

Activity Mining in Open Source Software

**Daniel Mack, Nitesh
Chawla, Greg Madey**

University of Notre
Dame



Outline

- Methods
- Activity
- Mining SourceForge
- Results
- Conclusions and Questions

Open Source: “Wild West”

- Software Engineering vs. Open Source
 - Group dynamics
 - Communication
- Group and social analysis
- Where to look?

“You insist that there is something a machine cannot do. If you will tell me precisely what is that a machine cannot do, then I can always make a machine which will do just that.” J. von Newman

SourceForge

- Cheap and Easy
 - Registration
 - Administration
 - CVS
 - Forums, Surveys
- Problems
 - Too Cheap
 - Too Easy

Data data everywhere, not a thought to think

- Separate the imitators from the real groups
- What about projects that are active?
 - What is active?
- Great, so how do we identify and define activity?
 - Right Questions

The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' (I found it!) but 'That's funny ...' Isaac Asimov

Methods: Exploration

- Associations
 - A-priori
 - Look for relationships among metrics
- M5 Rules
 - Regression Tree
 - Insight into attributes

The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' (I found it!) but 'That's funny ...' Isaac Asimov

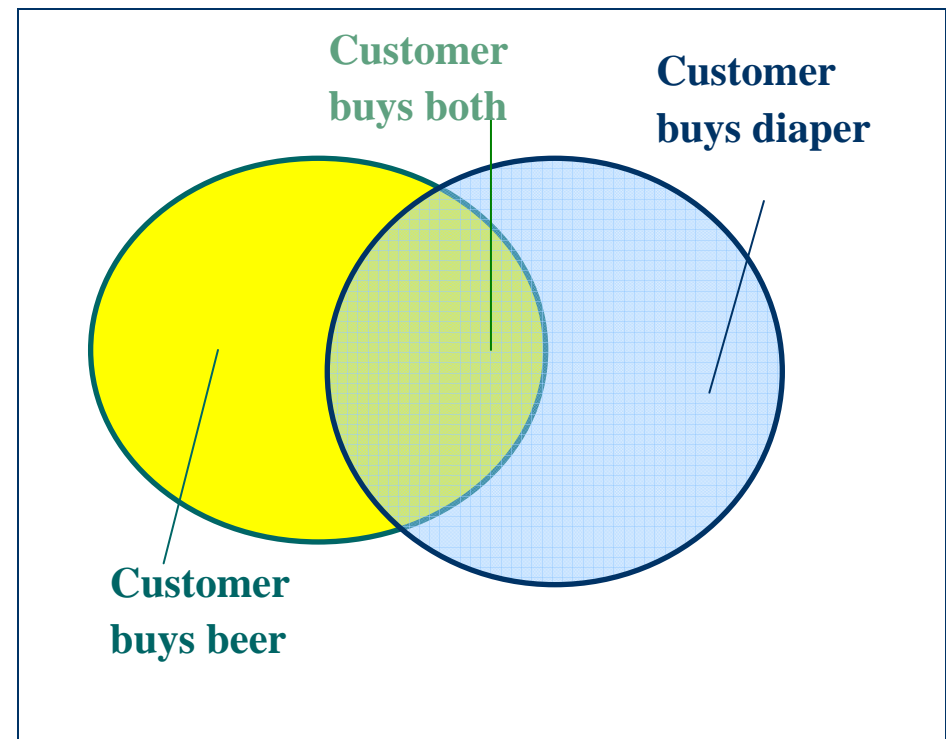
Association Rules

- Uncover *associations* between two or more attributes (identify the affinity among attributes)
- **Association rules:** Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.

