



International Journal of Intellectual Advancements and Research in Engineering Computations

Mining graphs for understanding time -varying volumetric data

Mrs.C.Navamani MCA.,M.PHIL.,ME¹., Mrs.G.Visithra²

¹Professor, Department of MCA, Nandha Engineering College (Autonomous), Erode-52.

²Student, Department of MCA, Nandha Engineering College (Autonomous), Erode-52.

ABSTRACT

In this paper specifically focus on time-varying volume data visualization and investigate transition relationships among data items over time. We present a mining approach that automatically extracts features from a graph-based representation for understanding time varying data. Beyond straightforward graph properties, users are given further guidance available through a series of graph analysis techniques including graph simplification, community detection, and visual recommendation. Graph simplification condenses a large graph to a smaller one by abstracting known structures, such as fan, connector, and clique, presenting a less cluttered view for quick comprehension of the overall graph structure. Community detection organizes nodes with close relationships into groups, allowing visual comparison between groups of nodes instead of individual nodes. Visual recommendation automatically highlights individual nodes or node groups based on user selected items, enabling users to spend more time on the actual analysis instead of painstaking interaction. In this paper plan to extend proposed work to handle multivariate data sets. We can either fuse multiple variables into one type of node, or construct one type of node for each individual variable and visualize the relations between variables using compound graphs. We will also investigate time-evolving graphs derived from scientific data sets for identifying temporal hotspots, detecting anomaly, and aligning multiple graphs for finding common features and distinct patterns. In this paper addition types of data sources are used for the recommendations, essentially these data sources can be modeled in the form of various types of graphs. This paper aims at providing a general framework on mining Web graphs for recommendations, 1) A novel diffusion method is proposed which propagates similarities between different nodes and generates recommendations 2) then it is illustrated how to generalize different recommendation problems into the graph diffusion framework.

Keywords: Data Mining, Mining Volumetric Data, Mining Web Graph, Fan Detection, Clique Detection.

INTRODUCTION

The idea of transforming the data and their relationships into an abstract view for exploring complex relationships and improving data understanding has been accepted as a viable means to analyze and visualize scientific data. Most of these representations lack enough guidance for user exploration and navigation. In many cases, users can only rely on low-level visual hints to figure out the relationships with the underlying

data through brushing and linking. This approach may work for small data with simple relationships, but becomes increasingly inefficient for larger data with greater complexity. Therefore, solutions that help users sift through the data and their relationships for cost-effective understanding are highly desirable.

We present a mining approach that automatically extracts features from a graph-based representation for understanding time varying data. Graph simplification condenses a large graph to a smaller

Author for correspondence:

Department of MCA, Nandha Engineering College (Autonomous), Erode-52.

one by abstracting known structures, such as fan, connector, and clique, presenting a less cluttered view for quick comprehension of the overall graph structure. Community detection organizes nodes with close relationships into groups, allowing visual comparison between groups of nodes instead of individual nodes. When the data set is large and the relationships are complex, navigating and exploring the resulting graph is a daunting task.

In this paper specifically focus on time-varying volume data visualization and investigate transition relationships among data items over time. We present a mining approach that automatically extracts features from a graph-based representation for understanding time varying data. Beyond straightforward graph properties, users are given further guidance available through a series of graph analysis techniques including graph simplification, community detection, and visual recommendation.

In this paper plan to extend proposed work to handle multivariate data sets. We can either fuse multiple variables into one type of node, or construct one type of node for each individual variable and visualize the relations between variables using compound graphs. We will also investigate time-evolving graphs derived from scientific data sets for identifying temporal hotspots, detecting anomaly, and aligning multiple graphs for finding common features and distinct patterns.

- It can be utilized in many recommendation tasks on the query suggestions, tag recommendations, image recommendations, etc.
- Abbreviation based Query suggestion is also considered.
- In addition, search engine results comparison is also considered.
- Personalized recommendations are given importance.
- Previous search query words are also taken into query suggestion calculation.

RELATED WORKS

Yi Gu and Chaoli Wang describe significant advances have been made in time-varying data analysis and visualization, mainly in improving

their ability to identify temporal trends and classify the underlying data. However, the ability to perform cost-effective data querying and indexing is often not incorporated, which poses a serious limitation as the size of time varying data continue to grow. They present a new approach that unifies data compacting, indexing and classification into a single framework.

