Review on *Ecological Inference in Empirical Software Engineering*

Akond Rahman
Department of CSC
North Carolina State University
Raleigh, NC, USA
aarahman@ncsu.edu

Manish Singh
Department of CSC
North Carolina State University
Raleigh, NC, USA
mrsingh@ncsu.edu

Bennett Narron
Department of CSC
North Carolina State University
Raleigh, NC, USA
bynarronx@ncsu.edu

ABSTRACT

Software systems are decomposed hierarchically, for example, into modules, packages and files. This hierarchical decomposition has a profound influence on evolvability, maintainability and work assignment. Hierarchical decomposition is thus clearly of central concern for empirical software engineering researchers; but it also poses a quandary. At what level do we study phenomena, such as quality, distribution, collaboration and productivity? At the level of files? packages? or modules? How does the level of study affect the truth, meaning, and relevance of the findings? In other fields it has been found that choosing the wrong level might lead to misleading or fallacious results. Choosing a proper level, for study, is thus vitally important for empirical software engineering research; but this issue hasnâĂŹt thus far been explicitly investigated. We describe the related idea of ecological inference and ecological fallacy from sociology and epidemiology, and explore its relevance to empirical software engineering; we also present some case studies, using defect and process data from 18 open source projects to illustrate the risks of modeling at an aggregation level in the context of defect prediction, as well as in hypothesis testing.

1. THESIS

Bennett: This ACM template is unusual. Paragraphs should be indenting themselves and a newline with a blank line should indicate a new paragraph. Should we choose a different one?

This paper focuses on the importance of ecological inference in software engineering and the risk of ecological fallacy as a result of mistaken ecological inference. Given that large systems consist of complex software resulting in hierarchical organization of the systems, the teams, and the processes to develop the systems, it becomes important to study the ecological inference performed at different levels of aggregation and how good the findings hold at other aggregated/disaggregated levels. The author emphasizes the importance of selecting the right level of aggregation to conduct empirical studies to measure observable outcomes such as quality and productivity. The paper discusses the various risks that accompany the ecological inference done at the aggregated levels and which result in ecological fallacy when there is discrepancy between the findings at the aggregated and disaggregated levels. The paper also discusses the various factors—sample size, zonation and class imbalance—that contribute to the risk of ecological fallacy.

1.1 Research Goals and Questions

The goal of this paper is to have a ”conceptual framework of ecological inference risk in software engineering and empirically demonstrate the existence of this risk”, by building prediction models at different aggregation levels and comparing inferences drawn from these models. While pursuing this goal,this paper tries to answer the following research questions:

1. What is the right level of study?
2. How does the level of study affect the truth, meaning and relevance of the findings?
3. Are prediction models subject to ecological inference risk?
4. Is hypothesis testing subject to ecological inference risk?
5. What are the effects of aggregation on model quality?
6. Do inferences drawn from models built at aggregated levels transfer to disaggregated levels used to build the aggregations?

1.2 Hypotheses

The authors do not clearly state a hypothesis. It is apparent that their work is of an exploratory nature, where existing ideas of ecological inference and ecological fallacy are being incorporated into a new domain, namely ESE. Through experimentation, they observe and report on trends regarding the risk of ecological fallacy in the study of statistical models of software systems, comparing results at aggregated and disaggregated levels. In order to approach this task, the authors implicitly consider the following hypotheses:

-

- ”Choosing the proper ”scale” and ”zonation” when studying phenomena that are subject to aggregation” will help mitigate the risk of ecological fallacy.

2. CONTRIBUTIONS

The authors explore the relevance of ecological inference and ecological fallacy in software engineering. The paper discusses their importance in software engineering and what are the risks in using ecological inference in software engineering. The authors further discuss the various factors that contribute to the risk of ecological fallacy—sample size, zonation and class imbalance. The authors conduct experiment , involving 18 open source projects in order to study and understand the incidence of ecological inference in these projects.
T̃hey further construct models at various aggregation levels and check if the inference derived at an aggregation level holds true for the disaggregated levels that build those levels. They find that there exists a risk of ecological fallacy if the inference is transferred from one level to another. Their findings support the claims that - “Prediction models are subject to ecological inference risk” and “Hypothesis testing is subject to ecological risk”.The paper lays out a conceptual framework of ecological risk in software engineering and concludes that ecological inference is unavoidable in software engineering research and coming up with ways to manage and mitigate the risks resulting from ecological fallacy is the way to go ahead [**?**].

3. INVESTIGATION METHODS

Bennett: Authors ”empirically study the incidence of ecological inference in 18 open source projects The authors gathered data from JIRA (an issue and defect tracking system) and associated git repositories for ”87 distinct versions of 18 different ASF (Apache Software Foundation) projects”. For each project, they cross-referenced issues on JIRA with commits related to those issues, and used the project’s git log to determine which packages and files were related to each issue. They took many variables into account, including the number of developers associated with a file or package, the amount of code, and the way in which the issue was resolved (see TABLE II).

The authors used ”an automated model selection technique to identify models”. They ranked each model using an Akaike Information Criterion (AIC) score, selecting the model with the lowest AIC score. They ignored models with a Variance Inflation Factor higher than 5 to mitigate high multicollinearity among variables. Each step within the procedure was performed to save time, while obtaining a model similar to what a researcher might choose for each revision and project.

4. RESULTS

The authors find that ”predication models are subject to ecological inference risk”.

Further, they find that hypothesis testing

4.1 "Power"

Notes on the impact of the authors’ results

4.2 Applicability

The authors’ assert that their research may be the first acknowledgement and study of ”ecological inference” and ”ecological fallacy” in Empirical Software Engineering (ESE). They support

5. TECHNICAL DEVELOPMENT

Summary of the authors’ technical development

5.1 Examples

Details of any examples to clarify technical developments

6. ACKNOWLEDGMENTS

Special thanks to Dr. Tim Menzies for his patience and co-operation.