

Research Statement

David Lo

School of Information Systems, Singapore Management University

Tel: (65) 6828-0599; Email: davidlo@smu.edu.sg

Updated on 05 February 2012

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.

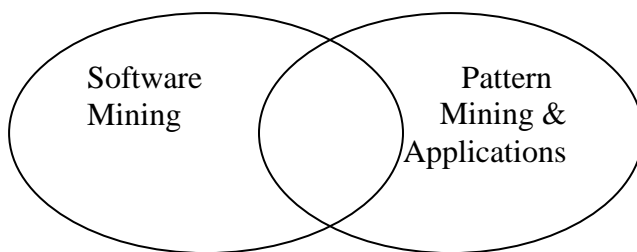


Figure 1. My Research Goals

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.

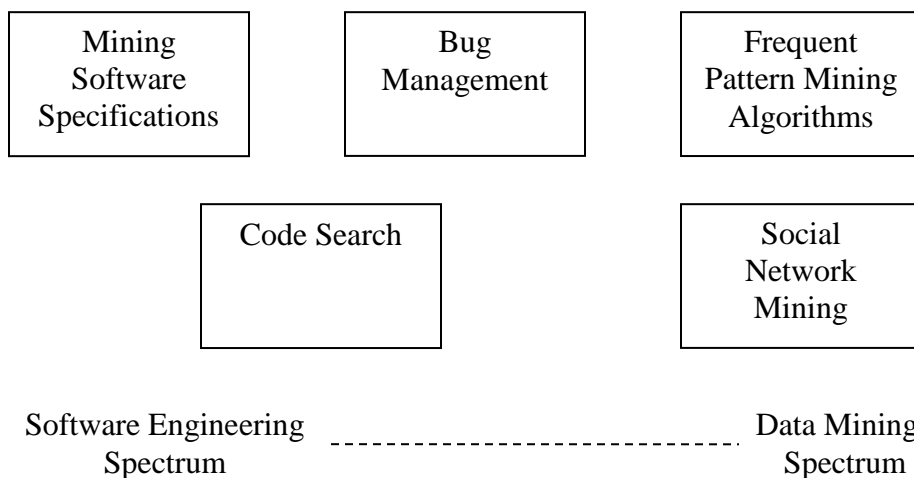


Figure 2. My Current Research Topics of Interest

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]. For bug localization, we have investigated various approaches that localize bugs from failure reports [11,12].

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 propose an approach 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.

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].

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

- [1] David Lo and Siau-Cheng Khoo. QUARK: Empirical Assessment of Automaton-based Specification Miners. In proceedings of the 13th Working Conference on Reverse Engineering (WCRE'06) . Benevento, Italy. Oct 23-27, 2006.
- [2] David Lo and Siau-Cheng Khoo. SMAR TIC: Towards Building an Accurate, Robust and Scalable Specification Miner. In proceedings of the 14th SIGSOFT Symposium on Foundation of Software Engineering (FSE'06). Portland, Oregon. Nov 5-11, 2006.
- [3] David Lo, Siau-Cheng Khoo and Chao Liu. Efficient Mining of Iterative Patterns for Software Specification Discovery. In proceedings of the 13th SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07). San Jose, California. Aug 12-15, 2007.
- [4] David Lo, Siau-Cheng Khoo, Chao Liu. Mining temporal rules for software maintenance, *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 20, no. 4, pp. 227–247, John Wiley & Sons, Inc., New York, NY, USA, 2008
- [5] David Lo, Shahar Maoz and Siau-Cheng Khoo. Mining Modal Scenario-based Specifications from Execution Traces of Reactive Systems. In proceedings of the 22nd IEEE/SIGSOFT International Conference on Automated Software Engineering (ASE'07). Atlanta, Georgia. Nov 5-9, 2007.
- [6] David Lo and Shahar Maoz. Mining Scenario-Based Triggers and Effects. In proceedings of the 23rd IEEE/SIGSOFT International Conference on Automated Software Engineering (ASE'08). L'Aquila, Italy. September 15-19, 2008.
- [7] David Lo and Shahar Maoz. Scenario-based and value-based specification mining: better together, in proceedings of the 25th IEEE/ACM International Conference on Automated Software Engineering (ASE'10). Antwerp, Belgium. September 20-24, 2010.
- [8] Julia L. Lawall and David Lo. An automated approach for finding variable-constant pairing bugs, in proceedings of the 25th IEEE/ACM International Conference on Automated Software Engineering (ASE'10). Antwerp, Belgium. September 20-24, 2010.
- [9] David Lo, Ganesan Ramalingam, Venkatesh-Prasad Ranganath, and Kapil Vaswani. Mining Quantified Temporal Rules: Formalism, Algorithms, and Evaluation, in *Science of Computer Programming (SCP)*, 2011 (to appear).
- [10] Chengnian Sun, David Lo, Xiaoyin Wang, Jing Jiang, and Siau-Cheng Khoo. A Discriminative Model Approach for Accurate Duplicate Bug Report Retrieval, in proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE'10), Cape Town, South Africa
- [11] Hong Cheng, David Lo, Yang Zhou, Xiaoyin Wang, and Xifeng Yan. Identifying Bug Signatures using Discriminative Graph Mining. In proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'09), Chicago, IL

