Analysing Programming Data: An example study, and the wider problems

Neil Brown

King’s College London
Blackbox data set

• BlueJ is a Java IDE aimed at beginners
• Blackbox collects data from all opted-in BlueJ users about their code and compilations
  – (Ask us to get access)
• Big data set:
  – 300 million compilations
  – ~3TB (compressed) of source code
Novice Java Programming Mistakes: Large-Scale Data vs Educator Beliefs

Neil Brown and Amjad Altadmri

Motivation: Educator Folk Wisdom

• “Students always mix up equality (==) and assignment (=)”
• “Students always miss out the semi-colons”
• “We all know that students...”

• Are they right?
• Knowing misconceptions matters: correcting them is an important part of teaching
Research Plan

• Ask educators to make prediction
• Use Blackbox data to prove them wrong

• Example mistakes:
  – Mismatched brackets
  – Use of = instead of ==
  – Use of == instead of .equals()
  – Invoking methods with wrong types
MISTAKE FREQUENCY VS TIME-TO-FIX

Which mistakes are most frequent?
Which mistakes take longest to fix?
How to measure mistake severity
Measure of error severity: Total time-to-fix

Mean Time-To-Fix (seconds)

Mistake frequency (millions)

Slow to fix

High Frequency
== for string comparison (180 Ms)

Wrong types in method call (230 Ms)

Mismatched brackets (250 Ms)
MISTAKE FREQUENCY (PROPORTIONS) VS USER “LIFETIME”

Which mistakes do students learn to avoid making?
MISTAKE TIME-TO-FIX VS USER “LIFETIME”

Which mistakes do users get better at fixing?
EDUCATOR OPINION VS BLACKBOX

Do educators predict Blackbox mistake frequency?
Is this affected by experience?
Educators accurate => points vertically close to line

Experience matters => horizontal trend
Educators accurate => points vertically close to line

Experience matters => horizontal trend
EDUCATOR “ACCORD” VS EXPERIENCE

Is educators’ predictive accuracy affected by experience?
Results: accord vs experience
Results: accord vs experience
Results: accord vs experience
Results: accord vs experience
Summary

- Little evidence of learning (except for brackets)
- Educators don’t match each other, or data
- Experience has no effect
Analysing program data
Big data

• Overall Blackbox dataset is Big Enough™
• But once you start slicing, it quickly gets small
• Trade-offs: context versus size
• Programming epidemiology?
• Evidence alignment
Example data

- DEMO
  - 998008
  - 999607
Question/analysis disconnect

• “What problems do students have with static typing?”
• AKA: “Write me a program that takes a temporally ordered list of source code file snapshots and prints out a category of static typing issue.”
Tools

• Basic tools: a lexer and parser (if you only want to analyse syntactically correct code)

• Higher-level feature detection in program code is rare
  – And often we only look for one very specific thing

• We don’t know how to analyse temporal changes in code (esp. given how noisy they are)
Discussion

• How do we analyse this?
• How do we develop an analysis community?