Community Feature Selection for Anomaly Detection in Attributed Graphs

Mario Alfonso Prado-Romero(✉) and Andrés Gago-Alonso

Advanced Technologies Application Center (CENATAV),
7a ♯ 21406, Rpto. Siboney, Playa, CP 12200 Havana, Cuba
{mprado,agago}@cenatav.co.cu

Abstract. Anomaly detection on attributed graphs can be used to detect telecommunication fraud, money laundering, intrusions in computer networks, atypical gene associations, or people with strange behavior in social networks. In many of these application domains, the number of attributes of each instance is high and the curse of dimensionality negatively affects the accuracy of anomaly detection algorithms. Many of these networks have a community structure, where the elements in each community are more related among them than with the elements outside. In this paper, an adaptive method to detect anomalies using the most relevant attributes for each community is proposed. Furthermore, a comparison among our proposal and other state-of-the-art algorithms is provided.

Keywords: Anomaly detection · Feature selection · Attributed graphs

1 Introduction

Many phenomena, from our world, like neural networks, bank transactions, social networks, or genes in our DNA can be modeled as networks of interconnected elements. In these networks, each element has a set of features, and it also has relationships with other elements. Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior [1]. Anomaly detection on the previously mentioned networks can be used to detect intrusions in computer networks, money laundering, identity thief in telecommunications, or strange gene associations, among other applications.

Traditional anomaly detection techniques only analyze the information about the elements [1], or the information regarding their relationship [2]. However, in many application domains like social networks, online shopping or bank transactions, both types of information can be found. Techniques for detecting anomalies in attributed graphs can deal with this heterogeneous data. Most of these techniques take advantage of the community structure present in the graphs, to analyze each element in a context relevant to it detecting subtle anomalies like products with a higher price than its co-purchased products.

C. Beltrán-Castañón et al. (Eds.): CIARP 2016, LNCS 10125, pp. 109–116, 2017.
DOI: 10.1007/978-3-319-52277-7_14

In many application domains the number of features describing an element can be very high, thus identify relevant patterns in the data become very difficult, this is known as the curse of dimensionality [3]. To overcome this fact, it is important that anomaly detection algorithms identify the most meaningful features to be used in the detection process. In this paper, a method for improving anomaly detection in attributed graphs, using an unsupervised feature selection algorithm to select the most relevant features for each community of elements, is proposed. The advantages of this method are shown on the Amazon co-purchase network of Disney products[1] [4].

In the next sections, the existing approaches to anomaly detection are analyzed (Sect. 2), some basic concepts are introduced (Sect. 3), our method is presented (Sect. 4) and experimental results on a real data set are analyzed (Sect. 5). Conclusions are presented in Sect. 6.

2 Related Work

The three major approaches for anomaly detection discussed in this section are the vector based approach, graph based one, and hybrid one. Furthermore, the advantages and disadvantages of each approach, and how it tackles with the curse of dimensionality, are discussed.

The commonly reported anomaly detection techniques are designed to deal with vector valued data [1]. Some of these are distance-based algorithms [5], density-based algorithms [6,7] and algorithms to find clustered anomalies [8–10], but none of them avoid the curse of dimensionality. Some recent techniques rank objects based on the selection of a relevant subset of its attributes to tackle the curse of dimensionality [11,12]. Nonetheless, none of them takes into consideration relationships among the elements, ignoring part of the information in the data set, thus these algorithms cannot identify the most significant features for each community of elements.

Relationships among elements give valuable information about the structure of a network, due to this, many algorithms to detect anomalous nodes in graphs, using their relationships, has been proposed [2]. Commonly used techniques include the analysis of the structural characteristics of network elements finding deviations from normal behavior [13], and searching for infrequent structures in the network [14,15]. None of these algorithms uses the attributes of the elements, thus they are not affected by the curse of dimensionality, but they present low accuracy in heterogeneous data sets because they ignore part of the existing information.

Anomaly detection in attributed graphs where both graph and vector data are analyzed has not received much attention. The algorithm described in [16] combines community detection and anomaly detection in a single process, finding elements deviated from its community behavior. This algorithm uses the full attribute space; thus, its results are affected by the curse of dimensionality.

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

The technique described in [4] proposes an outlier ranking function capable of use only a subset from the node attributes. In a first step, this technique uses a state-of-the-art graph clustering algorithm considering subsets of the node attributes [17–19] and in a second step detects elements whose behavior deviates from the one of its group. This technique does not take into consideration that, in some application domains, the relationships among elements give a context to analyze them, instead of directly indicate than an element is anomalous. Thus, the performance of this algorithm and the quality of its detection, in this application domains, is affected.

