Link Prediction Based on Generalized Cluster Information

Jungeun Kim, Minsoo Choy, Daehoon Kim, and U Kang
Korea Advanced Institute of Science and Technology
{je_kim, minsoo.choy, daehoonkim}@kaist.ac.kr, ukang@cs.kaist.ac.kr

ABSTRACT

Understanding of which new interactions among data objects are likely to occur in the future is crucial for a deeper understanding of network dynamics and evolution. This question is largely unexplored except a local neighborhood perspective, partly owing to the difficulty in finding major factors which heavily affect the link prediction problem. In this paper, we propose LPCSP, a novel link prediction method which exploits the generalized cluster information containing cluster relations and cluster evolution information. Experiments show that our proposed LPCSP is accurate, scalable, and useful for link prediction on real world graphs.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining

Keywords

Link Prediction, Cluster Relation, Cluster Evolution

1. INTRODUCTION

Link prediction in complex networks has attracted a lot of attentions from various domains including computer science and physics. Great efforts have therefore been made to define the similarity between two vertices since link prediction algorithms typically assume that similar vertices are likely to be connected [4]. A cluster is a densely connected sub-graph in the entire graph, which means the members in the same cluster are highly related to each other and have similar properties. Thus, cluster information can be utilized as a factor having a powerful predictive value. Although some state of the art link prediction methods consider cluster information [7, 5], they do not consider the relations and evolution of clusters, and thus they do not fully exploit cluster information. In this paper, we propose LPCSP (Link Prediction inferred from Cluster Similarity and cluster Power), a link prediction method which exploits both static and temporal cluster information. In the static perspective, LPCSP uses cluster similarity and static cluster power defined by cluster's structure. LPCSP gives more weight when cluster similarity is higher and the structure of the cluster is more densely connected. In the temporal perspective, LPCSP gives more weight when the structure of the cluster is more strongly evolving. Extensive experiments show that our proposed LPCSP is accurate, scalable, and useful for link prediction on real world graphs.

2. PROPOSED METHOD: LPCSP

LPCSP uses generalized cluster information which consists of two major factors: (i) *cluster similarity* and (ii) *cluster power*. Those can be used to improve baseline link prediction methods. An overview of LPCSP is shown in Fig. 1.

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2578807.

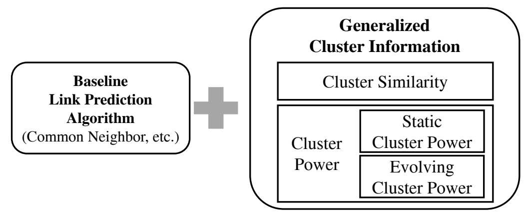

Figure 1: Overview of LPCSP method. LPCSP integrates the generalized cluster information with any base link prediction algorithm to achieve better prediction power.

In addition, LPCSP has an advantage of using any link prediction method as a *plug-in*. In other words, LPCSP can be regarded as a general method for link prediction. In this paper, we use several well-known link prediction metrics: Common Neighbors (CN), Adamic/Adar (AA), Resource Allocation (RA), and Preferential Attachment (PA).

To find clusters, we use Multi-level modularity graph clustering algorithm [1] which discovers high modularity partitions in large networks within a reasonable amount of time.

2.1 Cluster Similarity

Intuitively, two clusters are similar if there are many inter edges between them. Based on the intuition, we construct a *cluster graph* whose vertices are clusters and edges denote interactions between clusters. For two clusters α and β, let $|E_{\alpha,\beta}|$ be the number of existing inter edges between them and $|\alpha|$ be the number of vertices in α. Then, the edge weight between clusters α and β is defined as the ratio of the number of existing inter edges to the number of all possible inter edges (i.e., $\frac{|E_{\alpha,\beta}|}{|\alpha||\beta|}$). Finally, the cluster similarity is computed by applying *Random Walk with Restart* [6] on the cluster graph.

2.2 Cluster Power

We define a notion of *cluster power* based on both the static and temporal perspectives. The overall cluster power CP_β of a cluster β is computed by combining static cluster power CPS_β and evolving cluster power CPE_β. For the combination, we multiply the two values (i.e., $CP_\beta = CPS_\beta \cdot CPE_\beta$), but there can be other alternative ways such as summation, etc.

Static cluster power: A well-known result on graph evolution research [3] is that there exists a power law relationship, called "densification power law", between the numbers of edges ($\|G\|$) and vertices ($|G|$) of a graph G: $\|G\| = |G|^{\mathcal{R}}$ where the exponent R is a *densification coefficient* of the graph. We use R as the static cluster power to represent the time-invariant density of a cluster.

Evolving cluster power: For timestamps t_1 and t_2 ($t_1 < t_2$) and a cluster β discovered at t_2, we identify β's previous cluster α by computing $\arg\max_{\gamma} \min(\frac{|\beta\cap\gamma|}{|\beta|}, \frac{|\beta\cap\gamma|}{|\gamma|})$ over the clusters γ at t_1, where $\beta \cap \gamma$ represents a set of vertices which belong to both β and γ. In other words, we select the cluster having the maximum mutual membership ratio. However, if the maximum ratio is less than a threshold (e.g., 0.1), we assume that there was no previous cluster of β at t_1 and do not consider cluster evolution for such case. Finally, the evolving cluster power CPE_β of cluster β is computed by $CPS_\beta - CPS_\alpha + 1$ if α is found, or 1 otherwise.