*“I would rather discover one causal relation than be a King of Persia,”
Democritus*

Some relevant terms

- *support* s for a particular association rule $A \Rightarrow B$ is the proportion of transactions D that contain both A and B
 - *support* = number of transactions containing both A and B / total number of transactions
- *confidence* c for a particular association rule $A \Rightarrow B$ is percentage of transactions in D containing A that also contain B
 - *confidence* = $P(B|A)$ = number of transactions containing both A and B / number of transactions containing A



M5 Regression Rules

- Rules constructed from “regression trees”
- Differences to classification decision trees:
 - **Splitting criterion**: minimizing intra-subset variation
 - **Pruning criterion**: based on numeric error measure
 - Leaf node predicts average class values of training instances reaching that node
- Can approximate piecewise constant functions
 - Easy to interpret
 - More sophisticated version: **model trees**

Building the tree

- Splitting criterion: standard deviation reduction

$$SDR = sd(T) - \sum_i \frac{|T_i|}{|T|} \times sd(T_i)$$

- Termination criteria (important when building trees for numeric prediction):
 - Standard deviation becomes smaller than certain fraction of sd for full training set (e.g. 5%)
 - Too few instances remain (e.g. less than four)

Smoothing predictions

- Naïve method for prediction outputs value of LR for corresponding leaf node
- Performance can be improved by smoothing predictions using internal LR models
 - Predicted value is weighted average of LR models along path from root to leaf

KDD in SourceForge

- Acquisition
 - Data Dump
- Preprocessing
 - Understanding 150 tables
 - Focus on Games ~3800 tables
 - “SQL Hell” <- *Daniel*
- Models
- Analysis



**“Mining needle in a haystack.
So much hay and so little time”**

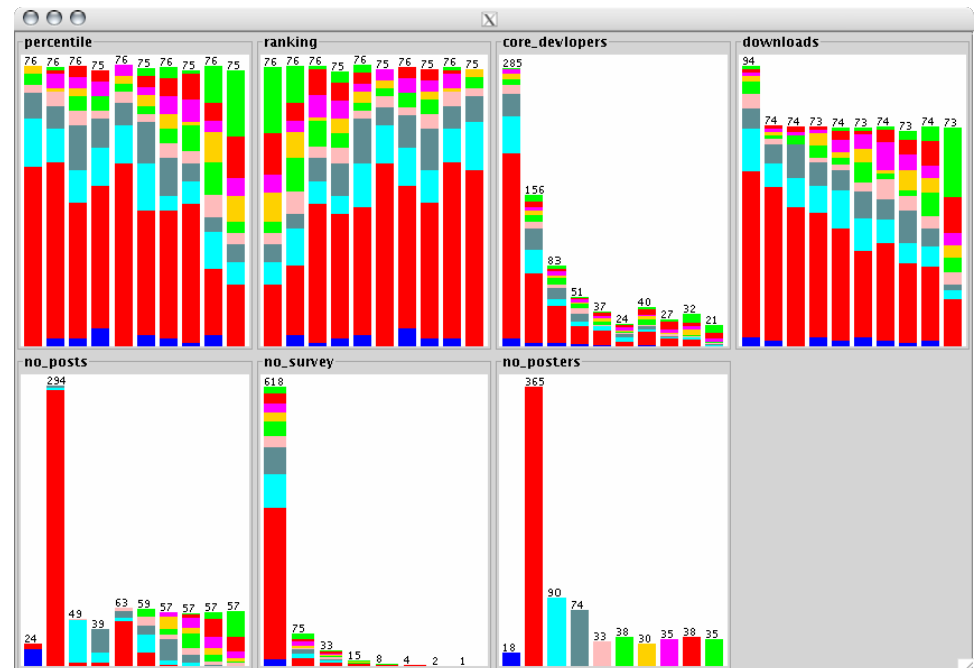
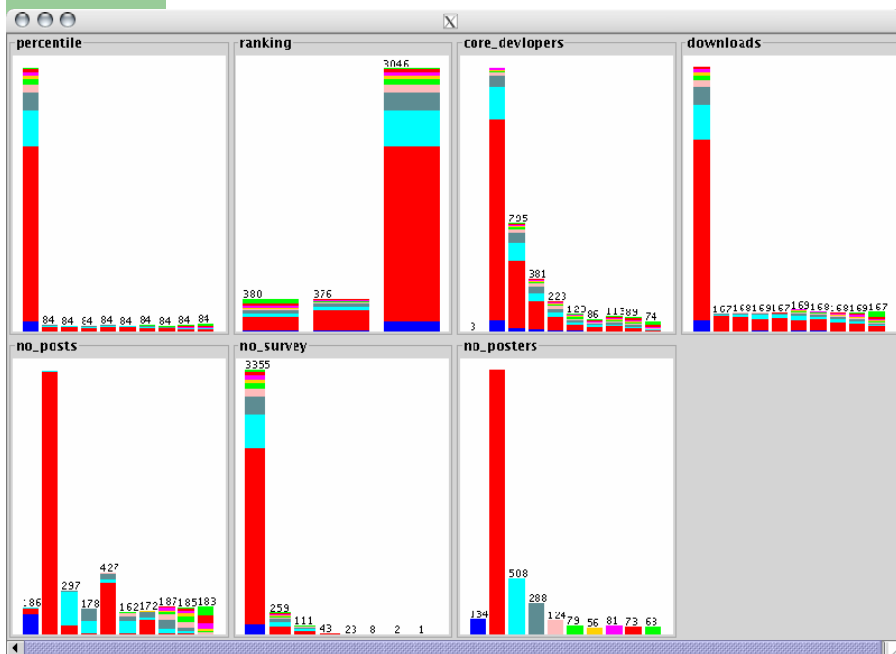
Data attributes considered