3 Basic Concepts

It is important to introduce some fundamental concepts before presenting our proposal.

Definition 1 (Attributed Graph). *An attributed graph* $G =< V, E, a >$ *is a tuple where:*

(i) *The set* $V = \{v_1, v_2, \ldots, v_n\}$ *contains the vertices of the graph.*
(ii) *The set* $E = \{(v, u)|v, u \in V\}$ *contains the edges of the graph.*
(iii) *The function* $a : V \rightarrow \mathbb{R}^d$ *assigns an attribute vector of size d to each vertex from G.*

The vertices of the graph are the elements of the network, and the edges the relationships among them.

Definition 2 (Disjoint Clustering). *A disjoint clustering* $C = \{C_1, C_2, \ldots, C_k\}$ *of elements from G is a set where:*

(i) $\forall C_i \in C$, $C_i \subset V$
(ii) *For each* $C_i, C_j \in C, i \neq j, C_i \cap C_j = \emptyset$
(iii) $\cup_{i=1}^{k} C_i = V$.

In this work, the disjoint clusterings are referred just as clusterings.

Definition 3 (Outlier Ranking). *An outlier ranking from a graph G is a set* $R = \{(v, r)|v \in V, r \in [0, 1]\}$ *of tuples, each one containing a vertex from G and its outlierness score.*

4 Improving Anomaly Detection Using Community Features

Many real networks are structured in communities. A community is a group of elements more connected among them, than with external elements. Usually, the elements of a same community share similar features. We propose to perform feature selection per community, selecting those features that better represent an element in its context. The elements outside the community of an element

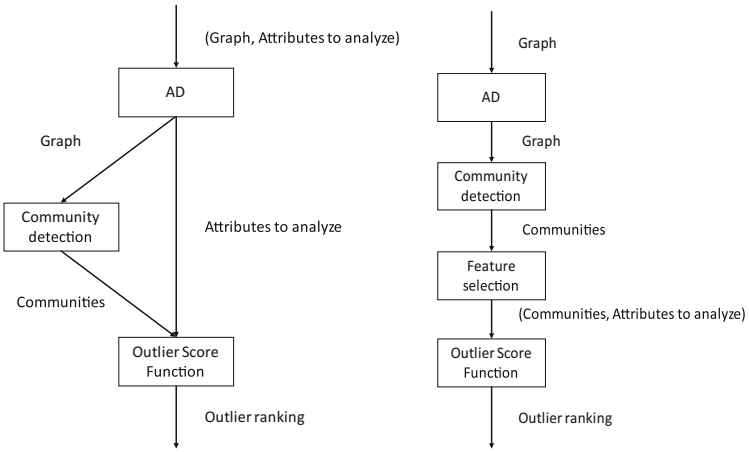

Fig. 1. Components interaction in the base algorithm (left) and in the improved algorithm (right)

are not relevant for it, and could affect the quality of the feature selection, for this reason they are ignored.

This idea was applied to our Glance algorithm, designed for anomaly detection in attributed graphs where the edges behave as contextual attributes. The algorithm originally received as parameters the attributed graph and the attributes to be considered by the anomaly score. In a first stage, the algorithm uses Louvain community detection method [20] to find communities of elements in the graph using connections among elements. In a second stage, it iterates over each community and uses an outlierness score function to determine the outlierness degree of each element. This function receives, as a parameter, the features to be used for the score.

The changes made to the original algorithm include removing the need of external parameters set by the user. Also, in the second stage, the algorithm selects the most relevant features for each community using a Laplacian Score [21] and then applies an outlierness score to each element of the community. The Laplacian Score ranks as more representatives those features, with large variance, that are similar in near elements. Using this feature selection technique, the anomaly detection algorithm becomes completely unsupervised. In Fig. 1, a comparison among the components interaction in the base algorithm and in the improved one can be observed.

The Glance algorithm with community feature selection can be observed in more detail in Algorithm 1. The algorithm receives an attributed graph G and returns an outlierness score of the vertices from G. In the second line, The Louvain community detection algorithm is used to find relevant groups of elements in G. In lines 3 to 10 the algorithm iterates over each community of G. In line 4 the more descriptive features for the community are selected. In lines 6 to 9, the outlierness score for each element in C_i is calculated using Glance score

function. This function defines the outlierness score of an element as the percent of elements in its same community that have a difference with it greater than the mean difference among the community members. Finally a ranking R of the vertices from G is returned in line 11.