They achieve this by transforming the time activity curve representation of a time-varying data set into a hierarchical symbolic representation. They further build an Indexable version of the data hierarchy, from which they create the iTree for visual representation of the time-varying data. A hyperbolic layout algorithm is employed to draw the iTree with a large number of nodes and provide focus context visualization for interaction. They achieve effective querying, searching and tracking of time-varying data sets by enabling multiple coordinated views consisting of the iTree, the symbolic view and the spatial view

Jeffrey Heer, Maneesh Agrawala, Wesley Willett selection is a fundamental task in interactive applications, typically performed by clicking or lassoing items of interest. However, users may require more nuanced forms of selection. Selecting regions or attributes may be more important than selecting individual items. Selections may be over dynamic items and selections might be more easily created by relaxing simpler selections (e.g., —select all items like this one). Creating such selections requires that interfaces model the declarative structure of the selection, not just individually selected items. The direct manipulation techniques that couple declarative selection queries with a query relaxation engine that enables users to interactively generalize their selections. Then, they apply their selection techniques in both information visualization and graphics editing applications, enabling generalized selection over both static and dynamic interface objects. A controlled study finds that users create more accurate selection queries when using their generalization techniques.

Jun Ma, Chaoli Wang and Ching-Kuang Shene describe a visual exploration of large and complex 3D flow fields is critically important for understanding many aero- and hydro-dynamical

systems that dominate various physical and natural phenomena in the world. They introduce the Flow Graph, a novel compound graph representation that organizes streamline clusters and spatial regions hierarchically for occlusion-free and controllable visual exploration. Their approach works with any seeding strategies as long as the domain is well covered and important flow features are captured. By transforming a flow field to a graph representation, they enable observation and exploration of the relationships among streamline clusters, spatial regions and their interconnection in the transformed space

Antonio Robles-Kelly and Edwin R. Hancock This paper shows how the Eigen structure of the adjacency matrix can be used for the purposes of robust graph-matching. They commence from the observation that the leading eigenvector of a transition probability matrix is the steady state of the associated Markov chain. When the transition matrix is the normalized adjacency matrix of a graph, then the leading eigenvector gives the sequence of nodes of the steady state random walk on the graph. They use this property to convert the nodes in a graph into a string where the node-order is given by the sequence of nodes visited in the random walk. They match graphs represented in this way, by finding the sequence of string edit operations which minimize edit distance. The aim in this paper is to investigate whether the Eigen structure of the adjacency matrix can be used to match graphs using a search method rather than by iteration. To do this they draw on the theory of Markov chains. They consider a Markov chain whose transition probability matrix is the normalized edge-weight matrix for a graph. The steady-state random walk for the Markov chain on the graph is given by the leading eigenvector of the transition probability, i.e. edge weight, matrix. String Edit Distance, Random Walks and Graph Matching Hence, by considering the order of the nodes defined by the leading eigenvector, were able to convert the graph into a string. This opens up the possibility of performing graph matching by using string alignment to minimize the Levenshtein or edit distance.

Frank van Ham, Martin Wattenberg presented a new technique, the phrase net, for generating visual overviews of unstructured text. A phrase net

displays a graph whose nodes are words and whose edges indicate that two words are linked by a user-specified relation. These relations may be defined either at the syntactic or lexical level. Different relations often produce very different perspectives on the same text. Taken together, these perspectives often provide an illuminating visual overview of the key concepts and relations in a document or set of documents.

Chaoli Wang describes a graphs represent general node-link diagrams and have long been utilized in scientific visualization for data organization and management. However, using graphs as a visual representation and interface for navigating and exploring scientific data sets has a much shorter history yet the amount of work along this direction is clearly on the rise in recent years. They take a holistic perspective and survey graph-based representations and techniques for scientific visualization. Specifically, they classify these representations and techniques into four categories, namely, partition-wise, relationship-wise, structure-wise, and provenance-wise. They survey related publications in each category, explaining the roles of graphs in related work and highlighting their similarities and differences. They also point out research trends and remaining challenges in graph-based representations and techniques for scientific visualization.

