Research Statement

David Lo
School of Information Systems, Singapore Management University
Tel: (65) 6828-0599; Email: davidlo@smu.edu.sg
Updated on 14 January 2014

Background

Software and software development activities produce a huge amount of data daily. The amount of new software code written by software companies and open source projects easily goes to millions of lines of code daily. Modern software development practices often include deployment of repositories, e.g., SVN, CVS, Git, SourceSafe, etc, which contains other forms of information aside from the code. These include information on when a piece of code is written, who is writing into what file, etc. Bug reports and bug tracking information stored in systems like Bugzilla are also widely available. These data sources covering people, processes, products, provide a rich source of information to be analyzed.

Software development itself faces many challenges. Difficulties in managing legacy systems and presence of bugs have cost billions of dollars annually. It is estimated that a substantial proportion of software cost is due to the difficulties in understanding existing/legacy systems especially during maintenance tasks, i.e. when new feature updates, bug fix, etc. are performed. US National Institute of Standards and Technology (NIST) estimated that software bugs have caused US economy to lose $59.5 billion dollars annually.

As a step forward to reduce software maintenance cost and detect bugs, machine learning and data mining techniques have been employed to mine knowledge from existing program artifacts (either from source code, execution traces, bug reports, comments, developer socio-technical network, etc). This is termed as software analytics and has been one of the new, hot topics in software engineering. The mined knowledge can be used for understanding legacy systems, reducing software maintenance cost, re-engineering legacy system, improving regression tests, aiding verification of programs, detecting bugs, etc.

Motivated by the above mentioned challenges and opportunities, application-wise, my research goal focuses on this area of software analytics; in particular, I’m interested in extending data analytics solution to transform the wealth of data available and could be collected from software and its development activities into actionable knowledge useful for software developers and other stakeholders in the software development process. Algorithm-wise, I work on improving frequent pattern mining, extending it to mine for more expressive patterns more efficiently from various data sources related to primarily, but not limited to, software engineering, and also: social network, spatio-temporal information, biology, etc.
Past Research Efforts

Most of my work could be grouped into 5 topics: mining software specifications, bug management, code search, frequent pattern mining algorithms, and social network mining. I describe these five topics in more detail in the following paragraphs. These studies were performed together with various collaborators around the globe.

Mining Software Specifications. Software specifications are often not available, incomplete, or outdated in the industry. I’m interested in reverse engineering or mining specifications from programs. I especially focus on the mining of specifications from program behaviors exhibited in systems’ execution traces. In the past, we have mined specifications in various formats ranging from: finite state machines, temporal rules, frequent usage patterns, and sequence diagrams [1-7].

Bug Management. Bugs are prevalent. We are interested in managing bugs in the various phases of its lifecycle: identification/detection, reporting, localization, and fixing. I
have been working mostly on the first 3 phases. For bug identification, we have proposed various approaches that automatically find likely bugs from programs [8,9]. For bug reporting, we have investigated the problem of duplicate bug reports and propose an approach to detect those duplicates using a combination of information retrieval and data mining approaches [10,25,26,29]. For bug localization, we have investigated various approaches that localize bugs from failure reports [11,12,30]. In addition to the above, we have also performed an empirical study on types of bugs that appear in real systems [31] and proposed an approach that can categorize bugs into types [32].

**Code Search.** Just like a regular search engine helps users in finding information that they want, a code search engine helps developers locate desired pieces of code in a code base. This would greatly help in performing maintenance tasks, e.g., finding a piece of code to be changed. We have proposed approaches that allow for dependency and basic textual search on a code base [13,27]. We are planning to extend this approach further to support more advanced queries. We have also proposed an approach that can recover similar software applications leveraging collaborative tagging [33].

**Frequent Pattern Mining.** I also work on novel pattern mining algorithms, especially sequential pattern mining. Along with co-authors, I have worked on mining sequence generators [14] and repetitive sequential patterns (closed patterns [15] and generators [16]). We also work on mining rules; different from patterns, a significant rule must have sufficient confidence. We’ve investigated non-redundant sequential rules [17] and temporal rule mining [18,19]. We are also interested in mining discriminative patterns; we have worked on mining discriminative sequential patterns [20], and dyadic sequential patterns [21]. We have applied discriminative graph mining to the problem of bug localization [12].

**Social Network Mining.** Recently, I’m also interested to mine patterns from social networks. We mine for patterns from software developer networks [22]. We also mine friendship propagation rules in social networks [23]. Furthermore, we also extract antagonistic communities from social networks [24,28,34].

For the above studies, I benefited from collaborations with co-authors from National University of Singapore, University of Illinois Urbana-Champaign, US, the Weizmann Institute of Science, Israel, Chinese University of Hong Kong, University of Milano-Bicocca, Peking University, University of Copenhagen, etc.

**Current and Future Research Efforts**

As extensions to the above studies, the following are my planned research directions:

- Application of existing mining techniques to interesting research problems in:
  - Security and intrusion detection
  - Program comprehension
  - Verification
  - Debugging
• Testing
• Re-engineering
• Further improvement to the efficiency of existing mining techniques and expressiveness of mined specifications and patterns.
• Utilization of the synergy of static and dynamic analysis in specification mining
• Investigation of new context-based automated debugging approaches
• Merging social network mining and analysis to software engineering
• Analyzing textual software engineering data
• Empirical studies in software engineering
• Construction of more research “bridges” joining the areas of data mining, information retrieval, and software engineering

Selected Publications and Research Outputs


Signatures using Discriminative Graph Mining. In proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA’09), Chicago, IL.


[15] Bolin Ding, David Lo, Jiawei Han and Siau-Cheng Khoo. Efficient Mining of Closed Repetitive Gapped Subsequences from a Sequence Database. In proceedings of the 25th International Conference on Data Engineering (ICDE'09), Shanghai, China. March 29-April 4, 2009


[19] David Lo, Bolin Ding, Lucia, and Jiawei Han. Bidirectional Mining of Non-Redundant Recurrent Rules from a Sequence Database. In proceedings of the 27th International Conference on Data Engineering (ICDE’11). Hannover, Germany. April 11-16, 2011.


[24] Kuan Zhang, David Lo, and Ee-Peng Lim. Mining Antagonistic Communities from Social Networks, in proceedings of the Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD’10), Hyderabad


[34] Zhang Kuan, David Lo, Ee-Peng Lim, and Philips K. Prasetyo. Mining Indirect Antagonistic Communities from Social Interactions, in Knowledge and Information Systems (KAIS), 2013.