Algorithm 1. Glance Algorithm with Community Feature Selection

Input: G // *Attributed Graph*
Output: R // *An anomaly ranking of the vertices from G*

1 $R \leftarrow \varnothing$
2 $C \leftarrow$ Clustering(G)
3 **foreach** $C_i \in C$ **do**
4 $\quad A \leftarrow$ FeatureSelection(C_i)
5 $\quad P_{C_i} \leftarrow$ mean values of attributes from A in C_i
6 $\quad$ **foreach** $v_j \in C_i$ **do**
7 $\quad\quad R_{v_j} \leftarrow$ a dictionary containing for each attribute a_l from v_j the number of elements u satisfying $|a_l(v_j) - a_l(u)| > a_l(P_{C_i})$
8 $\quad\quad R \leftarrow R \cup \{(v_j, max(a_l \in R_{v_j})\}$
9 $\quad$ **end**
10 **end**
11 **return** R

It is important to notice that community feature selection splits the number of elements to analyze in groups. This fact is useful with algorithms $O(V^2)$ or higher because it reduces the number of operations required to process the data. Thus the performance of the algorithm is increased in real networks where the number of communities is usually high. Also, this property can be useful to process the data in parallel. In the next section, the performance of the algorithm on a real network is analyzed.

5 Experimental Results

In this section, a comparison between Glance with community feature selection and the base algorithm is performed. Furthermore, our proposal is compared with other state-of-the-art algorithms.

The comparison was performed on the amazon co-purchase network of Disney products (124 nodes with 334 edges). This database was used as benchmark in [4], and the authors provided a labeled outlier ground truth. The ground truth was built from a user experiment where outliers were labeled by high school students. In the experiment, the products where clustered using a modularity technique, and then the students were asked to find outliers in each cluster. The experiment used the edges only to give context to the elements, and the attributes of the vertices were used to identify anomalies in each cluster. Thus, this database and its labeled anomaly ground truth represents a problem of anomaly detection in attributed graph where the edges are contextual attributes.

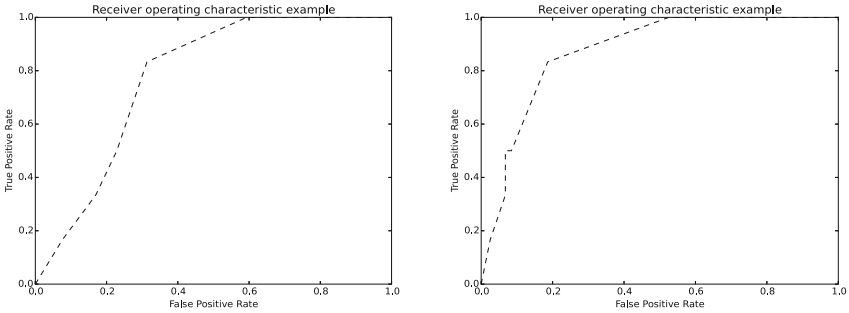

Fig. 2. ROC of the base Glance algorithm (left) and the one using community feature selection (right)

In Fig. 2, a comparison between the base Glance algorithm and the one improved with community feature selection is displayed. The former has an AUC (Area Under the ROC Curve) value of 77.4% and a runtime of 150 ms, assuming that the user selected all features as relevant. Thus the results were affected by the curse of dimensionality. The later has an AUC value of 87.43% with a runtime of 93 ms, and can be observed that the ROC curve rises faster than in the base algorithm. Using community feature selection, an improve of more than 10% was achieved and also the performance of the algorithm was improved due to a reduction in the dimensions of the data to be processed.

Table 1. AUC results for all algorithms on the Amazon database (Disney DVD selection).

Used data	Paradigm	Algorithm	AUC[%]	Runtime[ms]
Attribute data only	Full space outlier analysis	LOF [6]	56.85	41
	Subspace outlier analysis	SOF [11]	65.88	825
		RPLOF [12]	62.50	7
Graph structure only	Graph clustering	SCAN [22]	52.68	4
Attributes and graph data	Full space outlier analysis	CODA [16]	50.56	2596
	Subspace outlier analysis	GOutRank [4]	86.86	26648
	Contextual edges	Glance	77.40	150
		Glance + CFS	87.43	93

