Centaurs - a Component Based Framework to Mine Large Graphs

Ana Paula Appel[1,2], Estevam Rafael Hruschka Junior[1]

[1] Computer Science Departament – Federal Univeristy of São Carlos (UFSCar)
Rodovia Washington Luís, km 235 - SP-310 – São Carlos – SP – Brazil
{anaappel, estevam}@dc.ufscar.br
[2] Engineering and Computer Science Department – Federal University of Espírito Santo - CEUNES/UFES
Rodovia BR 101 Norte, Km. 60, – São Mateus, ES Brazil
anaappel@ceunes.ufes.br

Abstract. The increase of the amount of data represented as a graph, like complex networks, motivated the creation of a new research area called graph mining. This work proposes a new framework based on components, called `Centaurs`, to mine data represented as a graph. The main idea of Centaurs is to couple community detection and link prediction algorithms to mine missing edges that were missed during the graph building process. Graph preprocessing and storage algorithms are also explored in this proposal, given that large graphs cannot always be storage in main memory only. The main `Centaurs`'s case study is the Read the Web project that aims to build a graph to represent knowledge extract from the Web based on a never ending learning algorithm.

Categories and Subject Descriptors: H. Information Systems [**H.m. Miscellaneous**]: Databases

Keywords: graph mining, framework

1. INTRODUCTION

Over the past years the amount of data collected and stored has been substantially increased and it has been powered mainly by the World Wide Web expansion. Part of the data coming from the Web can be represented as graphs, such as page link structures, social and academic networks (Facebook, Orkut, DBLP) and so on. When substantial non-trivial topological features are present in a graph, with patterns of connection between their elements that are neither purely regular nor purely random, the graph-based representation can be called a Complex Network [Newman 2010]. The great amount of data and the new representation approach have motivated the start-up of a research area called graph mining, which has as its main focus to investigate, propose and develop new algorithms designed to mine complex networks.

The study of complex networks revealed useful properties related to data represented by graphs. Such properties reveal relevant common characteristics of different complex networks. Some interesting and relevant properties are: the power-law degree distributions [Albert et al. 1999], [Adamic et al. 2000], the diameter shrinkage present in evolving networks [Leskovec et al. 2005], the Small World phenomenon [Milgram 1967], among others. These patterns help us to understand not only the interaction among human being and social networks [Leskovec et al. 2008] but also the dissemination of information and diseases [Chakrabarti et al. 2008], intrusion detection [Fortunato 2010] and so on. A nice graph mining review can be found in [Newman 2003].

Considering, however, that graph mining is a new research area and the wide range of applications

Authors would like to thanks the support of Carnegie Mellon University, FAPESP, CNPQ and CAPES to the project.

of this research topic, there is no record (to our knowledge) of a framework designed to integrate and take advantage of coupling different graph mining algorithms in an easy way. The main motivation for such a coupled approach is to make the knowledge discovery in graphs an interactive and iterative process in which different algorithms (i.e. community detection and link prediction algorithms) can be used to reinforce learning and to boost the learning results. Another interesting issue which deserves more study and investigation is related to graph storage and pre-processing. It is important mainly because of the huge amount of data and the need of main memory use.

This article presents principles and main ideas of a new framework called `Centaurs` to adapt, integrate and combine graph mining algorithms to mine large graphs in main memory or in secondary memory. The initial focus of `Centaurs` is to integrate link prediction and community detection to find missing edges that were lost during the graph building process. Also, `Centaurs` will have as main case study the graph created by *Read the Web* project [Carlson et al. 2010]. The rest of this article is organized as follows: Section 2 presents the background needed for this work; Section 3 describes the *Read the Web* project and how `Centaurs` will work on it, Section 4 presents the `Centaurs` framework and finally Section 5 concludes the work.

2. BACKGROUND

A graph is a mathematical object composed by elements called vertexes and the connection between these elements called edges. Thus we can define a graph $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph without self-edges, where $\mathcal{V}$ is the set of vertices, also called nodes, and $\mathcal{E}$ is the set of edges. The number of edges directly connected to a node v_i is called its *node degree* $d(v_i)$.