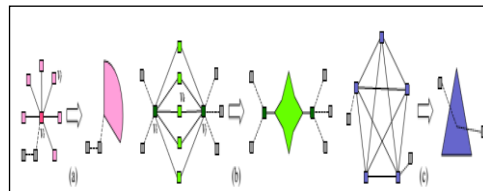
Wathsala Widanagamaachchi, Cameron Christensen and Peer-Timo Bremer describe exploring and analyzing the temporal evolution of features in large-scale time-varying datasets is a common problem in many areas of science and engineering. One natural representation of such data is tracking graphs, i.e., constrained graph layouts that use one spatial dimension to indicate time and show the "tracks" of each feature as it evolves, merges or disappears. However, for practical data sets creating the corresponding optimal graph layouts that minimize the number of intersections can take hours to compute with existing techniques. Furthermore, the resulting graphs are often unmanageably large and complex even with an ideal layout. Finally, due to the cost of the layout, changing the feature definition, e.g. by changing an iso-value, or analyzing properly adjusted sub-graphs is infeasible [1-5].

METHODOLOGY

The idea of transforming the data and their relationships into an abstract view for exploring complex relationships and improving data understanding has been accepted as a viable means to analyze and visualize scientific data. Nevertheless, most of these representations lack enough guidance for user exploration and navigation. In many cases, users can only rely on low-level visual hints (such as the size and density of nodes and edges) to figure out the relationships with the underlying data through brushing and linking. This approach may work for small data with simple relationships, but becomes increasingly inefficient for larger data with greater complexity. Therefore, solutions that help users sift through the data and their relationships for cost-effective understanding are highly desirable. In this paper, we specifically focus on time-varying volume data visualization and investigate transition relationships among data items over time. We present a mining approach that automatically extracts features from a graph-based

representation for understanding time varying data. Beyond straightforward graph properties, users are given further guidance available through a series of graph analysis techniques including graph simplification, community detection, and visual recommendation.

Graph simplification condenses a large graph to a smaller one by abstracting known structures, such as fan, connector, and clique, presenting a less cluttered view for quick comprehension of the overall graph structure. Community detection organizes nodes with close relationships into groups, allowing visual comparison between groups of nodes instead of individual nodes. Visual recommendation automatically highlights individual nodes or node groups based on user selected items, enabling users to spend more time on the actual analysis instead of painstaking interaction. Furthermore, visual recommendation based on node groups actually recommends the groups according to their structural relations and suggests similar groups regardless of their volume values, spatial regions, and temporal ranges. When the data set is large and the relationships are complex.



Navigating and exploring the resulting graph is a daunting task. We show that the suite of high-level functions introduced provides the convenience and capabilities for graph exploration which are difficult or impossible to achieve through standard interaction techniques. Our solution thus represents a step forward in applying graph-based techniques for scientific data analysis and visualization [6-10].

Transition graph

We first partition the volume data at each time step into blocks. Each block is, for example, $N \times N \times N$ voxels. Then we group these blocks based on their spatial and temporal adjacency within a smalltime window w . Specifically, given a target

block bt , we first check its spatial or temporal neighbor bn to see whether it is similar to bt . If so, then bn is clustered to bt . We continue to compare the neighbors of bn with bt . This process repeats until no similar blocks could be found in w . Note that if bn is already clustered to another block, we will not check its neighbors when bn is reached. Finally, we derive directional transition probabilities among blocks to construct the transition graph. In the transition graph, a node denotes a state which represents a group of spatiotemporally neighboring blocks, and a directed edge between two states indicates their transition probability. In extreme cases, a state may represent a single block. all number of transitions from all the groups

(including itself). As such, a transition indicates a chance for one group to transfer to another group, and its probability measures how high the chance is. To draw the transition graph, we apply a two-step process [11-15].

Since we draw nodes with certain sizes, visual occlusion becomes unavoidable for a large graph. We therefore utilize four forces: bidirectional repulsive force, unidirectional repulsive force, spring force, and attractive force, to reduce the overlap while preserving the overall graph structure.

GRAPH SIMPLIFICATION

Visual clutter is common in a transition graph due to the presence of a large number of nodes and edges. We can reduce the clutter through edge reduction or node reduction. For edge reduction, edge simplification and compression techniques to identify the relations among nodes, group the nodes with similar edge configurations together, and draw a single edge to the group instead of edges to group members. In our work, we focus on node reduction and replace certain graph features with symbols in order to highlight important graph structures. Similar to motif simplification, we select three graph features for simplification because they represent meaningful transition relations and do not impose restrictions to the order of simplification. Figure 1 illustrates these three graph features and their corresponding symbols for simplification. For graph simplification, we do not consider edge directions in the transition graph.

