ConOut and Radar algorithms. The rankings were averaged over 10 iterations and multiple graph models comprise the most dissimilar graphs according to Jaccard. We transform the final ranking into bins, ranging from 10 to 1, where 10 refers to top outlier scores and 1 to inliers scores. We increase the bins range by a factor of 2, starting from bin 10 covering rankings [1 - 5] and ending on bin 1 covering rankings [2556 - Maximum]. Ensembles 2 to 4 correspond to multiple graph models.

Table 7. Average ROC AUC values on Facebook data experiments when we select the most dissimilar models to combine in each iteration. For single graph models (ensemble size 1), we take the average of all possibilities.

Ens. Size	ConOut				Radar			
	Mean	Max	Median	Borda	Mean	Max	Median	Borda
	Jaccard Similarity							
1	0.919				0.678			
2	0.934	0.919	0.931	0.931	0.686	0.686	**0.685**	**0.685**
3	**0.974**	**0.959**	**0.951**	**0.963**	**0.687**	**0.688**	0.674	0.684
	DeltaCon Similarity							
1	0.919				0.678			
2	0.934	0.919	0.931	0.931	0.686	0.686	**0.685**	**0.685**
3	**0.974**	**0.959**	**0.951**	**0.963**	**0.687**	**0.688**	0.674	0.684

in Figures 9(e) and 9(g) have their degree of outlierness increased only using multiple models, as they are inliers in a single model approach using ConOut. In Figure 9(b), both algorithms have different results considering different graph models. It is only considered an outlier when using the combination of outputs of all graph models with the ConOut algorithm.

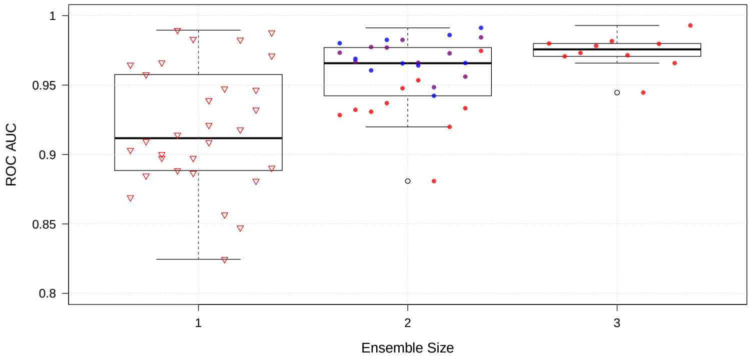
With the papers in Figures 9(a), 9(h) and 9(i) we have inliers according to ConOut and top outliers according to Radar. We see, especially with the paper "A Logic-Based Transformation System", a high degree of outlierness defined by Radar algorithm on single and multiple models. The paper in Figure 9(f) is labeled as outlier by ConOut and Radar.

5.4. Facebook Data

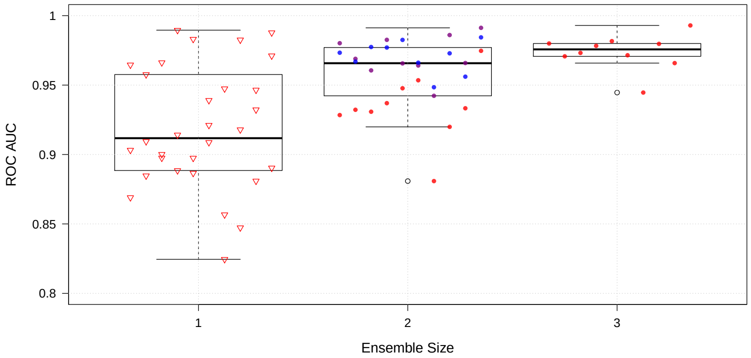
Table 7 shows the average ROC AUC results for ConOut and Radar on single and multiple graph models. Again, all results shows that using multiple graph models improves the outlier detection performance. ConOut shows superior performance compared to Radar, reaching average ROC AUC of 0.974 in the best scenario combining all graph models outputs, in comparison to 0.919 when using only single graphs to represent the data. The best combination technique in this experiment is taking the average score to represent the final degree of outlierness.

Figure 10 shows the ROC AUC results for ConOut on single and multiple graph models, using mean as consensus function. The results are over 10 variants and the columns inside each boxplot show the results for each iteration. Even though the experiments suggest the selection of as many models you have to represent the data, in this specific scenario, combining 2 dissimilar models have lower performance than combining similar models.

Findings on DBLP and Citation Network data can also be applied here. We observe on Facebook data that using multiple graph models improves the outlier detection performance.