Most of traditional data can be modeled as graphs. For example, a join operation result in a database system can be viewed as a graph and thus treated as a complex network. Transforming data into a graph is useful to understand the relation among the data elements. There are, on the other hand, some types of data, as ontologies, which can be easily viewed as a graph needing no transformation or adaptation. An ontology is a description of concepts and relationships that can exist for an agent or a community of agents [Terzi et al. 2003] and is commonly seen as a graph or a complex network. Therefore, graph mining algorithms (such as community detection and link prediction) have been widely applied to ontologies.

There is a significant amount of research related to our problem, which we categorize as community detection, link prediction, graph pre-processing and graph mining framework.

Community Detection: Communities are set of nodes that share some properties in a complex network [Fortunato 2010], [Leskovec et al. 2009]. Community detection is the task of finding these set of nodes. Usually the number of edges among nodes is used to determine if a set of nodes is a community or not given that nodes that are in the same community have more edges among each other than nodes from different communities.

Link Prediction: Among the link prediction techniques recently proposed (e.g., [Hasan et al. 2006], [Kashima et al. 2009], [Kunegis and Lommatzsch 2009], [Lu and Zhou 2009], [Acar et al. 2009]), we highlight the ones based in graph structural properties as [Liben-Nowell and Kleinberg 2003], [Huang 2006]. An interesting technique is presented by [Clauset et al. 2008], which uses a naive community detection approach to help in link prediction. However, this approach only works for higher clustered networks. In spite of this, the proposed idea is very useful, since the probability of an edge existing between nodes in the same community is higher than between nodes from different communities. Usually link prediction is used to find edges that will appear when a graph evolves. However, it can also be used to find missing edges. Link prediction can use graph structural information and relational characteristics, such as attributes related with graph's node. This kind of approach is more common in relational or multi-relational learning [Getoor and Diehl 2005], [Hasan et al. 2006], [Taskar et al. 2004], [Popescul et al. 2003], [Backstrom and Leskovec 2011].

Graph pre-processing: Among the graph pre-processing techniques we can highlight graph sampling methods such as Forest Fire [Leskovec et al. 2007], random sampling for edges and nodes [Leskovec and Faloutsos 2006] and others [Krishnamurthy et al. 2007], [Tsourakakis et al. 2009]. Usually sampling methods have a special purpose, as in [Tsourakakis et al. 2009], whose goal is to find an approximate number of triangles in the graph. Graph partitioning can be use not only to find communities but also to help pre-process large graphs, given that a partitioned graph can be threated easier than a large one. These techniques are used to partition the graph in k pieces with equal number of nodes. Usually the partitioning technique tries to minimize the number of edges between nodes of different partitions. Graph partitioning is also a useful technique for a wide number of applications, such as load balancing for parallel computing, graph storage, graph pre-processing and so on. The most traditional graph partitioning algorithm is known as METIS [Karypis and Kumar 1998].

Framework for graph mining: Over the last years, a large amount of graph mining algorithms have been proposed. However, there are a few works that combine graph mining algorithms to build a framework to mine large complex networks. The work presented in [Kang et al. 2009] shows a framework to mine large complex networks using the `MAP/REDUCE` [1] architecture with `HADOOP` [2]. This type of architecture allows only parallel algorithms and needs a specific hardware. Thus, sometimes this architecture becomes unfeasible due to expensive cost of such hardware. Another framework to mine graphs is proposed by [Inokuchi et al. 2005], however in this work the graph database is composed by a large number of small graphs instead of only one large graph in a database, which is the focus of our work.

3. READ THE WEB PROJECT

The main goal of the *Read the Web* project[3] is to develop a probabilistic, symbolic knowledge base that mirrors the content of the web. If successful, this will make text information on the web available in computer-understandable form, enabling much more sophisticated information retrieval, natural language understanding, and general problem solving. To achieve it, the main idea is to couple the learning of multiple extractors organized around a shared knowledge base (KB) that is incrementally and continuously grown and used by a collection of learning subsystem components that implement complementary knowledge extraction methods [Carlson 2010]. Initial results [Carlson et al. 2010] revealed the importance of our design principle of coupling components which make mostly independent errors. The RTW system is a never ending language learning called NELL, as shown in Figure 1 (a), currently NELL has 4 components namely, the Coupled Pattern Learner (CPL), the Coupled SEAL (CSEAL), the Coupled Morphological Classifier(CMC) and the Rule Learner (RL).

