Mining Software Engineering Data

Ahmed E. Hassan
Queen's University
Canada
ahmed@cs.queensu.ca

Tao Xie
North Carolina State University
USA
xie@csc.ncsu.edu

ABSTRACT

Software engineering data (such as code bases, execution traces, historical code changes, mailing lists, and bug databases) contains a wealth of information about a project's status, progress, and evolution. Using well-established data mining techniques, practitioners and researchers have started exploring the potential of this valuable data in order to better manage their projects and to produce higher quality software systems that are delivered on time and within budget.

This tutorial presents the latest research in mining software engineering data, discusses challenges associated with mining software engineering data, highlights success stories of mining software engineering data, and outlines future research directions. Attendees will acquire the knowledge and skills needed to integrate the mining of software engineering data in their own research or practice. This tutorial builds on several successful offerings at ICSE since 2007.

Categories and Subject Descriptors

D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement—*Restructuring, reverse engineering, and reengineering*; D.2.5 [**Software Engineering**]: Programming Environments —*Integrated environments*

General Terms

Documentation, Measurement, Reliability, Verification

Keywords

Mining software engineering data, mining software repositories

1. INTRODUCTION

Software engineering data (such as code bases, execution traces, historical code changes, mailing lists, and bug databases) contains a wealth of information about a project's progress and evolution. Many studies have emerged that use this data to support various aspects of software development within industrial and open source settings. Working with Nokia, Gall *et al.* [5] showed that exploiting software repositories can provide support to developers in changing legacy systems by pointing out hidden code dependencies. Working with Bell Labs and Avaya, Graves *et al.* [7] and Mockus *et al.* [11] demonstrated that using historical change information can support management in building reliable software systems by predicting bugs and effort. Working on open source projects, Chen *et al.* [3] showed that using historical information can assist developers in understanding large systems.

Although the idea of applying data mining techniques on software engineering data has existed since mid 1990s [10], the idea has especially attracted a large amount of interest lately within software engineering [4, 6, 8, 9, 13]. The International Working Conference on Mining Software Repositories (MSR) is recognized as the most attended co-located event at the International Conference on Software Engineering (ICSE) since 2001. More information about MSR is available at `http://msrconf.org/`.

This tutorial will provide attendees with an overview of the field of mining software engineering data, as shown in Figure 1. In particular, the tutorial will cover the following topics along three dimensions (software engineering, data mining, and future directions):

1. **Software Engineering:**
   (a) What types of software engineering data are available for mining?
   (b) Which software engineering tasks can benefit from mining software engineering data?
   (c) How are data mining techniques used in software engineering?
2. **Data Mining:**
   (a) What are the challenges in applying data mining techniques to software engineering data?
   (b) Which data mining techniques are most suitable for specific types of software engineering data?
   (c) What are the freely available data sources and data mining and analysis tools (e.g., R [1] and WEKA [2])?
3. **Future Directions:**
   What are the challenges and opportunities for the data mining and software engineering communities?

ICSE '10, May 2-8 2010, Cape Town, South Africa

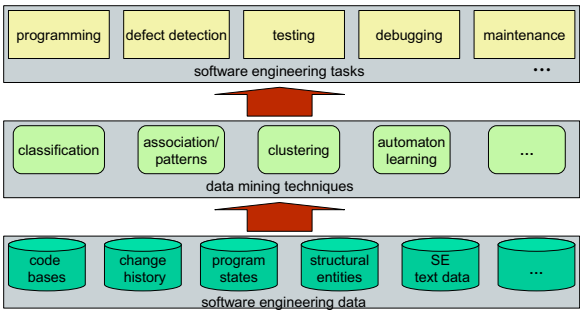

Figure 1: Overview of mining software engineering data

The tutorial will cover these topics through case studies from recent software engineering publications. Attendees will gain the knowledge needed to accomplish the following tasks:

1. Appreciate the latest advancement and success stories in the field of mining software engineering data;
2. Conduct leading-edge research in the field of mining software engineering data;
3. Apply data mining techniques on their own software engineering data using advanced data mining analysis tools and algorithms;
4. Contrast their results relative to other work within the field;
5. Recognize open problems and possible research directions within the field.