(a) Facebook with ConOut using Jaccard similarity measure



(b) Facebook with ConOut using DeltaCon similarity measure

Fig. 10. ROC AUC on single and multiple graph models on Facebook dataset as we vary the ensemble size (number of combined graph models, where 1 represents single graph models). The colors of the points reflect the (average) similarity of the graph models selected for combination: blue (more similar) to red (less similar). Each column of points inside each boxplot presents the results for an independent subset of graphs.

6. Conclusion

Outlier detection is a subjective and unsupervised task that demands good knowledge and understanding of the data. Using a single graph model of relation-rich datasets may only model some aspects of the data, thus not making proper use of potential information. Using multiple graph models may capture more and complementary information. We therefore suggest, based on our findings, to explore real world data using multiple graph models that are as complementary as possible.

In a practical application, a data analyst is interested in certain entities that lend themselves as a set of nodes in a graph representation while several attributes or inter-relational connections may be represented as edges between nodes. Instead of looking for the one and only, best-ever graph representation of some given raw data, the data analyst should therefore generate multiple graph models describing different aspects of the raw data, capturing a large variety of characteristics, or putting different emphasis on certain characteristics. That is, the graphs may differ both quantitatively (how dense they are) and qualitatively (which relationships are expressed in the graph structure). These multiple graph models aim to materialize the various perspectives that the analyst wants to highlight, that is, they should cover the problem scenario as well as possible and in as many different ways as suitable.

Clearly, many questions remain open. We focused in this study purely on the aspect of the impact of multiple graph models for a given dataset. We evaluated this impact using two different outlier detection algorithms, four combination functions, and two similarity measures on synthetic and real world data. For a practical application, various aspects will have strong influence on the achievable quality, for example the algorithm used to detect outliers on the individual graphs and the method used to combine the individual results (as we have seen in this evaluation). However based on our study we can maintain the recommendation to consider several different graph representations in any case.

Acknowledgments. This work was partially supported by CAPES - Brazil, Fapemig, CNPq, and by projects InWeb, MASWeb, EUBra-BIGSEA (H2020-EU.2.1.1 690116, Brazil/MCTI/RNP GA-000650/04), INCT-Cyber, and Atmosphere (H2020-EU 777154, Brazil/MCTI/RNP 51119).

References

1. Akoglu, L., McGlohon, M., Faloutsos, C.: oddball: Spotting anomalies in weighted graphs. In: Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part II. pp. 410–421 (2010), https://doi.org/10.1007/978-3-642-13672-6_40
2. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data Min. Knowl. Discov. 29(3), 626–688 (2015)
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509–512 (1999)
4. de Borda, J.C.: Mémoire sur les élections au scrutin. Histoire de l'Académie Royale des Sciences (1784)
5. Campos, G.O., Zimek, A., Sander, J., Campello, R.J.G.B., Micenková, B., Schubert, E., Assent, I., Houle, M.E.: On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. Data Mining and Knowledge Discovery 30, 891–927 (2016)
6. Campos, G.O., Meira Jr., W., Zimek, A.: Outlier detection in graphs: On the impact of multiple graph models. In: Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, WIMS 2018, Novi Sad, Serbia, June 25-27, 2018. pp. 21:1–21:12 (2018), http://doi.acm.org/10.1145/3227609.3227646
7. Chakrabarti, D.: Autopart: Parameter-free graph partitioning and outlier detection. In: Knowledge Discovery in Databases: PKDD 2004, 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy, September 20-24, 2004, Proceedings. pp. 112–124 (2004), https://doi.org/10.1007/978-3-540-30116-5_13
8. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection for discrete sequences: A survey. IEEE Transactions on Knowledge and Data Engineering 24(5), 823–839 (2012)
9. Erdős, P., Rényi, A.: On random graphs i. Publicationes Mathematicae (Debrecen) 6, 290–297 (1959)
10. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: Knowledge discovery and data mining: Towards a unifying framework. In: Proceedings of the 2nd ACM International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR. pp. 82–88 (1996)