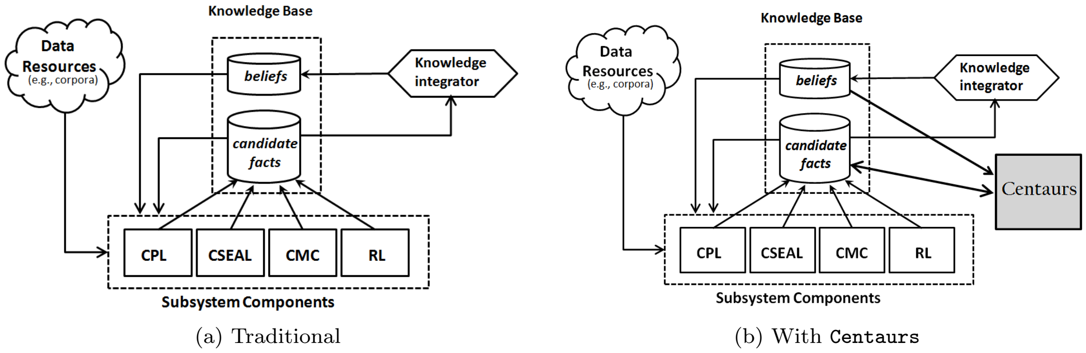
CPL is described in more details in [Carlson et al. 2010] and works as a free-text extractor which learns and uses contextual patterns like "mayor of X" and "X plays for Y" to extract instances of categories and relations. CSEAL is based on [Wang and Cohen 2008] and implements a semi-structured extractor which queries the Internet with sets of beliefs from each category or relation, and then mines lists and tables to extract novel instances of the corresponding predicate. CMC is a simple set of binary L2-regularized logistic regression models which classify noun phrases based on various morphological features (words, capitalization, affixes, parts-of-speech, etc.). Finally, the Rule Learner is a first-order relational learning algorithm similar to FOIL [Quinlan and Cameron-Jones 1993], which learns probabilistic Horn clauses from the RTW ontology that can be represented as a graph called *rtwgraph*, the entities become nodes and relations edges. A visualization of the giant connected component of *rtwgraph* is presented in Figure 2[4].

