Summary of **Ecological Inference in Empirical Software Engineering** [1]

Akond Rahman Manish Singh Bennett Narron

August 27, 2015

1 Keywords

We identify the following keywords to be the most significant ones. Each of the keywords are accompanied with definitions.

- ii1. Varying aggregation Levels:
  A software system that has a hierarchical organization has different layers, where each higher layer is comprised of sub-layer components. The paper defines each of these higher layers as varying aggregation levels. The paper uses Eclipse as a real-life example: Eclipse consists of modules, which is a collection of packages. Each package is a collection of files. Here, each package can be labelled as an *aggregated level*, and file can be labelled as *dis-aggregated level*.

- ii2. Ecological Inference:
  Ecological inference is the empirical finding that is evident at aggregated level of software, as well as, dis-aggregated level of software. For example, in case of ecological inference, if an empirical finding that is evident at package level, which is collection of files, will also be evident at dis-aggregated levels such as files.

- ii3. Ecological Fallacy:
  Ecological fallacy is that particular empirical finding that is evident at aggregated level of software, is *not evident* at dis-aggregated level of

software. In case of ecological fallacy, if an empirical finding that is evident at package level, will not be evident at dis-aggregated levels such as files.

- ii4. Ecological Inference Risk:
  The paper defines empirical inference risk as generalizing an empirical inference that is evident at aggregated level, to a dis-aggregated level without running the same model with the factors existent at the dis-aggregated level.

2 Brief Notes

- iii1. Motivational Statement:
  Researchers have mined large scale software repositories at different levels of the software hierarchy and presented their findings. However, what remains unknown is, to what level, a hypotheses that was achieved at aggregated level, is applicable to a dis-aggregated level of the software. This paper presents a conceptual framework that investigates the *ideal* level to look for empirical findings, and whether those findings hold for both: aggregated and dis-aggregated levels of the software.

- iii2. Study Instruments:
  The paper used data extracted from the JIRA tracking system and Github repositories which contained 18 different Apache Software Foundation projects including Cassandra, Lucene, and OpenEJB.

- iii3. Statistical Tests:
  The paper uses hypotheses testing to determine if a certain hypotheses holds at aggregated level and disaggregated level. The authors state that they found a number of cases where the null hypotheses is rejected for aggregated level, which however, was not rejected at dis-aggregated level. The opposite observation was also observed and reported in the paper. z-test statistics were used in the paper to determine predictor performance.

- iii4. Future Work:
  One possible future direction that can work on top of this paper is replicating the study for other software repositories which follow different

hierarchical architectures, and are implemented in different languages. The authors have reported two levels of the hierarchy: files and packages. Another potenial extension of this study can look at the effects of looking at modules and observe if the empirical findings hold.

3 Areas of Improvement

- iv1. Other variables such as organizational effects, geographical factors, zonation, and class imbalanace can be plugged in the model and see whether the empirical findings change or not.
- iv2. The authors do not mention how they selected the specific regression-based models, and what specific models were used. One area of improvement can be of applying other software models such as linear, bi-objective, or multi-objective models.
- iv3. To quantitavely test how inferences can be different at aggregated and dis-aggreagted levels, the authors have used teh concept of diffrence in significance. Statistical signifacance has its limitations and other tests can be applied to test the difference in inferred hypotheses at two levels.

References

[1] D. Posnett, V. Filkov, and P. Devanbu. Ecological inference in empirical software engineering. In *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*, pages 362–371. IEEE Computer Society, 2011.