DENGRAPH: A Density-based Community Detection Algorithm

Tanja Falkowski, Anja Barth, Myra Spiliopoulou
Faculty of Computer Science, Otto-von-Guericke University Magdeburg, Germany
falkowski@iti.cs.uni-magdeburg.de

Abstract

Detecting densely connected subgroups in graphs such as communities in social networks is of interest in many research fields. Several methods have been developed to find communities but most of them have a high time complexity and are thus not applicable for large networks. Inspired by the clustering algorithm incremental DBSCAN we propose a density-based graph clustering algorithm DENGRAPH that is designed to deal with large dynamic datasets with noise and present first experimental results.

1. Introduction

The interaction in online communities has become for many people part of their everyday live. Since platforms were people exchange experiences about products or share information about objects such as research papers or bookmarks, e.g., in form of tags (annotations) are meanwhile a widely prevalent phenomenon in the WWW, methods have been developed to learn more about the structure and the temporal development of these interactions (see, e.g. [4]).

A widely used algorithm to detect community structures in graphs is the edge betweenness clustering (EBC) [5]. However, when applying the hierarchical technique to large social networks it has some serious drawbacks: (i) Due to its high time complexity it is unfeasible to use the algorithm for large networks with several thousand nodes, (ii) a measure is needed to determine the cut-off level in the dendrogram and (iii) the temporal development can only be observed by aggregating data in time windows; this makes it difficult to compare detected subgroups in different time windows over time. Furthermore, since social networks usually have a high number of actors that do not contribute (*noise objects*), we need for the analysis of large dynamic networks a fast algorithm that efficiently deals with noise objects.

In this paper we propose to transfer density-based clustering to graph structures. In Sect. 2 we briefly discuss related work. In Sect. 3 we present DENGRAPH and show experimental results and a comparison with EBC in Sect. 4.

2. Related Work

Social networks have been analyzed widely and several statistical properties have been determined. Besides a short average distance between two nodes (small world effect) and a right-skewed degree distribution, it appears to be common that the network shows a community structure: The networks consist of subsets of vertices that are more closely connected with each other than with the rest of the network.

A widely used method to detect communities in networks is the edge betweenness clustering (EBC) [5]. The hierarchical divisive algorithm removes iteratively edges with the highest edge betweenness. The output is represented by a dendrogram. Each level of the dendrogram represents a clustering where each cluster is a subset of vertices. In [6], a measure was proposed to select the clustering with the highest modularity. As mentioned above, the main disadvantages of the EBC algorithm are the high time complexity ($|E|^2|V|$), the problem of local maxima in the modularity curve and the comparison of clusterings over time.

However, when analyzing interaction networks we often have to deal with networks that are characterized by a very high number of actors (nodes), a large number of rather irrelevant actors (*noise*) and temporal dynamics in the interaction behavior. Besides *partitioning algorithms* that partition a dataset into k clusters and *hierarchical algorithms* that create a hierarchical decomposition of the objects, *density-based algorithms* have been developed such as the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [3]. The main idea is that for each point in a cluster the neighborhood of a given radius (ϵ) has to contain at least a minimum number of points (η). The two parameters ensure that the density of the points in the cluster exceeds some threshold. DBSCAN has been extended to an incremental algorithm, resulting in a more efficient method for large dynamic noisy data sets for spatial data [2].

3. The Algorithm DENGRAPH

To adapt the idea of density-based incremental clustering of spatial data to large dynamics graph structures we need

0-7695-3026-5/07 $25.00 © 2007 IEEE
DOI 10.1109/WI.2007.74

to define a distance function between pairs of vertices in graphs and update functions for the incremental changes.

3.1. Distance Function

To group actors according to their closeness in graph structures, we need to define a function that determines the distance between two actors. In social networks we can assume that the closeness of actors is reflected by the number of interactions between them.

Definition 1 *The distance between two actors p and q is defined as*

$$dist(p,q) = \begin{cases} 0 & x = y \\ min(I_{p,q}, I_{q,p})^{-1} & (I_{p,q} > 1) \\ & \wedge(I_{q,p} > 1) \\ 1 & otherwise, \end{cases} \tag{1}$$

where $I_{p,q}$, $I_{q,p}$ is the number of interactions between actors p and q initiated by p and q, respectively.