[1] `http://labs.google.com/papers/mapreduce.html`

[2] `http://hadoop.apache.org/`

[3] `http://rtw.ml.cmu.edu/rtw/`

[4] The authors used GraphViz to visualization



(a) Traditional (b) With `Centaurs`

Fig. 1. NELL - Never Ending Language Learning architeture

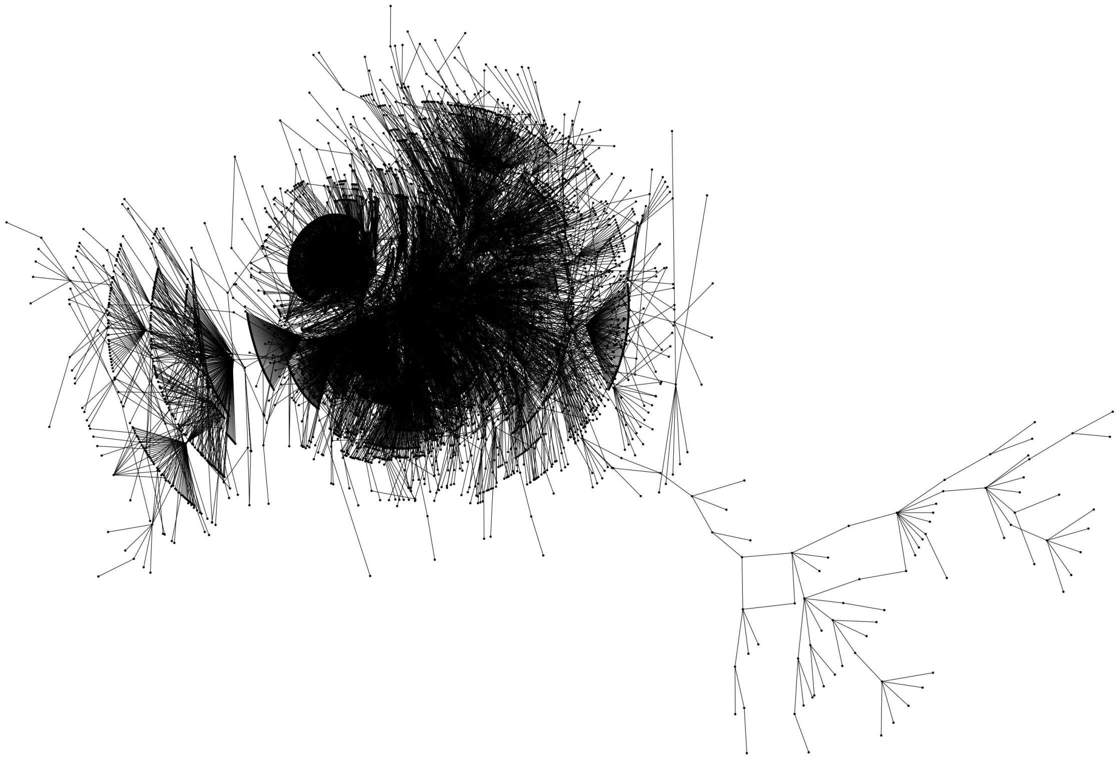


Fig. 2. The giant connected component of *rtwgraph*.

Analyzing results presented by RL, it is possible to notice that finding new relations, hidden in the *rtwgraph*, might help the system to learn better and even help to extend the initial ontology. Thus, a new independent method, based on the idea of use “Link Prediction” techniques can play an important role in the Read The Web System. Therefore, `Centaurs` is the new NELL’s component as shown in Figure 1 (b).

An example of how `Centaurs` works is presented in Figure 3. The outside triangle represents the nodes categories, that is, the entities and the label in the edges are the relations. The inside triangle has the instanced entities. Usually, it is not common to say that somebody plays soccer but that somebody plays for a soccer time [Carlson et al. 2010]. Thus, NELL’s text and html-based learning components are usually not able to learn the relationship between Sport and Athlete. An example is presented in Figure 3, in which there is Neymar (a Brazilian soccer player) as an Athlete, Soccer

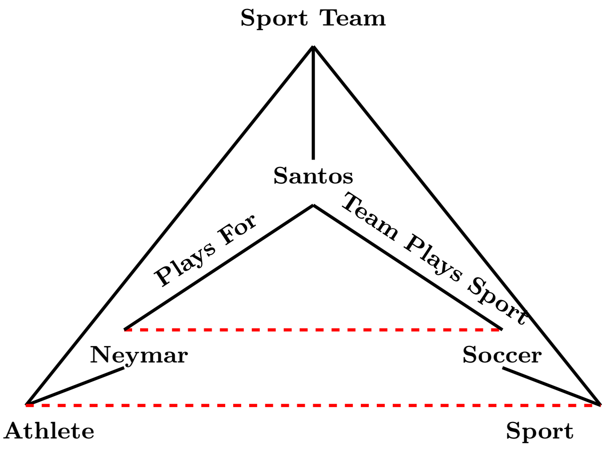


Fig. 3. An example of edges predict by `Centaurs`: lines represent the relations that already exists in KB and dashed lines represent the edges predict (rules and relations) by `Centaurs`

as a Sport and Santos (a Brazilian soccer team) as Sport Team. `Centaurs` is able to analyses this graph and using graph mining algorithms, as link prediction techniques, to find the hidden relation between Neymar and Soccer (dashed line in Figure 3) and therefore the relation Athlete and Sport can be induced.