Fan detection

Our fan detection creates a data container called map that stores a set of tuples. Each tuple represents a pair of nodes: a central node and a leaf node. A tuple uses the central node as the key and the leaf node as the value. Thus, a set of tuples sharing the same key represents a fan feature. The detection algorithm first goes through all the nodes in the graph, finds the nodes with degree of one, and inserts them to the map. Then, we find those keys with more than one tuple in the map as the central nodes and their corresponding values as leaf nodes.

Connector detection

Similar to fan detection, our connector detection also creates a map. To detect a connector, we use the two ending nodes as the key and an intermediate node as the value. We first go through all the nodes in the graph, find all the nodes with degree of two, and insert them to the map. If any key has more than two tuples in the map, the keys are the ending nodes and their corresponding values are intermediate nodes [16-20].

Clique detection

This algorithm has a time complexity of $3|V|/3$, where $|V|$ is the number of nodes in the graph. Since this algorithm finds all the cliques, the cliques may share nodes. To avoid this situation, we utilize a greedy algorithm to select cliques in an iterative manner. In each iteration, we always select the largest clique and rule out all other cliques that share node(s) with any previously selected clique.

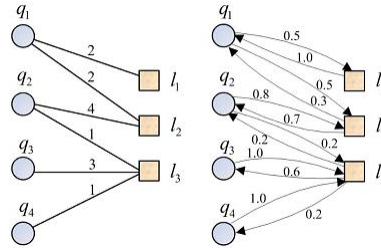
Community recommendation

Besides node recommendation, we also recommend similar communities when a community is selected. For simplicity, we first convert each community to an unweighted, undirected graph. Then we treat the similarity between two graphs as the similarity between the two corresponding communities.

Our community detection and recommendation algorithms have the following limitations which we would like to improve. First, the community detection algorithm automatically suggests the number of communities. In some cases, this leads to a large number of communities with each having a small number of nodes. We will seek a different solution which allows users to specify the number of communities. Second, unlike community detection, we would like to allow users to select a subgraph of interest and apply graph alignment techniques to suggest similar subgraph. This would give more flexibility to users in their selection and matching. Third, the community recommendation algorithm treats each community as an unweighted, undirected graph. provides us a measure of the similarity between communities. However, the

measure does not consider other information, such as node type, node importance, edge weight, and edge direction. We would like to seek a better solution that incorporates these graph attributes into similarity measure design. Finally, we will allow domain scientists to define general patterns and analysis procedures of interest based on their

Graph construction



Fortunately, on the Web, no matter what types of data sources are used for recommendations, in most cases, these data sources can be modeled in the form of various types of graphs. If we can design a general graph recommendation algorithm, we can solve many recommendation problems on the Web. However, when designing such a framework for recommendations on the Web, we still face several challenges that need to be addressed. The first challenge is that it is not easy to recommend latent semantically relevant results to users. Take Query Suggestion as an example; there are several outstanding issues that can potentially degrade the quality of the recommendations, which merit investigation. The first one is the ambiguity which commonly exists in the natural language. Queries containing ambiguous terms may confuse the algorithms which do not satisfy the information needs of users. Another consideration is that users tend to submit short queries consisting of only one or two terms under most circumstances, and short queries are more likely to be ambiguous. Through the analysis of a commercial search engine's query logs recorded over three months in 2006, we observe that 19.4 percent of Web queries are single term queries, and further 30.5 percent of Web queries contain only two terms. Third, in most cases, the reason why users perform a search is because they have little or even no knowledge about the topic they are searching for. In order to

domain knowledge, enabling more customized analytics.

QUERY SUGGESTION

Query Suggestion is a technique widely employed by commercial search engines to provide related queries to users' information need.

find satisfactory answers, users have to rephrase their queries constantly. The second challenge is how to take into account the personalization feature. Personalization is desirable for many scenarios where different users have different information needs. As an example, Amazon.com has been the early adopter of personalization technology to recommend products to shoppers on its site, based upon their previous purchases. Amazon makes an extensive use of collaborative filtering in its personalization technology. The adoption of personalization will not only filter out irrelevant information to a person, but also provide more specific information that is increasingly relevant to a person's interests.