- [12] Lucia, David Lo, Lingxiao Jiang, and Aditya Budi. Comprehensive Evaluation of Association Measures for Fault-Localization, in proceedings of the 26th IEEE International Conference on Software Maintenance (ICSM'10). Timisoara, Romania. September 12-18, 2010.
- [13] Xiaoyin Wang, David Lo, Jiefeng Cheng, Lu Zhang, Hong Mei, and Jeffrey Xu Yu. Matching dependence-related queries in the system dependence graph, in proceedings of the 25th IEEE/ACM International Conference on Automated Software Engineering (ASE'10). Antwerp, Belgium. September 20-24, 2010.
- [14] David Lo, Siau-Cheng Khoo and Jinyan Li. Mining and Ranking Generators of Sequential Patterns. In proceedings of the 8th SIAM International Conference on Data Mining (SDM'08). Atlanta, USA. April 24-26, 2008.
- [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
- [16] David Lo, Jinyan Li, Limsoon Wong, and Siau-Cheng Khoo. Mining Iterative Generators and Representative Rules for Software Specification Discovery. IEEE Transaction on Knowledge and Data Engineering, Feb 2011.
- [17] David Lo, Siau-Cheng Khoo, and Limsoon Wong. Non-Redundant Sequential Rules - Theory and Algorithms, Information Systems, vol. 34, no. 4-5, pp. 438-453, Elsevier, 2009
- [18] David Lo, Siau-Cheng Khoo and Chao Liu. Efficient Mining of Recurrent Rules from a Sequence Database. In proceedings of the 13rd International Conference on Database Systems for Advance Applications (DASFAA'08). New Delhi, India. March 19-21, 2008.
- [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.
- [20] David Lo, Hong Cheng, Jiawei Han, Siau-Cheng Khoo, and Chengnian Sun. Classification of Software Behaviors for Failure Detection: A Discriminative Pattern Mining Approach. In Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'09), Paris, France. June 28-July 1, 2009
- [21] David Lo, Hong Cheng and Lucia, "Mining Closed Discriminative Dyadic Sequential Patterns", Proceedings of the 2011 International Conference on Extending Data Base Technology (EDBT 11). Uppsala, Sweden, Mar. 2011.
- [22] Didi Surian, David Lo, and Ee-Peng Lim. Mining Collaboration Patterns from a Large Developer Network, in proceedings of the 17th IEEE Working Conference on Reverse Engineering (WCRE'10) (Short Paper). Boston, USA. October 13-16, 2010.
- [23] Cane Wing-ki Leung, Ee-Peng Lim, David Lo and Jianshu Weng. Mining Interesting Link Formation Rules in Social Networks, in Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010). Toronto, Canada. October 26-30, 2010.
- [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
- [25] Chengnian Sun, David Lo, Siau-Cheng Khoo, and Jing Jiang. Towards More Accurate Retrieval of Duplicate Bug Reports, in proceedings of the 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011), Lawrence, USA.
- [26] Tian Yuan, Chengnian Sun, and David Lo. Improved Duplicate Bug Report Identification, to appear in Proceedings of 15th European Conference on Software Maintenance and Reengineering (CSMR 2012), ERA Track, Szeged, Hungary

- [27] Shaowei Wang, David Lo, and Lingxiao Jiang. Code Search via Topic-Enriched Dependency Graph Matching, to appear in Proceedings of the 18th IEEE Working Conference on Reverse Engineering (WCRE 2011), Limerick, Ireland
- [28] David Lo, Didi Surian, Zhang Kuan, and Ee Peng Lim. Mining Direct Antagonistic Communities in Explicit Trust Network, to appear in Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM 2011), Glasgow, United Kingdom