At first glance, predicting a link in *rtwgraph* may seem as just insert an edge connecting two nodes that are two step way from each other and forming as many triangles as possible. However, this can create a clause and a relation that do not make any sense. For example, `HasOfficeInCountry(company, country)` and `CountryStates(country, stateorprovince)`. If we only link company with `stateorprovince`, we might link a company with states or provinces when there is no company there. Thus the question is decide whether two step way nodes should be connected, becoming a triangle, or not.

One might say that NELL already has the RL component that infer rules. However, the RL does not infer new relations and new rules using new relations. RL only finds rules that already exist in NELL's KB composing a triangle. `Centaurs` can do what RL does and also infer new relations and rules. The abilit of NELL learns new relations is an important task because it helps NELL's knowledge base grows. Also, news relations make NELL learns more and better.

4. CENTAURS

The aim of this work is to propose a framework called `Centaurs` to mine large complex networks and to be used as a new component of the *Read the Web* project combining graph mine algorithms, specially those related to community detection and link prediction, to find missing edges that were lost during the building process of the *rtwgraph*.

However, the *rtwgraph* will not be the only input supported by `Centaurs`. The proposed framework will be generic enough to allow any data modeled as a graph be mined by it. Thus, in addition to be a new *Read the Web* component, `Centaurs` is a components-based framework itself, so new algorithms (components) can be easily added to it and the NELL design principle of using components which make mostly independent errors will also be present in `Centaurs`.

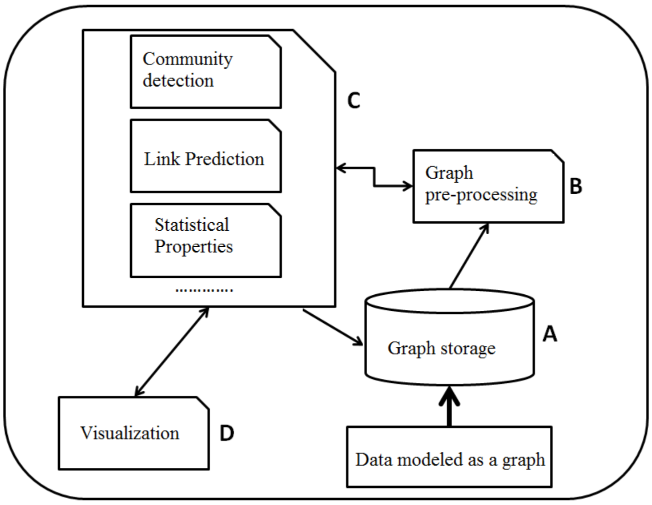


Fig. 4. Centaurs architecture

4.1 Centaurs architecture

The Centaurs architecture is presented in Figure 4. The Centaurs framework has four main parts: storage (A), pre-processing (B), mining (C) and visualization (D).

The storage: All graphs mined using Centaurs are stored on disk as presented by Figure 4 (B). A relational database is used and a proper index structure might be employed.

The pre-processing: Considering Centaurs uses disk to store graphs and most graph mining algorithms available consider the graph be available in main memory, Centaurs uses components to pre-process the input graph represented by box (B) in Figure 4. In the first stage of Centaurs the main algorithm used is the graph partition method METIS [Karypis and Kumar 1998]. This method allows partitioning the nodes of a graph in p roughly equal parts, such that the number of edges connecting nodes in different parts is minimized. Shortly, METIS works as follow: a graph G is first coarsened down to a few hundred nodes, a bisection of this much smaller graph is computed, and then this partition is projected back towards the original graph, by periodically refining the partition.

Mining: The main idea behind the mining component of Centaurs framework, represented by box (C) in Figure 4, is coupling graph mining algorithms by taking advantage of each algorithm's strength to obtain better results. The initial focus is to find missing edges, so we propose coupling traditional link prediction [Liben-Nowell and Kleinberg 2003], as Adamic/Adar, Jaccardt́s coefficient, cosine similarity, common neighbor and so on, and *rtwgraph* information such as node categories, for example *sport* is a category that node soccer belongs. Category and relation nodes can have a large number of instances, so if we only consider *rtwgraph* to predict a rule, the same rule could be predicted as many times as the number of nodes. To avoid this, we decided to group the nodes into their categories and then select the rules that might be predicted based on traditional link predictions measures. Category nodes are used to find the open triangles because, if they are ignored the wrong categorized open triangles might be generated. For example a node representing *Arnold Schwarzenegger* will be in both categories: actor and governor, but its connection with movie comes from actor and not governor. We are also planning couple link prediction and community detection techniques.