The distance between two actors is normalized by the number of reciprocal interactions, thus $dist(p,q)$ ranges from 0 to 1. Two actors with a high number of reciprocal interactions are closer and the distance for two actors that do not interact is 1. Positivity and symmetry of the distance function are fulfilled; the triangle inequality is usually not satisfied. However, this property is not necessary as we calculate only the distance between interacting actors.

Based on the distance function we define the ϵ-*neighborhood* of a node p as follows:

Definition 2 *The ϵ-neighborhood of a node p is defined by $N_\epsilon(p) = \{q \in V \; |dist(p,q) \leq \epsilon\}$.*

On the basis of [2], we define the concepts that are necessary for the incremental graph clustering.

Definition 3 *A vertex p is directly density-reachable from a vertex q wrt. ϵ and η in the set of vertices V if $p \in N_\epsilon(q)$, where $N_\epsilon(q)$ is a subset of V contained in the ϵ-neighborhood of q and $|N_\epsilon(q)| \geq \eta$. (q is a core vertex)*

Definition 4 *A vertex p is density-reachable from a vertex q wrt. ϵ and η in the set of vertices V, denoted as $p >_V q$, if there is a chain of vertices $p_1, \ldots, p_n$ with $p_1 = q$ and $p_n = p$ such that $p_i \in V$ and p_{i+1} is directly density-reachable from p_i wrt. ϵ and η.*

Two vertices that are at the border of a dense subgroup are not necessarily density-reachable because there might not be enough vertices in their ϵ-neighborhoods. However, since both vertices belong to one subgroup, there must be a third vertex in the subgroup from which both vertices are density-reachable.

Definition 5 *A vertex p is density-connected to a vertex q wrt. ϵ and η in the set of vertices V, if there is a vertex $o \in V$ such that both vertices p and q are density-reachable from o wrt. ϵ and η.*

A *dense subgroup* is defined as a maximal set of vertices that are density-connected wrt. ϵ and η. *Noise* is the set of vertices that do not belong to any dense subgroup.

Definition 6 *Let V be a set of vertices. A dense subgroup DS wrt. ϵ and η is defined as a non-empty subset of V that satisfies the following conditions: Maximality: $\forall p, q \in V$: if $p \in DS$ and $q >_V p$ wrt ϵ and η, then also $q \in DS$. Connectivity: $\forall p, q \in DS$: p is density-connected to q wrt. ϵ and η in V.*

The DENGRAPH clustering procedure to detect dense subgraphs works as follows:

- Select an object p from V and determine all vertices that are density-reachable from p wrt. ϵ and η.

- If p is a core vertex, a new dense subgroup is yielded. A new *id* is created and all vertices in the neighborhood are marked with the current *id*. All vertices from N_ϵ are put onto a stack and the ϵ-neighborhood of these vertices is checked. If the number of vertices in the neighborhood exceeds η, all vertices that are not yet classified are pushed onto the stack and marked with the current id. This procedure is repeated until the stack is empty.

- If $|N_\epsilon(p)| < \eta$, p is no core vertex and marked as noise.

3.2 Incremental Updates

Since DBSCAN is designed for data in a metric space and due to its density-based nature, the insertion or deletion of an object affects the current clustering only in the neighborhood of this object. In graph structures, besides the insertion and deletion of vertices, distances between objects can change. If two actors *p* and *q* communicate in period *t+1* more or less often than in period *t*, $dist(p,q)$ will decrease or increase respectively. Thus, the incremental approach for DENGRAPH distinguishes between the following update scenarios:

New Actor. Actor is added in $t+1$ to graph structure. Node is classified as *noise*, since it has only one neighbor and can thus not be a core vertex. In case no further actions on this node take place (such as a further observation due to the appearance of a relevant relationship) no further updates are necessary.

Changing Relationship. 1. *New Relationship.* The new relationship between actor *p* and *q* results in a new edge and

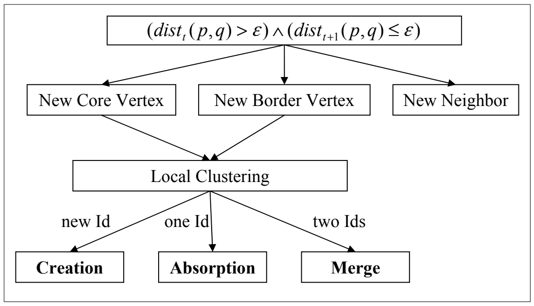

Figure 1. New close relation.

the distance $dist(p, q)$ between the respective nodes is calculated. 2. *Changing Relationship Strength.* If the strength of the relationship changes, $dist(p, q)$ is updated.