In Table 1, it is displayed the AUC measure and the runtime for different approaches to anomaly detection. In the approaches considering just the attribute vectors, only those performing subspace analysis can overcome the

curse of dimensionality. Thus, these techniques have better accuracy compared with the ones performing full subspace analysis. Nonetheless, none of them can detect the complex outliers present in the data, because they ignore the relationship among elements. The approach considering only the graph structure has poor results in this database. This is mainly because, in this problem, edges are contextual attributes and they do not directly determine the outlierness of an element. The only approach able to detect the complex outliers in this database is the one that considers both attribute data and graph structure. Although the CODA algorithm belongs to this approach, it is greatly affected by the curse of dimensionality. The GOutRank algorithm obtains good results in this database, but ignores the contextual nature of the edges, affecting the quality of its results. Also, it is the most time consuming algorithm used in this comparison. The Glance algorithm using all the features achieves better detection than other algorithms but is affected by the curse of dimensionality. The Glance algorithm with community feature selection has the best AUC value and also has better runtimes than the other algorithms for anomaly detection in attributed graph. This results are an example of the potential of community feature selection to improve anomaly detection in attributed graphs.

6 Conclusions

In this paper, an adaptive method to improve anomaly detection in attributed graphs using community feature selection was proposed. The method was used to improve an algorithm to detect anomalies in attributed graphs where the edges behave as contextual attributes, and an improve in the result of more than a 10% was achieved. Also, the improved algorithm was compared with others from the state of the art and it achieved better results.

There are many open challenges in the field of anomaly detection in attributed graphs, we will focus on some of them in future work. The first one is to integrate feature selection and anomaly detection in a single process for avoiding redundant calculations and improve the performance of the algorithms. Finally, the algorithm could be parallelized to improve its performance, because it selects features and detects anomalies in disjoint communities.

References

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. (CSUR) **41**, 15 (2009)
2. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data Min. Knowl. Disc. **29**, 626–688 (2015)
3. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful? In: Beeri, C., Buneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 217–235. Springer, Heidelberg (1999). doi:10.1007/3-540-49257-7_15
4. Müller, E., Sánchez, P.I., Mülle, Y., Böhm, K.: Ranking outlier nodes in subspaces of attributed graphs. In: 2013 IEEE 29th International Conference on Data Engineering Data Engineering Workshops (ICDEW), pp. 216–222 (2013)

5. Knorr, E.M.: Outliers and Data Mining: Finding Exceptions in Data. The University of British Columbia, Vancouver (2002)
6. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. ACM Sig. Rec. **29**(2), 93–104 (2000)
7. Papadimitriou, S., Kitagawa, H., Gibbons, P.B., Faloutsos, C.: LOCI: fast outlier detection using the local correlation integral. In: ICDE, pp. 315–326 (2003)
8. Xiong, Y., Zhu, Y., Yu, P.S., Pei, J.: Towards cohesive anomaly mining. In: AAAI (2013)
9. Liu, F.T., Ting, K.M., Zhou, Z.-H.: Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data (TKDD) **6**(1), 3 (2012)
10. Liu, F.T., Ting, K.M., Zhou, Z.-H.: On detecting clustered anomalies using SCiForest. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6322, pp. 274–290. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15883-4_18
11. Aggarwal, C.C., Yu, P.S.: Outlier detection for high dimensional data. ACM Sigmod Rec. **30**(2), 37–46 (2001)
12. Lazarevic, A., Kumar, V.: Feature bagging for outlier detection. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 157–166 (2005)
13. Akoglu, L., McGlohon, M., Faloutsos, C.: OddBall: spotting anomalies in weighted graphs. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS (LNAI), vol. 6119, pp. 410–421. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13672-6_40
14. Eberle, W., Holder, L.: Discovering structural anomalies in graph-based data. In: Data Mining Workshops 2007. ICDM Workshops 2007, pp. 393–398 (2007)
15. 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, pp. 631–636 (2003)
16. 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, pp. 813–822 (2010)
17. Moser, F., Colak, R., Rafiey, A., Ester, M.: Mining cohesive patterns from graphs with feature vectors. In: SDM, vol. 9, pp. 593–604 (2009)
18. Günnemann, S., Farber, I., Boden, B., Seidl, T.: Subspace clustering meets dense subgraph mining: a synthesis of two paradigms. In: Data Mining (ICDM) 10th International Conference on Data Mining, pp. 845–850 (2010)
19. Akoglu, L., Tong, H., Meeder, B., Faloutsos, C.: PICS: parameter-free identification of cohesive subgroups in large attributed graphs. In: SDM, pp. 439–450 (2012)
20. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. **2008**(10), P10008 (2008)
21. He, X., Cai, D., Niyogi, P.: Laplacian Score for feature selection. In: Advances in Neural Information Processing Systems, pp. 507–514 (2005)
22. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.: Scan: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 824–833 (2007)