Method Popularity Distributions of Software Artefacts within Maven Central

How is Popularity distributed among Methods within Software Artefacts?

Overview of Sampling Process:

1. Select all artefacts between October 2021 and March 2022
2. Filter all testing-related artefacts
3. Filter artefacts based on a unique version
4. Calculate the number of dependents for an artefact
5. Perform weighted random sampling based on the number of dependents
6. Select 384 artefacts for a margin of error of 5% with a confidence level of 95%

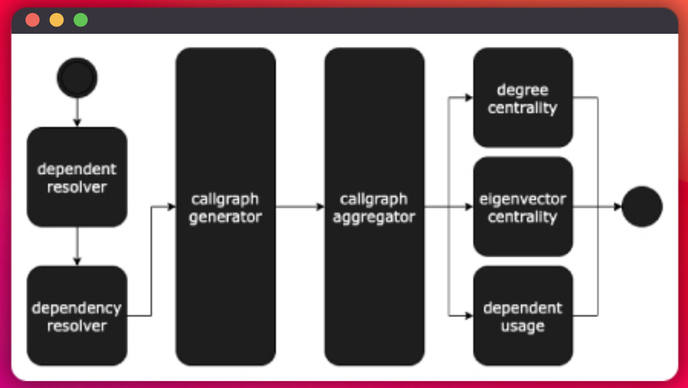

Figure 1: Application Flow of the Approach

See Figure 1 for an overview of this Section.

The we have to resolve the dependents of an artefact, and for each of them their dependents. Then we can generate callgraph of all interactions; where a callgraph is a directed graph where nodes are methods and edges are method invocations.

Metrics

- Eigenvector Centrality: a metric that considers both the number and the quality of the connections. Shortly, it also considers the importance of the invoking method.
- Degree Centrality: the number of edges that connect to a vertex, we consider only indegree as within a callgraph that is a method invocation.
- Dependent Usage Percentage: the percentage of dependents that call a given method M.

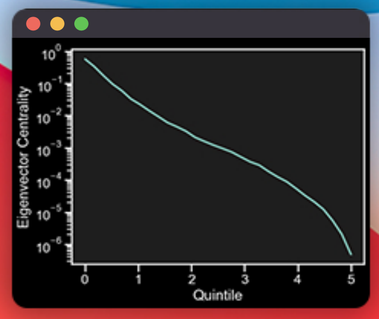

Figure 2: Popularity Distribution of Eigenvector Centrality

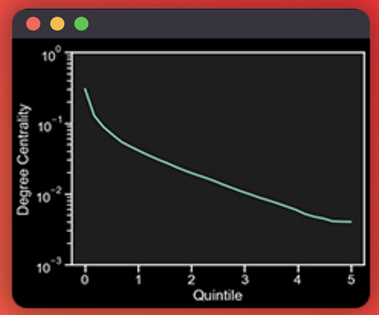

Figure 3: Popularity Distribution of Degree Centrality

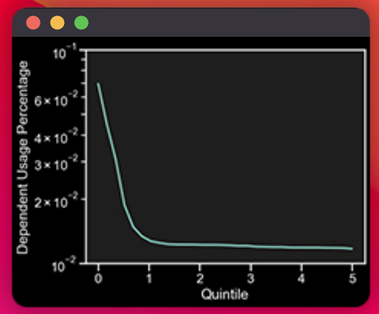

Figure 4: Popularity Distribution of Dependent Usage Percentage

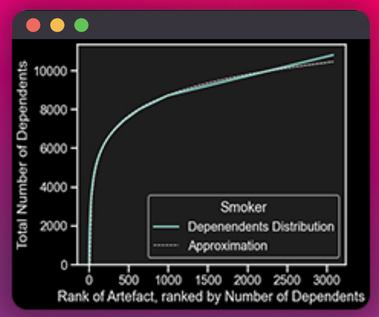

Figure 5: Power Law Distribution of Number of Dependents

We analysed the callgraphs using the three metrics and created a popularity distribution for each metric. The popularity distributions are shown in Figures 2, 3 & 4 respectfully.

- Eigenvector Centrality: a logarithmic distribution with a drop-off because of the variance of infrequently called methods.
- Degree Centrality: it follows a logarithmic distribution with more emphasis on the popular methods, because every method call influences the result equally, favouring popular methods.
- Dependent Usage Percentage: strong logarithmic distribution because the values are non-normalised; most methods are invoked in a small percentage of dependents.

Method calls and the number of dependents of an artefact both follow the Pareto Principle. 80% of method calls are to 26% of methods, see Figure 5 for the power law distribution of the number of dependents.

A popularity distribution can be integrated into a developers workflow; it can prioritise what issues to work on based on popularity of a method.

Future research opportunities include finding a different popularity metric that might be more appropriate. Also, one could analyse if any correlation exists between popularity and e.g. method signature, cyclomatic complexity etc.

We performed a method level analysis to determine a popularity distribution within software artefacts. Popularity distributions of software artefacts follow a logarithmic distribution, see Figure 2, 3 & 4.

Besides this, method calls and the number of dependents of an artefact both follow the Pareto Principle.

We proposed an approach to determine a popularity distribution of any software artefact within Maven Central. Finally, we proposed several future research opportunities regarding method level analysis.

TU Delft

e-mail: t.j.nulle@student.tudelft.nl