In both cases, $dist(p, q)$ must be updated. The subgroup structure of the graph does only change if (i) a new relevant edge appears, i.e. $dist_t(p, q) > \epsilon$ and $dist_{t+1}(p, q) \leq \epsilon$ (cf. Fig. **??**) or if (ii) a former relevant edge disappears, i.e. $dist_{t+1}(p, q) > \epsilon$ and $dist_t(p, q) \leq \epsilon$ (cf. Fig. **??**)

3.2.1 New close relation

If $dist_{t+1}(p, q) \leq \epsilon$ and $dist_t(p, q) > \epsilon$, p becomes in $t+1$ a member of q's ϵ-neighborhood and vice versa. In this case, the following scenarios are possible (cf. Fig. **??**):

New core vertex. If p and q have no subgroup-*id*, one or both vertices might become a core vertex. Using one prospective new core vertex as input, the clustering routine is started. If the stack contains vertices that did not belong to a subgroup before, a new subgroup is established. (If both vertices become cores, the second will be handled when dealing with the neighborhood of the first core.)

New subgroup member. A noise vertex p becomes a new member of a subgroup if it becomes (directly) density-reachable from a core vertex q. p and possibly some other former noise vertices become members of the subgroup.

New neighbor. If p and q do not belong to a subgroup and do not become a core or border vertex, they are marked as *noise*. If both vertices are already core vertices in t, no changes in the subgroup structure are invoked.

If the detected close relationship results in a new core vertex or new subgroup member, the respective core p is used as the input for the clustering algorithm. Note that only core vertices are used for the calculation as the algorithm considers all relevant neighbors of the cores anyway. The procedure results in one of the following actions:

Creation. A new subgroup is created if the stack only contains core objects with no subgroup-*id*. A new cluster is created with p as the core vertex.

Absorption. If the neighbors of p belong to the same subgroup *DS*, p and possible other noise vertices are added to the subgroup *DS*.

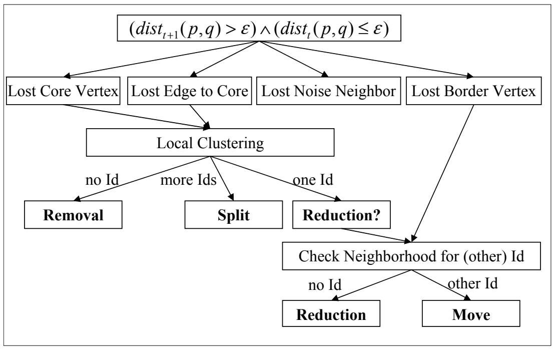

Figure 2. Lost close relation.

Merge. If the stack includes core vertices that belong to several dense subgroups before the new close relationship was established, all these subgroups are merged.

3.2.2 Lost close relation

If $dist_{t+1}(p, q) > \epsilon$ and $dist_t(p, q) \leq \epsilon$, p and q are no longer members of each other's ϵ-neighborhood. The following cases must be distinguished (cf. Fig. **??**):

Lost core vertex. If p looses its core vertex characteristic, a procedure recalculates the subgroup and checks whether the subgroup disappears, is reduced or split.

Lost edge to core vertex. The border vertex looses its directly-reachability to the subgroup. It has to be checked whether the vertex is still a core vertex and the subgroup still exists. Otherwise the subgroup is reduced, split or removed.

Lost edge between cores. If two core members lose their directly-reachability, the subgroup must be recalculated to check whether the subgroup is reduced, split or removed.

Lost neighbor. The edge between two noise vertices p and q is removed if the distance between them drops below ϵ. If two border vertices p and q are no longer directly-reachable it must be checked whether one or both still belong to a subgroup.

The lost ϵ-neighbor of p might result in the following changes in the subgroup DS that it belonged to:

Removal. If after the deletion of a directly-density reachable vertex p in a subgroup DS, no core objects are in the neighborhood $N_\epsilon(p)$, the subgroup DS is deleted. All former subgroup members become noise or members of other subgroups.

Reduction. The subgroup DS survives, but p is removed. Some vertices in $N_\epsilon(p)$ may become noise.

Move. p is removed from DS and becomes a member of another subgroup.