The last challenge is that it is time consuming and inefficient to design different recommendation algorithms for different recommendation tasks. Actually, most of these recommendation problems have some common features, recommendation tasks on the Web. Moreover, most of existing methods are complicated and require tuning a large number of parameters. In this paper, aiming at solving the problems analyzed above, we propose a general framework for the recommendations on the Web. It is a general method, which can be utilized to many recommendation tasks on the Web. It can provide latent semantically relevant results to the original information need. This model provide a natural treatment for personalized recommendations. The

designed recommendation algorithm is scalable to very large data sets.

Image recommendation

Besides query suggestion, another interesting recommendation application on the Web is image recommendation. Image recommendation systems, like Photoree,⁸ focus on recommending interesting images to Web users based on users' preference. Normally, these systems first ask users to rate some images as they like or dislike, and then recommend images to the users based on the tastes of the users. In the academia, a few tasks are proposed to solve the image recommendation problems since this is a relatively new field and analyzing the image contents is a challenge job. Recently, by employing the Flickr data set, proposed a context-based image search and recommendation method to improve the image search quality and recommend related images and tags. However, since it is a context-based method, the computational complexity is very high and it cannot scale to large data sets. While in our framework proposed in this paper, by diffusing on the image-tag bipartite graph with one or more images, we can accurately and efficiently suggest semantically relevant non-personalized or personalized images to the users. In general, comparing with previous work, our work is a general framework which can be effectively, efficiently, and naturally applied to most of the recommendation tasks on the Web.

Query suggestion

In order to recommend relevant queries to Web users, a valuable technique, query suggestion, has been employed by some prominent commercial search engines, such as Yahoo!,³ Live Search,⁴ Ask,⁵ and Google.⁶ However, due to commercial reasons, a few public papers have been released to reveal the methods they adopt. The goal of query suggestion is similar to that of query expansion query substitution and query refinement which all focus on understanding users' search intentions and improving the queries submitted by users. Query suggestion is closely related to query expansion or query substitution, which extends the original query with new search terms to narrow down the scope of the search. But different from

query expansion, query suggestion aims to suggest full queries that have been formulated by previous users so that query integrity and coherence are preserved in the suggested queries. Query refinement is another closely related notion, since the objective of query refinement is interactively recommending new queries related to a particular query. In local (i.e., query-dependent documents) and global (i.e., the whole corpus) documents are employed in query expansion by applying the measure of global analysis to the selection of query terms in local feedback. Although experimental results show that this method is generally more effective than global analysis, it performs worse than the query expansion method proposed in based on user interactions recorded in user logs.

In another approach reported in anchor texts are employed for the purpose of query refinement. This work is based on the observation that Web queries and anchor texts are highly similar. These methods employ different kinds of data sources (documents, anchor texts, query logs, etc.) for suggesting queries. The main disadvantage of these two algorithms is that they ignore the rich information embedded in the query-click bipartite graph, and consider only queries that appear in the query logs, potentially losing the opportunity to recommend highly semantically related queries to users. Cao et al. developed a context-aware query suggestion method by mining click through and session data. This work first extracts some concepts from the click through data by building clusters. Then, these concepts as well as the query sessions are employed to build a concept sequence suffix tree for query suggestion. Recently, Mei et al. proposed a general query suggestion method using hitting time on the query-click bipartite graph. This method can generate semantically relevant queries to users' information needs. The main advantage of this work is that it can suggest some long tail queries (infrequent queries) to users. However, this is also the disadvantage of this approach since sometimes it may accidentally rank the infrequent queries highly in the results while potentially downgrades the ranks of the most related queries.

CONCLUSION

In this project proposes a graph based method to find the relationships over data. As these data's are used for recommendation purposes. Since the web usage is also calculated in order to derive accurate relationships among data. In this project plan to extend proposed work to handle multivariate data sets. We can either fuse multiple variables into one type of node, or construct one type of node for each individual variable and visualize the relations between variables using compound graphs. We will also investigate time-evolving graphs derived from scientific data sets for identifying temporal hotspots, detecting anomaly, and aligning multiple graphs for finding common features and distinct patterns. As the connection can be detected by two methods such as query based URL and converted query to URL.

These graphs are mined using user given data and recommendations are made. This graph employs for identifying exact connection between data's. Community detection organizes nodes with close relationships into groups, allowing visual comparison between groups of nodes instead of individual nodes. The project has covered almost all the requirement. Further requirements and improvements can easily be done since the coding in mainly structured or modular in nature. Improvements can be appended by changing the existing modules or adding new modules.