- percentile
- ranking
- core developers
- # of downloads
- # of posts
- # of surveys
- # of posters

Associations

- First Round
 - Imbalanced data
 - Lots of trivial associations
 - Remove Unranked~756 examples left
- Second Round
 - 24 Rules
 - Posters and Posts
- Third Round
 - 8 Rules
 - Activity

Data visualization



Nitesh Chawla

University of Notre Dame

Assoc. Rules

Best rules found:

1. `core_developers=(-inf-1.5]` `no_posters=(0.5-1.5]` 189 \implies `no_survey=(-inf-0.5]` 177
conf:(0.94)
2. `no_posters=(0.5-1.5]` 365 \implies `no_survey=(-inf-0.5]` 334 conf:(0.92)
3. `downloads=(-inf-0.5]` 94 \implies `no_survey=(-inf-0.5]` 85 conf:(0.9)
4. `core_developers=(-inf-1.5]` 285 \implies `no_survey=(-inf-0.5]` 256 conf:(0.9)
5. `core_developers=(1.5-2.5]` 156 \implies `no_survey=(-inf-0.5]` 132 conf:(0.85)
6. `core_developers=(-inf-1.5]` `no_survey=(-inf-0.5]` 256 \implies `no_posters=(0.5-1.5]` 177
conf:(0.69)
7. `core_developers=(-inf-1.5]` 285 \implies `no_posters=(0.5-1.5]` 189 conf:(0.66)
8. `core_developers=(-inf-1.5]` 285 \implies `no_survey=(-inf-0.5]` `no_posters=(0.5-1.5]` 177
conf:(0.62)

M5 Rules

- Chose three attributes
 - Downloads, core developers, posters
- Splitting attributes
 - Values they split on
 - Which values?

Rules for downloads

Rule: 1

IF

no_posters <= 5.5
core_developers <= 2.5

THEN

downloads =
+ 2611.1884 [398/45.996%]

Rule: 2

IF

no_posters <= 11.5
core_developers <= 6.5
no_posts <= 29.5

THEN

downloads =
+ 4212.8614 [166/29.405%]

Rule: 3

IF

no_posters <= 22.5

THEN

downloads =
+ 15037.6755 [151/62.846%]

Rule: 4

IF

core_developers <= 16.5

THEN

downloads =
+ 54154.8125 [32/62.09%]

Rule: 5

downloads =
+ 331176.4444 [9/105.433%]

M5 Rules for core_developers

Rule: 1

IF no_posters <= 5.5
 percentile <= 47.086

THEN

core_developers =
 + 2.1167 [300/62.713%]

Rule: 2

IF no_posters <= 9.5
 no_posts <= 7.5
 percentile <= 82.764

THEN

core_developers =
 + 2.4972 [181/54.361%]

Rule: 3

IF no_posters <= 11.5
 percentile <= 91.618
 no_posts <= 45.5

THEN

core_developers =
 + 3.7967 [123/61.244%]

Rule: 4

core_developers =
 + 9.2434 [152/153.894%]

Conclusion

- Important elements
 - Right questions
 - Data framework and preprocessing
 - Modeling algorithms
- Break the activity into none, low and high
 - Decompose and optimize
- SourceForge is an exciting repository to explore hierarchies, activities, and even be able to identify characteristics of successful projects