Chapter 8
Finding Influential Nodes in Sourceforge.net Using Social Network Analysis

K.G. Srinivasa

Chaudhary Brahm Prakash Government Engineering College, India

Ganesh Chandra Deka

M. S. Ramaiah Institute of Technology, India

Krishnaraj P.M.

M. S. Ramaiah Institute of Technology, India

ABSTRACT

The contribution of volunteers in the development of Free and Open Source Software in Sourceforge.net is studied in this paper. Using Social Network analysis, the small set of developers who can maximize the information flow in the network are discovered. The propagation of top developers across past three years are also studied. The four algorithms used to find top influential developers gives almost similar results. The movement of top developers over past years years was also consistent.

INTRODUCTION

Social Network Analysis (SNA) has been an important tool for analyzing vari- ous domains of human behaviour. By bringing human interactions into graph structures they have helped gain new insights. Basic measures like betweenness, centrality, cohesion, reach etc can unveil lot of information about the modelled scenario. Concepts like 'Small World Phenomenon' are empirically verified across different cases. With the emergence of new computational tools there has been renewed interest in the applica- tion of Social Network Analysis into new domains. FOSS is primarily a global network of volunteers and thus makes for an ideal case to study using SNA Tools. The rich research available in SNA can be helpful in understanding the complex phenomenon which makes FOSS work. The present work focuses

on one aspect of SNA namely the influence maximization problem. The main objective here is to find a small set of most influential nodes in a social network so that their aggregated influence in the network is maximized. In the context of FOSS, the most influential nodes correspond to developers who are very well connected in FOSS ecology and therefore have maximum chance to propagate information in the network. Four well known algorithms namely High Degree, LDAG, SPS-CELF++ and SimPath are used to find top 5 influential developers. The propagation of top developers during past three years are also studied.

Early attempts of applying social network analysis to Free and Open Source Software phenomenon was undertaken by studying the properties such as degree distribution, diameter, cluster size and clustering coefficient. The emergence of the small world phenomenon in such networks was also studied (Xu, Christley, & Madey, 2006). The issue of finding the most influential nodes in social networks is treated in the literature as problem of centrality. In one of earliest comprehensive study on the centrality of social networks nine centrality measures were proposed. Among them three were based on degrees of points, next three on betweenness of points and last three on closeness. These different conceptions give three different views of centrality namely control, independence and activity (Freeman, 1979). The first natural greedy strategy solution which performed better than node selection heuristics based on notions of degree centrality and distance centrality continues to influence the studies in this domain (Kempe, Kleinberg & Tardos, 2003). A greedy algorithm for the target set selection problem where an initial set of nodes are selected to maximize the propagation in a social network was also proposed by same authors (Kempe, Kleinberg, & Tardos, 2005).

The later works have used variety of approaches to find top-k nodes in social networks. Finding top influential nodes using bond percolation method is tried (Kimura, Saito & Nakano, 2007). Using the metaphor of 'Outbreak Detection' the problem of informa- tion diffusion was discussed in different way. The novel solution was proposed as CELF algorithm which is reported to perform 700 times faster than simple greedy algorithm (Leskovec et al., 2007). Effective heuristics derived from the independent cascade model was comparable or better then greedy algorithms in finding top nodes for influ- ence maximization problem (Chen, Wang & Yang, 2009). Most of the studies on influence maximization assume single degree of influence across the network. In contrast, the study of topic level social influence on large networks is also undertaken (Tang et al., 2009). A novel algorithm using the concept of Shapley value from cooperative game theory is also proposed to find top influential nodes in social network (Narayanam & Narahari, 2010).

Greedy algorithm for mining top-K influential nodes which works by identifying communities within the network and then detecting the influential nodes is also reported (Wang et al., 2010). A recent attempt to improve the CELF algorithm by exploiting the property of submodularity of the spread function for influence propagation models is reported as SPS-CELF++ (Goyal, Lu, & Lakshmanan, 2011). While most solutions for influence maximization depend on the social graph structure a data based approach using historical data and avoiding the need for learning influence probabilities is also proposed by same authors (Goyal, Bonchi & Lakshmanan, 2011). They also propose another efficient algorithm SIMPATH for influence maximization under the linear threshold model (Goyal, Lu & Lakshmanan, n.d.). The first scalable influence maximization algorithm specific for the linear threshold model was proposed using LDAG algorithm (Chen, Yuan & Zhang, 2010). Identifying influential nodes using Principal Component Centrality is also proposed (Ilyas & Radha, 2011). A comprehensive survey of the development in this area is also reported (Bonchi, 2011).

9 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/finding-influential-nodes-in-sourceforgenet-using-social-network-analysis/286570

Related Content

Creating Open Source Lecture Materials: A Guide to Trends, Technologies, and Approaches in the Information Sciences
William H. Hsu (2015). *Open Source Technology: Concepts, Methodologies, Tools, and Applications (pp. 336-363).*
www.irma-international.org/chapter/creating-open-source-lecture-materials/120924

Impact of Programming Language Fragmentation on Developer Productivity: A Sourceforge Empirical Study
Jonathan L. Krein, Alexander C. MacLean, Charles D. Knutson, Daniel P. Deloreyand Dennis L. Eggett (2010). *International Journal of Open Source Software and Processes (pp. 41-61).*
www.irma-international.org/article/impact-programming-language-fragmentation-developer/44971

Free Software Philosophy and Open Source
Niklas Vainioand Tere Vadén (2012). *International Journal of Open Source Software and Processes (pp. 56-66).*
www.irma-international.org/article/free-software-philosophy-and-open-source/101218

Web and Cloud Management for Building Energy Reduction: Toward a Smart District Information Modelling
Patrizia Lombardi, Andrea Acquaviva, Enrico Macii, Anna Osello, Edoardo Pattiand Giulia Sonetti (2015). *Open Source Technology: Concepts, Methodologies, Tools, and Applications (pp. 1557-1572).*
www.irma-international.org/chapter/web-and-cloud-management-for-building-energy-reduction/120987

Dependencies, Networks, and Priorities in an Open Source Project
Juha Järvensivu (2007). *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives (pp. 116-125).*
www.irma-international.org/chapter/dependencies-networks-priorities-open-source/21183