Visualization: As the goal is a graph mining framework with a wide range of applications, graph visualization methods are incorporated in Centaurs as represented by box (D) in by Figure 4. However, in `Centaurs` first version we will use only pre-existent tools, such as GraphViz[5], GMine [Rodrigues Jr. et al. 2006] and to statistical proprieties GNUPLOT[6].

5. CONCLUSION

This article described the design principles and the architecture of a new graph mining framework named `Centaurs`. As one can see, the proposed framework aggregates many research areas as graph mining, semantic web, database, visualization aiming at implementing a robust tool based on coupling independent components. `Centaurs` is an ambitious and complex system which to be developed in independent steps starting with the integration of traditional graph mining algorithms to find missing edges on *rtwgraph*. In the first stage, `Centaurs` uses traditional algoritms for partitioning, link prediction and community detection. The validation of the obtained results will be empirically performed using a comparison analysis strategy applied to the current NELL and the new version having `Centaurs` as a new component. It is expected that `Centaurs` helps NELL to improve its learning ability as well as extending its ontology by means of discovery missing relations (edges).

REFERENCES

Acar, E., Dunlavy, D. M., and Kolda, T. G. Link Prediction on Evolving Data Using Matrix and Tensor Factorizations. In *Proceedings of IEEE International Conference on Data Mining, Workshops*. Miami, USA, pp. 262–269, 2009.

Adamic, L. A., Huberman;, B. A., Barabási, A., Albert, R., Jeong, H., and Bianconi;, G. Power-Law Distribution of the World Wide Web. *Science* 287 (5461): 2115, March, 2000.

Albert, R., Jeong, H., and Barabasi, A.-L. The diameter of the World Wide Web. *Nature* 401 (6749): 130–131, Sep, 1999.

Backstrom, L. and Leskovec, J. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of International Conference on Web Search and Web Data Mining*. Hong Kong, China, pp. 635–644, 2011.

Carlson, A. *Coupled Semi-Supervised Learning*. Ph.D. thesis, Carnegie Mellon University, 2010.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E. R. H., and Mitchell, T. M. Toward an Architecture for Never-Ending Language Learning. In *Proceedings of Conference on Artificial Intelligence*. Atlanta, Georgia, USA, pp. 1–8, 2010.

Carlson, A., Betteridge, J., Wang, R. C., Jr., E. R. H., and Mitchell, T. M. Coupled Semi-Supervised Learning for Information Extraction. In *Proceedings of ACM International Conference on Web Search and Data Mining*. New York City, USA, pp. 101 – 110, 2010.

Chakrabarti, D., Wang, Y., Wang, C., Leskovec, J., and Faloutsos, C. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security* 10 (4): 1–26, 2008.

Clauset, A., Moore, C., and Newman, M. E. J. Hierarchical structure and the prediction of missing links in networks. *Nature* 453 (7191): 98–101, Nov, 2008.

Fortunato, S. Community detection in graphs. *Physics Reports* 486 (3-5): 75–174, February, 2010.

Getoor, L. and Diehl, C. P. Introduction to the special issue on link mining. *ACM SIGKDD Explorations Newsletter* 7 (2): 1–2, 2005.

Hasan, M. A., Chaoji, V., Salem, S., and Zaki, M. Link Prediction Using Supervised Learning. In *Proceedings of SDM Workshop on Link Analysis, Counterterrorism and Security*. Bethesda, USA, pp. 1–10, 2006.

Huang, Z. Link Prediction Based on Graph Topology: The Predictive Value of the Generalized Clustering Coefficient. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, PA, USA, pp. 1–8, 2006.

Inokuchi, A., Washio, T., and Motoda, H. A General Framework for Mining Frequent Subgraphs from Labeled Graphs. *Fundamenta Informaticae* 66 (1-2): 53–82, 2005.