2.3 LPCSP Measure

The LPCSP measure for two vertices x and y is defined as follows: $base(x, y) \cdot \sum_v \frac{Sim(C_x, C_v) + Sim(C_v, C_y)}{2} \cdot Sim(C_x, C_y) \cdot CP_\beta$, where $base(x, y)$ is a baseline link prediction method like Adamic/Adar, $Sim()$ is a function for computing the similarity between two clusters, C_x is a cluster the vertex x belongs to, and v is a common neighbor of x and y. If $C_x \neq C_y$, CP_β is ignored in the computation (i.e., $CP_\beta = 1$).

3. EXPERIMENTS

We test LPCSP on synthetic and 5 real world datasets [1]: DBLP, Slashdot (social network), AS-733 (autonomous system), Oregon (autonomous system), and Gnutella (peer-to-peer network). These networks range in size from 3,028 vertices and 11,705 edges (Oregon) to 214,408 vertices and 563,688 edges (DBLP).

3.1 Accuracy

We compare the performance of LPCSP with four baseline link prediction algorithms: (i) CN (Common Neighbor), (ii) AA (Adamic/Adar), (iii) RA (Resource Allocation), and (iv) PA (Preferential Attachment). To evaluate the performance, we draw an ROC (Receiver Operating characteristics) curve and get AUC (Area Under the ROC Curve) score for each method [2].

Table 1 shows the detailed results of our experiments. The term "Naive" in this table represents a baseline method itself. In most cases, LPCSP shows significantly better performances. In addition, LPCSP performs the best in four out of five datasets. Even in the Oregon graph where the LPCSP performs worse than the baseline, the difference is very small (0.7106 and 0.6951).

Table 1: Comparisons of AUC scores between baseline and LPCSP on all datasets. The best score for each dataset is in bold.

	DBLP		Slashdot		AS-733		Oregon		Gnutella	
	Naive	LPCSP	Naive	LPCSP	Naive	LPCSP	Naive	LPCSP	Naive	LPCSP
CN	0.5472	0.6163	0.6380	**0.7192**	0.6478	**0.6954**	0.5831	0.6449	0.3666	0.3631
AA	0.5783	0.6314	0.6281	0.6952	0.5594	0.6127	0.6284	0.6593	0.3216	0.4680
RA	0.5763	0.6350	0.6141	0.6048	0.5868	0.5735	**0.7106**	0.6951	0.4619	**0.6977**
PA	0.5937	**0.6835**	0.5916	0.5904	0.6179	0.6129	0.5295	0.5794	0.4982	0.5328

3.2 Scalability

We perform the scalability experiments of our proposed algorithm. We generate the synthetic graphs using the NetworkX package varying the number of vertices from 100,000 to 1,000,000 with the average degree fixed. The running times of our proposed algorithm for these graphs is plotted in Fig. 2. Note that it has near-linear scalability.

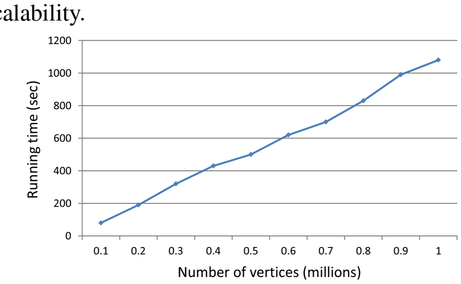

Figure 2: Near-linear scalability of LPCSP.

[1]http://snap.stanford.edu/data/

3.3 LPCSP at Work

The result of applying LPCSP on the DBLP data is shown in Fig. 3 where solid lines are actually formed links, while a dotted line implies a link not actually formed. Each circle represents a cluster, and the colors denote cluster similarity: the left two clusters are similar, while they are not similar to the third cluster. We use the CN as the baseline method to compare with LPCSP. First, note that when two vertices (e.g. Tiani Wu and Jiawei Han) belong to a same cluster which is also evolving, the proximity score of LPCSP is higher than that of the baseline method. Second, when two vertices belong to different clusters, if the cluster similarity is high (e.g. between Jiawei Han and Kyu-Young Whang), the proximity score of LPCSP is *slightly* lower than that of the baseline; however, if the cluster similarity is low (e.g. between Xiaoxin Yin and William Yurcik), the proximity score of LPCSP is *much* lower than that of the baseline.

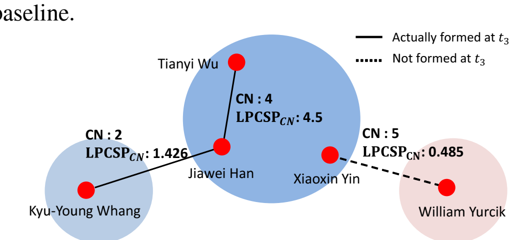

Figure 3: LPCSP on DBLP data showing two representative cases.

4. CONCLUSIONS

In this paper, we propose LPCSP, a novel link prediction method based on generalized cluster information. The main contributions are the followings.

- **Static and Temporal Cluster information**: Unlike previous methods, the LPCSP method utilizes both static and temporal cluster information to improve the quality of link prediction. LPCSP outperforms all competitors on most datasets.
- **Generality**: LPCSP is general in the sense that any link prediction method can be plugged in as a baseline algorithm.
- **Scalability**: LPCSP scales near-linearly on the edges.

Acknowledgments

Funding was provided by KAIST under project number G0413002.

5. REFERENCES

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[2] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[3] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD*, 2005.

[4] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

[5] Sucheta Soundarajan and John Hopcroft. Using community information to improve the precision of link prediction methods. In *WWW*, 2012.

[6] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *ICDM*, 2006.

[7] Jorge Carlos Valverde-Rebaza and Alneu de Andrade Lopes. Link prediction in complex networks based on cluster information. In *Advances in Artificial Intelligence-SBIA 2012*, pages 92–101. Springer, 2012.