Split. The vertices in the stack belonged to exactly one cluster before p was removed. It has to be checked whether the vertices are density-connected by other vertices in the former subgroup DS or if a new subgroup-*id* is created. In the latter case, the subgroup has been split.

The incremental DENGRAPH clustering algorithm yields basically the same results as the non-incremental version of the algorithm. Some slight discrepancies could be observed because the clustering results depend to some extend on the order of the data input. Due to space restriction we cannot present the results here and refer to [1].

4. Experiments

We applied the edge betweenness clustering algorithm and DENGRAPH to a social network graph and compared both results to obtain information about the characteristics of both clustering methods.

For the experiments we used the ENRON email data set (http://www.isi.edu/∼adibi/Enron/Enron.htm). Since the number of messages differs considerably at the beginning and end we selected the interval October 1999 to March 2002. In this time period 248.353 distinct messages to 2.046.843 recipients were sent. The weight of the edges (distance) between two actors is calculated according to Def. **??**. The data set shows a right-skewed degree distribution and a short average path lengths (small-world effect). Due to its size and social network characteristics, it is particularly suitable for dense subgraph detection.

We determine the parameters ϵ and η for the experiments by analyzing the email data over three months in 2001 to see how the size of each vertice's neighborhood and the percentage of noise depend on a given η. Furthermore, for different ϵ values we analyzed the number of messages that are on average exchanged between vertices in each interval. For our experiments we chose $\eta = 3$ and $\epsilon = 1/3$. Due to space restriction we refer to [1].

We performed a clustering on 130 intervals (one interval comprises one week) with the weighted edge betweenness clustering and with DENGRAPH. Since a comparison to a 'real clustering' is not possible for this data set, we quantitatively compare both clustering results and discuss basic characteristics. The EBC shows about 12,000 subgroups; 81 percent are singletons, 40 percent of the remaining actors are grouped in subgroups with more than 50 members. On average, 18 groups are built per interval with on average 10 members. On average one group per interval has more than 100 members and the average group size is very low due to the large amount of singletons. In summary, one could say that even though the EBC reveals a high number of subgroups, most contain only one member. The other 20 percent tend to be members of rather large groups. For $\epsilon = 1/3$ and $\eta = 3$, DENGRAPH classifies on average 93 percent of all vertices as noise. If we compare the subgroups of DENGRAPH with the EBC subgroups (with more than one member) we obtain the following: DENGRAPH subgroups are in 39 percent of all cases subsets of larger EBC subgroups. On average, 3 DENGRAPH subgroups form one EBC subgroup. DENGRAPH thus tends to find smaller subgroups that are more closely connected. Since most subgroups detected with DENGRAPH have an (large) overlap with the EBC groups - most of them are subsets of EBC groups - we conclude that both methods in general reveal similar community structures, however with a different granularity. DENGRAPH reveals smaller more dense groups whereas EBC tends to merge also less close groups. Even though a more thorough investigation regarding the time complexity needs to be performed, we can already state that the density-based approach outperforms the hierarchical methods by an order of magnitude.

5. Conclusion

We presented an incremental density-based algorithm to detect dense subgroups in graphs. We adapted the incremental DBSCAN algorithm to graph structuresby defining a distance function for two interacting vertices and update functions for the incremental clustering. First experiments with the ENRON data set and comparisons with the edge betweenness clustering show, that the algorithm reveals meaningful subgroups. DENGRAPH is capable to efficiently remove small groups as noise and detects smaller more dense structures compared to the EBC.

References

[1] A. Barth, Entwicklung eines dichtebasierten Clusterverfahrens für Graphen zur Erkennung von Community-Strukturen, Masterthesis [in German], Otto-von-Guericke-University Magdeburg, 2007.

[2] M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu, Incremental Clustering for Mining in a Data Warehouse Environment, In: Proc. of 24th VLDB Conference, 1998.

[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, In: Proc. of KDD, 226-231, 1996.

[4] T. Falkowski, J. Bartelheimer, and M. Spiliopoulou, Community Dynamics Mining, In: Proc. of 14th Europ. Conf. on Information Systems, 2006.

[5] M. Girvan, and M.E.J. Newman, Community structure in social and biological networks, PNAS, 99(12), 7821-7826, 2002.

[6] M.E.J. Newman, and M. Girvan, Finding and evaluating community structure in networks, Physical Review, E 69(026113), 2004.