Kang, U., Tsourakakis, C., and Faloutsos, C. PEGASUS: A Peta-Scale Graph Mining System - Implementation and Observations. In *Proceedings of IEEE International Conference on Data Mining*. Miami, USA, pp. 1–10, 2009.

Karypis, G. and Kumar, V. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing* 20 (1): 359–392, 1998.

[5] `http://www.graphviz.org/`

[6] `http://www.gnuplot.info/`

Kashima, H., Kato, T., Yamanishi, Y., Sugiyama, M., and Tsuda, K. Link Propagation: A Fast Semi-supervised Learning Algorithm for Link Prediction. In *Proceedings of SIAM Conference on Data Mining*. Sparks, USA, pp. 1099–1110, 2009.

Krishnamurthy, V., Faloutsos, M., Chrobak, M., Cui, J.-H., Lao, L., and Percus, A. G. Sampling large internet topologies for simulation purposes. *Computer Networks* 51 (15): 4284–4302, 2007.

Kunegis, J. and Lommatzsch, A. Learning Spectral Graph Transformations for Link Prediction. In *Proceedings of International Conference On Machine Learning*. Montreal, Canada, pp. 561–568, 2009.

Leskovec, J., Backstrom, L., Kumar, R., and Tomkins, A. Microscopic evolution of social networks. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, NV, USA, pp. 462–470, 2008.

Leskovec, J. and Faloutsos, C. Sampling from large graphs. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, PA, USA, pp. 631–636, 2006.

Leskovec, J., Kleinberg, J., and Faloutsos, C. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA, pp. 177–187, 2005.

Leskovec, J., Kleinberg, J. M., and Faloutsos, C. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data* 1 (1): 1 – 40, 2007.

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics* 6 (1): 29–123, 2009.

Liben-Nowell, D. and Kleinberg, J. The link prediction problem for social networks. In *Proceedings of International Conference on Information and Knowledge Management*. New York, NY, USA, pp. 556–559, 2003.

Lu, L. and Zhou, T. Role of weak ties in link prediction of complex networks. In *Proceedings of ACM International Workshop on Complex Networks in Information and Knowledge Management*. Hong Kong, China, pp. 55–58, 2009.

Milgram, S. The small world problem. *Psychology Today* 1 (1): 61–67, 1967.

Newman, M. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.

Newman, M. E. J. The structure and function of complex networks. *SIAM Review* 45 (2): 167–256, 2003.

Popescul, A., Popescul, R., and Ungar, L. H. Statistical relational learning for link prediction. In *Proceedings of Workshop on Learning Statistical Models from Relational Data*. Acapulco, Mexico, pp. 1–7, 2003.

Quinlan, J. R. and Cameron-Jones, R. M. FOIL: A Midterm Report. In *Proceedings of European Conference on Machine Learning*. London, UK, pp. 1–20, 1993.

Rodrigues Jr., J. F., Tong, H., Traina, A. J. M., Faloutsos, C., and Leskovec, J. GMine: A System for Scalable, Interactive Graph Visualization and Mining. In *Proceedings of International Conference on Very Large Data Bases*. Seoul, South Korea, pp. 1195–1198, 2006.

Taskar, B., Wong, M., Abbeel, P., and Koller, D. Link prediction in relational data. In *Proceedings of Neural Information Processing Systems*. Vancouver, Canada, pp. 1–8, 2004.

Terzi, E., Vakali, A., Hacid, M.-S., Dpt, I., and Lyon, U. C. B. Knowledge representation, ontologies, and the semantic web. In *Proceedings of Asian-Pacific Web Conference*. Xian, China, pp. 382–387, 2003.

Tsourakakis, C. E., Kang, U., Miller, G. L., and Faloutsos, C. DOULION: counting triangles in massive graphs with a coin. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Paris, France, pp. 837–846, 2009.

Wang, R. C. and Cohen, W. W. Iterative set expansion of named entities using the web. In *Proceedings of IEEE International Conference on Data Mining*. Washington, DC, USA, pp. 1091–1096, 2008.