Learning from Large Codebases

Author(s):
Raychev, Veselin

Publication Date:
2016

Permanent Link:

https://doi.org/10.3929/ethz-a-010737712 →

DISS. ETH NO. 23746

LEARNING FROM LARGE CODEBASES

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES OF ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

VESELIN RAYCHEV
Master in Informatics
Sofia University "St. Kliment Ohridski"
born on 23.04.1984
citizen of Bulgaria

accepted on the recommendation of

Prof. Dr. Martin Vechev
Prof. Dr. Eran Yahav
Prof. Dr. Armando Solar-Lezama
Prof. Dr. Charles Sutton

2016

ABSTRACT

As the size of publicly available codebases has grown dramatically in recent years, so has the interest in developing programming tools that solve software tasks by learning from these codebases. Yet, the problem of learning from programs has turned out to be harder than expected and thus, up to now, there has been little progress in terms of practical tools that benefit from the availability of these massive datasets.

This dissertation focuses on addressing this problem: we present new techniques that learn probabilistic models from large datasets of programs as well as new tools based on these probabilistic models which improve software development.

The thesis presents three new software systems (JSNice, Slang and DeepSyn) that learn from large datasets of programs and provide likely solutions to previously unsolved programming tasks including deobfuscation, static type prediction for dynamic languages, and code synthesis. All three of these systems were trained on thousands of open source projects and answer real-world queries in seconds and with high precision. One of these systems, JSNice, was publicly released and is already widely used in the JavaScript community.

An important ingredient of the thesis is leveraging static analysis techniques to extract semantic representations of programs and building powerful probabilistic models over these semantics (e.g., conditional random fields). Working at the semantic level also allows us to enforce important constraints on the predictions (e.g. typechecking). The net result is that our tools make predictions with better precision than approaches whose models are learned directly over program syntax.

Finally, the dissertation presents a new framework for addressing the problem of program synthesis with noise. Using this framework, we show how to construct programming-by-example (PBE) engines that handle incorrect examples, and introduce a new learning approach based on approximate empirical risk minimization. Based on the framework, we developed a new code synthesis system (DeepSyn) which generalizes prior work and provides state-of-the-art precision.

ZUSAMMENFASSUNG

So wie die Größe der öffentlich zugänglichen Codebasen in den letzten Jahren dramatisch zugenommen hat, so hat auch das Interesse an der Entwicklung von Programmier-Tools zugenommen, die Software-Probleme lösen indem sie von diesen Codebasen lernen. Doch das Problem des Lernens von Programmen hat sich als schwieriger als erwartet herausgestellt und bisher hat es wenig Fortschritt bei praktischen Tools gegeben, die von massiven Datenmengen profitieren. Die vorliegende Arbeit konzentriert sich auf die Lösung dieses Problems: Wir präsentieren neue Techniken, die Wahrscheinlichkeitsmodelle von großen Datensätzen von Programmen lernen, sowie neue Tools, die Software-Entwicklung verbessern.

Diese Doktorarbeit präsentiert drei neue Software-Systeme (JSNice, Slang und DeepSyn), die von großen Datenmengen von Programmen lernen und Lösungen für bisher ungelöste Programmierprobleme bieten, unter anderem Deobfuscation, statische Typinferenz für dynamische Sprachen und Programmsynthese. Alle drei dieser Systeme wurden mit Tausenden von Open-Source-Projekten trainiert und beantworten reale Abfragen in Sekunden und mit hoher Präzision. Eines dieser Systeme, JSNice wurde veröffentlicht und in großem Umfang in der JavaScript-Community verwendet.

Ein wichtiger Bestandteil der Arbeit ist die Verwendung von Techniken der statischen Analyse zur Extraktion semantischer Repräsentationen von Programmen und der Erzeugung von mächtigen Wahrscheinlichkeitsmodellen anhand dieser Semantiken (z.B. Conditional Random Fields). Auf semantischer Ebene zu arbeiten erlaubt es auch wichtige Einschränkungen auf die Vorhersagen zu erzwingen (z.B. Typkorrektheit). Das Endergebnis ist, dass unsere Tools Vorhesagen mit einer höheren Präzision machen als Ansätze, deren Modelle direkt von Programmsyntax lernen.

Schließlich stellt die Dissertation einen neuen Framework für die Behandlung des Problems der Programmsynthese mit Rauschen vor. Mit diesem Framework zeigen wir, wie man Programming-by-Example-Systeme (PBE) konstruiert, die falsche Beispiele verstehen. Wir führen einen neuen Lernansatz basierend auf approximierter empirischer Risi-

kominimierung (ERM) ein. Basierend auf dem Framework haben wir ein neues Programmsynthesesystem (DeepSyn) entwickelt, welches vorherige Resultate verallgemeinert und Präzision auf dem Stand der Technik bietet.