Clustering Analysis of Motif Significance Profile in Software Networks

ZHANG LIN    QIAN GUANQUN    ZHANG LI
School of Compute Science and Engineering,
Beihang University
37# Xueyuan Rd, Beijing, CHINA
+086 82339230

Abstract: There has been considerable recent interest in network motif for understand network local features, growth and evolution mechanisms. Here, we extended the researches of network motif to the domain of software which is absent in previous researches of network categories based on motif. After compared triad significance profile of 110 java open source software packages, we found that software network could be divided into 3 clusters which are consistent with the known superfamilies from various networks. Moreover, we discuss the representative motif in software, and presented evidence that software scale may be one of the reason cause different motif SP distribution.

Keywords: Software networks, motif, significance profile, superfamily, complex network.

1. INTRODUCTION
Many of the complex networks have been shown to share global statistical features, the ‘small world’ property of short paths between any two nodes and highly clustered connections [1].

A large software system is composed with many hundreds or even thousands of smaller units (e.g. functions, classes), whose logical connections define a network in a natural way. This network reflects the internal structure of the program. Some researches show that (1) this network displays a scale-free feature, and (2) as a result of performance optimization of the program the network has a small-world structure [2]. Well designed software follows the high cohesion and low coupling rule. So(3) the topology of software is naturally hierarchical and modular [3].

It was recently found that the networks having similar global statistical features maybe have very different local structures, which is important to understand network growth and evolution mechanisms. Term ‘network motif’ is known as the basic module unit, which has much higher frequency than expected in randomized networks [4]. It may be the pattern encouraged by designers. On the other hand, the motif which has lower frequency than occurred in randomized networks may be the anti-pattern that designers try to avoid.

Ron Milo, Uri Alon, and colleagues proposed an approach to compare network local structure based on the significance profile (SP) of motif . Comparing to randomized networks with the same degree sequence, they divided various networks into four superfamilies[5]. But software networks are absent from their studies.

Our work extends the previous network superfamily studies to the domain of software networks, and tries to answer the following question: how many clusters may software networks are divided into? What’s the relationship between the software categories and superfamilies in Ron Milo and Uri Alon’s researches?

2. DETECTION OF SOFTWARE MOTIF
2.1 Method
According to approach proposed by Ron Milo and etc, the statistical significance of each subgraph i is described by the Z score :

\[ Z_i = \frac{(N_{\text{real}_i} - N_{\text{rand}_i})}{\text{std}(N_{\text{rand}_i})} \]

where Nreali is the number of times the subgraph appears in the network, and <Nrand> and std(Nrand) are the mean and standard deviation of its appearances in the randomized network ensemble.

The SP is the vector of Z scores normalized to length 1:

\[ SP_i = Z_i / (\sum Z_i^2)^{1/2} \]

The normalization emphasizes the relative significance of subgraphs, rather than the absolute significance [5].
2.2 Steps of Experiment
3-node subgraphs are the basic building blocks of local structure of complex network. In software network, the single and mutual edges in these subgraphs show the collaborative relations between software components, design patterns are good examples. Their basic topology structures are 3-node subgraphs too.

In the experiment, we choose the triads to study the SP of software networks. Experiment data come from 110 java open source software packages. Firstly, we translate the classes and dependencies between classes into network. Secondly, we use Mfinder[6] to calculate the Z score. Thirdly, normalize Z score to SP value. Finally, cluster these software networks according to correlation coefficient of SP. All experiment data could be downloaded from http://groups.google.com/group/complex-network-group-buaa .

3. ANALYSIS OF EXPERIMENT RESULT

3.1 Clustering Analysis
Fig 1 is serialized result of correlation coefficient matrix of SP. We found the software network can be roughly divided into 3 categories based on SP. Top left corner of the matrix is cluster1 (C1), Central block is cluster2 (C2), and bottom right is cluster3(C3). The proportion is about 6%, 73% and 21%.

Figure 1. The correlation coefficient matrix of the triad SP.

3.2 SP Analysis
Corresponding to the result of clustering analysis, we plot the SP value of 13 sub-graphs for each cluster in Fig2.C1, C2 and C3. Their distributions are similar with ones of superfamilies from previous research (shown in Fig2.S1, S2 and S3).

3.3 Discussion
Here are some interesting phenomena:
In most software (about 93%), Sub-graph7 is the most important motif and may presents some basic design principle. For example, when class A collaborates with class B, they usually would share some sub functions.
C1 is totally different with C2 or C3. It seems to be reverse to them.C2 and C3 both have high positive value on sub-graph7 and low negative one on sub-graph1, 2 and 3. But SP values on sub-graph 9 and 10 in C2 are much higher than in C3.
Through comparing part of the global features of each cluster (shown in Table 1), we found that the distribution of SP value is related to software scale:

<table>
<thead>
<tr>
<th>Cluster</th>
<th>size</th>
<th>&lt;N&gt;</th>
<th>&lt;L&gt;</th>
<th>&lt;CC1&gt;</th>
<th>$\alpha \times \log L/\log N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>small</td>
<td>34.2</td>
<td>66.4</td>
<td>0.089</td>
<td>0.764</td>
</tr>
<tr>
<td>C2</td>
<td>large</td>
<td>315.1</td>
<td>1129</td>
<td>0.230</td>
<td>1.172</td>
</tr>
<tr>
<td>C3</td>
<td>middle</td>
<td>71</td>
<td>251.7</td>
<td>0.208</td>
<td>1.525</td>
</tr>
</tbody>
</table>

Table 1. Global features of Clusters based on the triad SP.

In small software, SP trend to C1. Sub-graph1, 2 and 3 are encouraged. Low clustering coefficient <CC1> implies that the interaction between classes is sparse. Topology of such software is star or tree style;
With the growing of scale, SP of middle software trend to C3. Collaboration between classes increase fast, $\alpha$ and <CC1> are both doubled. Relative significance of sub-graph7 means the topology structure of such software becomes a net style;
In large software, its SP seems to trend to C2. Sub-graph7, 9 and 10 are relative significant. $\alpha$ decreases from 1.525 to 1.172. It means that when we add a new class to software, only constant number of existing nodes would interact with it. On the other hand, <CC1>of C2 is even much higher than that of C3. This may be the result of optimizing software design. How to control the growth of L and improve modularization of software maybe focus more attention of software architectures.
4. FUTURE WORK
This study of SP clustering gives us some tips for future work
(1) Why does SP similarity occur within each cluster? SP clustering offers us exciting opportunities for understanding the software growth and evolution mechanism with the help of other networks in the same superfamilies.
(2) Why does SP differ between clusters? Why do so many architecture styles or design patterns only cause a few SP clusters? Is SP related to software flexibility, program languages, design pattern and software scale?
(3) What is the relationship between local and global features of networks? How to predict each other?

5. ACKNOWLEDGMENTS
This research project was supported by the National Natural Science Foundation of China under Grant No.60773155.

6. REFERENCES


Figure 2. The triad significance profile of software networks and superfamilies. C1, C2 and C3 are from software networks. S1, S2, S3 are from superfamilies of Milo R and colleagues [5].