2. DETAILED OVERVIEW

The tutorial will provide a good understanding of existing research on mining software engineering data. The tutorial will categorize the existing research [12] in this field into three major perspectives: data sources being mined, tasks being assisted, and mining techniques being used. Figure 1 shows such a categorization with the bottom part as a set of software engineering data being mined, the middle part as a set of mining techniques being used, and the top part as a set of software engineering tasks being assisted. From the categorization, we intend to investigate the following four issues.

First, we intend to identify inherent challenges of mining software engineering data. We shall elaborate the essential requirements in software engineering, and analyze the differences between mining software engineering data and mining other types of scientific and engineering data. We shall discuss what types of data mining techniques are desired in software engineering, and how they should be customized to fit the requirements and characteristics of software engineering data.

Second, we intend to understand the current research and development frontier of data mining practice in software engineering. We shall summarize several types of data mining problems in software engineering that are under active investigation based on three major perspectives: data sources being mined, tasks being assisted, and mining techniques being used. Through this discussion, researchers can rapidly join this active research and gain immediate accesses to commonly available mining techniques for real problems.

Third, we intend to analyze successful cases of mining software engineering data. We shall review and demonstrate briefly several research prototypes of mining systems for software engineering data. Through the case studies, the attendees can understand how to build a test bed for research and development.

Fourth, we intend to overview commonly used data mining tools and available data sources. Our overview will help the attendees gain a better understanding of available tools. The attendees can use such tools to explore their data and integrate data mining techniques in their research and day-to-day work.

Acknowledgment

Tao Xie's work is supported in part by NSF grant CCF-0845272 and ARO grant W911NF-08-1-0443, as well as ARO grant W911NF-08-1-0105 managed by NCSU Secure Open Systems Initiative (SOSI).

3. REFERENCES

[1] The R Project for Statistical Computing. Available online at `http://www.r-project.org/`.

[2] Weka 3: Data Mining Software in Java. Available online at `http://www.cs.waikato.ac.nz/ml/weka/`.

[3] A. Chen, E. Chou, J. Wong, A. Y. Yao, Q. Zhang, S. Zhang, and A. Michail. CVSSearch: Searching through source code using CVS comments. In *Proceedings of the 17th International Conference on Software Maintenance*, pages 364–374, Florence, Italy, 2001.

[4] S. Diehl, H. Gall, and A. E. Hassan. Guest editors introduction: special issue on mining software repositories. *Empirical Software Engineering*, 14(3):257–261, 2009.

[5] H. Gall, K. Hajek, and M. Jazayeri. Detection of logical coupling based on product release history. In *Proceedings of the 14th International Conference on Software Maintenance*, pages 190–198, Bethesda, Washington D.C., Nov. 1998.

[6] M. W. Godfrey, A. E. Hassan, J. D. Herbsleb, G. C. Murphy, M. P. Robillard, P. T. Devanbu, A. Mockus, D. E. Perry, and D. Notkin. Future of mining software archives: A roundtable. *IEEE Software*, 26(1):67–70, 2009.

[7] T. L. Graves, A. F. Karr, J. S. Marron, and H. Siy. Predicting fault incidence using software change history. *IEEE Trans. Softw. Eng.*, 26(7):653–661, 2000.

[8] A. E. Hassan. The road ahead for mining software repositories. In *Proceedings of the Future of Software Maintenance at the 24th IEEE International Conference on Software Maintenance*, pages 48–57, Beijing, China, 2008.

[9] A. E. Hassan, A. Mockus, R. C. Holt, and P. M. Johnson. Guest editor's introduction: Special issue on mining software repositories. *IEEE Trans. Softw. Eng.*, 31(6):426–428, 2005.

[10] M. Mendonca and N. L. Sunderhaft. Mining software engineering data: A survey. A DACS state-of-the-art report, Data & Analysis Center for Software, Rome, NY, 1999.

[11] A. Mockus, D. M. Weiss, and P. Zhang. Understanding and predicting effort in software projects. In *Proceedings of the 25th International Conference on Software Engineering*, pages 274–284, Portland, Oregon, May 2003.

[12] T. Xie. Bibliography on mining software engineering data. `https://sites.google.com/site/asergrp/dmse`.

[13] T. Xie, S. Thummalapenta, D. Lo, and C. Liu. Data mining for software engineering. *IEEE Computer*, 42(8):35–42, August 2009.