Modeling Bug Report Quality

Pieter Hooimeijer
Westley R. Weimer
University of Virginia
Motivation

- Software maintenance is costly

- Large software projects lack resources to with every bug report effectively

- Some reports slip through the cracks
Outline

• The Firefox data set

• A descriptive model of bug reports
  − features of interest
  − training and cross validation

• Experiments
  − model instance selection
The Data

• 27,984 bug reports for Mozilla Firefox

Bugzilla
New bug from a user with canconfirm or a product without UNCONFIRMED state

UNCONFIRMED

Bug confirmed or receives enough votes

Bug is reopened, was never confirmed

NEW

Developer takes possession

Ownership is changed

ASSIGNED

Developer takes possession

Development is finished with bug

RESOLVED

Developer takes possession

Development is finished with bug

Bug is closed

Possible resolutions:
FIXED
DUPLICATE
WONTFIX
WORKSFORME
INVALID

Issue is resolved
The (Linear) Model

• Goal—classify bug reports:
  'good/cheap' vs. 'bad/expensive'

• 'Good' ≈ Moves from New to Resolved before a certain resolution cutoff

• What is a good cutoff?
Independent Variables

- At-submission:
  - Self-reported severity
  - Readability measures
  - System load
  - Submitter reputation

- Post-submission:
  - Severity Changes
  - Comment Count
  - Attachment Count

- Eight decision deadlines: {1 hour, ..., 10 days}
Training and Cross Validation

1) Pick some *resolution cutoff* and some decision deadline

2) *Cross validation* (10 times):
   - Train model, continuous in the range \([0, 1]\)
     \(~0 = \text{'bad'} \text{ vs. } \sim 1 = \text{'good'}~\)
   - Find the *model cutoff* in the range \([0; 1]\) that maximizes classification accuracy

3) Train the model on the entire data set
Model Instance Selection

• Treat model as an information retrieval system:

\[
\text{precision} = \frac{|C \cap R|}{|R|} \quad \text{recall} = \frac{|C \cap R|}{|C|}
\]

\[
F_\alpha = \frac{(1 + \alpha)pr}{\alpha p + r}
\]

*R* is the model's output; *C* is the correct output
Results

Precision

Recall

F_1-Score

Results
Results

Precision

Recall

F₁-Score

Decision made after (days)

Resolution Cutoff (days)

0.775

0.755

0.765

0.77

0.73
Results

Cumulative

Most Recent Snapshot Only

Performance

Decision made after (days)

---

Precision

Recall

F-Score
Results

• **Decision Deadline:** need about a day's worth of post-submission data

• **Resolution Cutoff:** little change relative to degenerate model

• **Important Features (ANOVA):**
  - Severity
  - Coleman-Liau
  - System Work Load
  - # Comments (60 min.)
  - # Comments (12 hrs.)
  - # Attachments (1 day)
  - # Attachments (6 days)
Conclusion

• We used a simple linear regression model for ease of analysis

• **Early features** mattered; were able to make better-than-chance predictions

• **Might achieve cost savings in certain settings**
Cross Validation

The Universe of Bug Reports

Firefox
Cross Validation

The Universe of Bug Reports
Cross Validation

The Universe of Bug Reports
Cross Validation
Evaluation: A Cost Model

• Idea: measure the relative 'cost' of using our model as a filter

• Two symbolic costs:
  
  * Triage  * Miss

• A 'miss' = Filtered out a 'good' bug report and all of its duplicates
Cost Model Result

- Cost of triaging every report = 27,984 Triage
- Our model achieves cost savings if: Miss < 50 Triage