11. Gao, J., Tan, P.N.: Converting output scores from outlier detection algorithms into probability estimates. In: Proceedings of the 6th IEEE International Conference on Data Mining (ICDM), Hong Kong, China. pp. 212–221 (2006)
12. Gao, J., Liang, F., Fan, W., Wang, C., Sun, Y., Han, J.: On community outliers and their efficient detection in information networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010. pp. 813–822 (2010)
13. Henderson, K., Gallagher, B., Li, L., Akoglu, L., Eliassi-Rad, T., Tong, H., Faloutsos, C.: It's who you know: graph mining using recursive structural features. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011. pp. 663–671 (2011), http://doi.acm.org/10.1145/2020408.2020512
14. Jaccard, P.: Distribution de la florine alpine dans la bassin de dranses et dans quelques regiones voisines. Bulletin de la Societe Vaudoise des Sciences Naturelles 37, 241–272 (1901)
15. Kirner, E., Schubert, E., Zimek, A.: Good and bad neighborhood approximations for outlier detection ensembles. In: Proceedings of the 10th International Conference on Similarity Search and Applications (SISAP), Munich, Germany. pp. 173–187 (2017)
16. Koutra, D., Ke, T., Kang, U., Chau, D.H., Pao, H.K., Faloutsos, C.: Unifying guilt-by-association approaches: Theorems and fast algorithms. In: Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II. pp. 245–260 (2011), https://doi.org/10.1007/978-3-642-23783-6_16
17. Koutra, D., Shah, N., Vogelstein, J.T., Gallagher, B., Faloutsos, C.: Deltacon: Principled massive-graph similarity function with attribution. TKDD 10(3), 28:1–28:43 (2016), http://doi.acm.org/10.1145/2824443
18. Kriegel, H.P., Kröger, P., Schubert, E., Zimek, A.: Outlier detection in axis-parallel subspaces of high dimensional data. In: Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Bangkok, Thailand. pp. 831–838 (2009)
19. Kriegel, H.P., Kröger, P., Schubert, E., Zimek, A.: Interpreting and unifying outlier scores. In: Proceedings of the 11th SIAM International Conference on Data Mining (SDM), Mesa, AZ. pp. 13–24 (2011)
20. Lazarevic, A., Kumar, V.: Feature bagging for outlier detection. In: Proceedings of the 11th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Chicago, IL. pp. 157–166 (2005)
21. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data (Jun 2014)
22. Li, J., Dani, H., Hu, X., Liu, H.: Radar: Residual analysis for anomaly detection in attributed networks. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. pp. 2152–2158 (2017), https://doi.org/10.24963/ijcai.2017/299
23. Liu, C., Yan, X., Yu, H., Han, J., Yu, P.S.: Mining behavior graphs for "backtrace" of non-crashing bugs. In: Proceedings of the 2005 SIAM International Conference on Data Mining, SDM 2005, Newport Beach, CA, USA, April 21-23, 2005. pp. 286–297 (2005), https://doi.org/10.1137/1.9781611972757.26
24. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD) 6(1), 3:1–39 (2012)
25. Micenková, B., McWilliams, B., Assent, I.: Learning outlier ensembles: The best of both worlds–supervised and unsupervised. In: Workshop on Outlier Detection and Description under Data Diversity (ODD2), held in conjunction with the 20th ACM International Conference on Knowledge Discovery and Data Mining, New York, NY. pp. 51–54 (2014)