Several areas to be developed in future, so the application must be upgraded for the new ones required and it is possible to modifications according to new requirements and specifications. The project deals with the query and URLs which is stored in the database, and it will be displayed the by the use of web browser control

REFERENCES

- [1]. Y. Gu and C. Wang, "iTree: Exploring time-varying data using indexable tree", In Proceedings of IEEE Pacific Visualization Symposium, 2013, 137–144.
- [2]. J. Heer, M. Agrawala, and W. Willett, "Generalized selection via interactive query relaxation", In Proceedings of ACM SIGCHI Conference, 2008, 959–968.
- [3]. J. Ma, C. Wang, and C.-K. Shene, "Flow Graph: A compound hierarchical graph for flow field exploration", In Proceedings of IEEE Pacific Visualization Symposium, 2013, 233–240.
- [4]. A. Robles-Kelly and E. R. Hancock, "String edit distance, random walks and graph matching", International Journal of Pattern Recognition and Artificial Intelligence, 18(3), 2004, 315–327.
- [5]. F. van Ham, M. Wattenberg, and F. B. Viegas, "Mapping text with phrase nets", IEEE Transactions on Visualization and Computer Graphics, 15(6), 2009, 1169–1176.
- [6]. C. Wang, "A survey of graph-based representations and techniques for scientific visualization", In Eurographics Conference on Visualization State of The Art Reports, 2015, 41–60, 2015..
- [7]. W. Widanagamaachchi, C. Christensen, P.-T. Bremer, and V. Pascucci, "Interactive exploration of large-scale time-varying data using dynamic tracking graphs", In Proceedings of IEEE Symposium on Large Data Analysis and Visualization, 2012, 9–17.
- [8]. J. Xie and B. K. Szymanski, "Towards linear time overlapping community detection in social networks", In Proceedings of Pacific Asia Knowledge Discovery and Data Mining Conference, 2012, 25–36.
- [9]. C. Dunne and B. Shneiderman, "Motif simplification: Improving network visualization readability with fan, connector, and clique glyphs", In Proceedings of ACM SIGCHI Conference, 2013, 3247–3256.
- [10]. C. Dunne and B. Shneiderman, "Motif simplification: Improving network visualization readability with fan, connector, and clique glyphs", In Proceedings of ACM SIGCHI Conference, 2013, 3247–3256.
- [11]. Z. Du, Y.-J. Chiang and H.-W. Shen, "Out-of-core volume rendering for time-varying fields using a space-partitioning time (SPT) tree", In Proceedings of IEEE Pacific Visualization Symposium, 2009, 73–80.

- [12]. Viégas, F., Wattenberg, M., van Ham, F., Kriss, J., and McKeon, M. “Many-Eyes: A Site for Visualization at InternetScale”, Proc. IEEE InfoVis ’07, 2007, 1121-1128.
- [13]. C.-M. Chen, L. Xu, T.-Y. Lee, and H.-W. Shen. “A flow-guided file layout for out-of-core streamline computation”, In Proceedings of IEEE Pacific Visualization Symposium, 2012, 145–152.
- [14]. A. Robles-Kelly and E. R. Hancock. “A graph-spectral approach to shape-fromshading”, IEEE Trans. on Image Processing, To Appear, 2004.
- [15]. C. Collins, DocuBurst: Document Content Visualization Using Language Structure. In Proceedings of IEEE Symposium on Information Visualization, Poster Compendium, IEEE CS Press, 2006.
- [16]. E. Tomita, T. Seki, “An efficient branch-and-bound algorithm for finding a maximum clique”, DMTCS 2003, Lecture Notes in Computer Science, 2731, 2003, 278–289;
- [17]. D. Laney, P. T. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci, “Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities”, IEEE Transactions on Visualization and Computer Graphics, 12(5), 2006, 1053-1060.
- [18]. Lee, C., Reid, F., McDaid, A., Hurley, “N.: Detecting highly overlapping community structure by greedy clique expansion”, In: snakdd. 2010, 33–42.
- [19]. Smith, M., Shneiderman, B., Milic-Frayling, N., Rodrigues, E. M., Barash, V., Dunne, C., Capone, T., Perer, A., and Gleave, E. Analyzing (social media) networks with NodeXL. In ACM C&T 2009, 255–264.
- [20]. E. Tomita, A. Tanaka, H. Takahashi, The worst-case time complexity for generating all maximal cliques, COCOON 2004, Lecture Notes in Computer Science, 3106, 2004, 161–170.