26. Müller, E., Schiffer, M., Seidl, T.: Adaptive outlierness for subspace outlier ranking. In: Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM), Toronto, ON, Canada. pp. 1629–1632 (2010)
27. Müller, E., Sánchez, P.I., Mülle, Y., Böhm, K.: Ranking outlier nodes in subspaces of attributed graphs. In: Workshops Proceedings of the 29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013. pp. 216–222 (2013)
28. Nguyen, H.V., Ang, H.H., Gopalkrishnan, V.: Mining outliers with ensemble of heterogeneous detectors on random subspaces. In: Proceedings of the 15th International Conference on Database Systems for Advanced Applications (DASFAA), Tsukuba, Japan. pp. 368–383 (2010)
29. Noble, C.C., Cook, D.J.: Graph-based anomaly detection. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003. pp. 631–636 (2003)
30. Perozzi, B., Akoglu, L., Sánchez, P.I., Müller, E.: Focused clustering and outlier detection in large attributed graphs. In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014. pp. 1346–1355 (2014)
31. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
32. Rayana, S., Akoglu, L.: Less is more: Building selective anomaly ensembles. ACM Transactions on Knowledge Discovery from Data (TKDD) 10(4), 42:1–42:33 (2016)
33. Rayana, S., Zhong, W., Akoglu, L.: Sequential ensemble learning for outlier detection: A bias-variance perspective. In: Proceedings of the 16th IEEE International Conference on Data Mining (ICDM), Barcelona, Spain. pp. 1167–1172 (2016)
34. Rotabi, R., Kamath, K., Kleinberg, J.M., Sharma, A.: Detecting strong ties using network motifs. In: Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017. pp. 983–992 (2017), `http://doi.acm.org/10.1145/3041021.3055139`
35. Sánchez, P.I., Müller, E., Irmler, O., Böhm, K.: Local context selection for outlier ranking in graphs with multiple numeric node attributes. In: Conference on Scientific and Statistical Database Management, SSDBM '14, Aalborg, Denmark, June 30 - July 02, 2014. pp. 16:1–16:12 (2014)
36. Sánchez, P.I., Müller, E., Laforet, F., Keller, F., Böhm, K.: Statistical selection of congruent subspaces for mining attributed graphs. In: 2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013. pp. 647–656 (2013)
37. Schubert, E., Wojdanowski, R., Zimek, A., Kriegel, H.P.: On evaluation of outlier rankings and outlier scores. In: Proceedings of the 12th SIAM International Conference on Data Mining (SDM), Anaheim, CA. pp. 1047–1058 (2012)
38. Schubert, E., Zimek, A., Kriegel, H.P.: Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. Data Mining and Knowledge Discovery 28(1), 190–237 (2014)
39. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008. pp. 990–998 (2008), `http://doi.acm.org/10.1145/1401890.1402008`
40. Tong, H., Lin, C.: Non-negative residual matrix factorization with application to graph anomaly detection. In: Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, April 28-30, 2011, Mesa, Arizona, USA. pp. 143–153 (2011), `https://doi.org/10.1137/1.9781611972818.13`
41. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.J.: SCAN: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007. pp. 824–833 (2007), `http://doi.acm.org/10.1145/1281192.1281280`

42. Zimek, A., Campello, R.J.G.B., Sander, J.: Ensembles for unsupervised outlier detection: Challenges and research questions. ACM SIGKDD Explorations 15(1), 11–22 (2013)
43. Zimek, A., Campello, R.J.G.B., Sander, J.: Data perturbation for outlier detection ensembles. In: Proceedings of the 26th International Conference on Scientific and Statistical Database Management (SSDBM), Aalborg, Denmark. pp. 13:1–12 (2014)
44. Zimek, A., Gaudet, M., Campello, R.J.G.B., Sander, J.: Subsampling for efficient and effective unsupervised outlier detection ensembles. In: Proceedings of the 19th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Chicago, IL. pp. 428–436 (2013)
45. Zimek, A., Schubert, E., Kriegel, H.P.: A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining 5(5), 363–387 (2012)

Guilherme Oliveira Campos is currently a PhD student in Computer Science at Federal University of Minas Gerais (UFMG). He holds master-level degree in Computer Science at University of São Paulo (USP). His research interests include outlier detection, ensemble techniques for unsupervised learning and anomaly detection in graphs.

Edré Moreira is currently a PhD student in Computer Science at Federal University of Minas Gerais (UFMG). He holds master-level degree in Computer Science at Federal University of Minas Gerais. His research interests include unsupervised learning and anomaly detection in graphs.

Wagner Meira Jr. obtained his PhD in Computer Science from the University of Rochester in 1997 and is Full Professor at the Computer Science Department at Universidade Federal de Minas Gerais, Brazil. He has published more than 200 papers in top venues and is co-author of the book Data Mining and Analysis - Fundamental Concepts and Algorithms published by Cambridge University Press in 2014. His research focuses on scalability and efficiency of large scale parallel and distributed systems, from massively parallel to Internet-based platforms, and on data mining algorithms, their parallelization, and application to areas such as information retrieval, bioinformatics, and e-governance.

Arthur Zimek is Associate Professor in the Department of Mathematics and Computer Science (IMADA) at University of Southern Denmark (SDU), in Odense, Denmark and Head of the Section ”Data Science and Statistics”. He joined SDU in 2016 after previous positions at Ludwig-Maximilians-University Munich (LMU), Germany, Technical University Vienna, Austria, and University of Alberta, Edmonton, Canada. Arthur holds master-level degrees in bioinformatics, philosophy, and theology, involving studies at universities in Germany (TUM, HfPh, LMU Munich, and JGU Mainz) as well as Austria (LFU Innsbruck). His research interests include ensemble techniques for unsupervised learning, clustering, outlier detection, and high dimensional data, developing data mining methods as well as evaluation methodology. He published more than 80 papers at peer reviewed international conferences and in international journals. He co-organized several workshops and mini-symposia at SDM, KDD, and Shonan and served as workshop co-chair for SDM 2017.

Received: October 10, 2018; Accepted: May 1, 2019.