MTAT.03.183: Data Mining

Data Mining of Software Repositories

Dietmar Pfahl
email: dietmar.pfahl@ut.ee

Spring 2017
About me

• Assoc. Prof. at UT (Software Engineering)
• Adjunct Prof. at University of Calgary, Canada (since 2005)
  • Senior Member of ACM & IEEE
  • Certified SCRUM Product Owner
• Group Leader & Department Head at Fraunhofer Inst. of Experimental SW Engineering (1996-2005)
• Siemens Corporate Research (1987-1995)
Acknowledgement

• The following persons contributed to the lecture slides:
  – Ezequiel Scott
  – Riivo Kikas
  – Didar Al-Alam
  – Faiz Shah
Data Mining of SW Repositories – Why and What?
Data Mining of SW Repositories – Why and What?

• To support decision making at all stages of the software development process

• To complement other sources of evidence
  – Surveys, Case Studies, Experiments
Context:
Evidence-Based SE

- Knowledge in SE: Anecdotal vs. Evidence-based
- Evidence in Science -> Data
- Data Sources?
  - Surveys, Case Studies, Experiments, Project Repos,
    Dedicated collections:
    [http://promise.site.uottawa.ca/SERespository/datasets-page.html](http://promise.site.uottawa.ca/SERespository/datasets-page.html)

- Tip: Link to Lecture by Gregory Wilson:
  [https://vimeo.com/9270320](https://vimeo.com/9270320)
Research Questions – Taxonomy

Exploratory Question

Knowledge Question

Base-Rate Question

Research Question

Causality Question

Simple Causality Question

Causality-Comparative Question

Existence Question

Description and Classification Question

Descriptive Comparative Question

Frequency and Distribution Question

Descriptive-Process Question

Relationship Question

Causality-Comparative Interaction Question
Exploratory Questions

- **Existence questions** -> Does X exist?
  - Example: *Do issue reports actually exist?*

- **Description and classification questions** -> What is X like? / What are its properties? / How can it be categorized? / How can we measure it? / What is its purpose? / What are its components? / How do the components relate to each other?
  - Example: *What are all the types of issue reports?*

- **Descriptive comparative questions** -> How does X differ from Y?
  - Example: *How do issue report formats differ between open source and closed source development projects?*
Knowledge and Design Questions

- **Knowledge Questions**: focusing on the way the world is
  - Questions about the normal pattern of occurrence of a phenomenon (**Base-rate Questions**)
  - Questions about relationships between two different phenomena (**Relationship Questions**)
  - Questions about causality between two phenomena (**Causality Questions**)

- **Design Questions**: concerned with how to do things better
Knowledge Questions

• Base-rate:
  – **Frequency and Distribution Questions** -> How often does X occur? / What is an average amount of X?
    Example: *How many distinct issue reports per issue report type are created in large software development projects?*
  – **Descriptive-Process Questions** -> How does X normally work? / What is the process by which X happens? / In what sequence do the events of X occur?
    Example: *How do software developers use issue reports?*
Knowledge Questions (cont’d)

• Relationship:
  – Relationship Questions -> Are X and Y related? / Do occurrences of X correlate with occurrences of Y?

Example: Do project managers’ claims about how often their teams use test tool X correlate with the actual use of test tool X?
Knowledge Questions (cont’d)

• Causality:
  – Simple Causality Questions -> Does X cause Y? / Does X prevent Y? / What causes Y? / What are all the factors that cause Y? / What effect does X have on Y?
    Example: *Does the use of GUI test tool X improve software quality?*
  – Causality-Comparative Questions -> Does X cause more Y than does Z? / Is X better at preventing Y than Z?
    Example: *Does the use of GUI test tool X improve software quality more than other GUI test tools?*
  – Causality-Comparative Interaction Questions
Knowledge Questions (cont’d)

• Causality:
  – Causality-Comparative Interaction Questions -> Does X or Z cause more Y under one condition but not others?

Example: Does the use of GUI test tool X improve software quality more than GUI test tools in web application projects, but not in genuine mobile applications?
Design Questions

-> ”What is an effective way to achieve X?” / What strategies help to achieve X?”

Examples:

What is an effective way for teams to test mobile applications in order to improve quality without increasing cost?

or

What is an effective way for teams to design mobile applications in order to improve energy efficiency?
The Wallace Model

Theories

Theory Construction
Logic (induction)

Logical Inference
(deduction)

Empirical Generalizations
(Laws)

Hypotheses
(Research Questions)

Data Analysis,
Parameter Estimation

Research Design

Research Methods

Observations

Data Collection & Research Methods

- Survey
  - Questionnaire-based (primary study)
  - Literature-based (secondary / tertiary study)

- Case Study
  - Descriptive
  - Exploratory
  - Confirmatory

- Experiment
  - Controlled Experiment
  - Quasi-Experiment
  - Longitudinal studies

- Many other …
  - Action Research
  - Ethnography
  - Design Science
Survey Research
Survey – Characterisation

- A survey is a data collection method or tool used to gather information about individuals in order to identify the characteristics of a broad population.
- The defining characteristic is the selection of a representative sample from a well-defined population with the aim to generalise from the sample to the population.
- Usually conducted with questionnaires, but can also involve structured interviews or data logging techniques.
- Example:
  - Investigate to what extent, how, by which companies, and by whom within the companies, TDD is used.
Survey – Characterisation (cont'd)

When to use it?

– Either at start of research to get an understanding of the current situation …
– or at the end of a research phase to see the impact/acceptance/etc. of a new method/technique/tool

Issues:

– 'Superficial' --> no explanation / no causality --> not suitable for hypothesis testing
– 'Generalisability' of results depends on the choice of population and 'response rate', as well as validity and reliability of the data collection instrument
Survey – Example

What?

Research Questions:
- How is Agile practiced at Microsoft?
  - i.e. What do engineers do?
- How do engineers feel about it?
  - i.e. Do they like it?

Who, Where, and When?

Microsoft (worldwide, 2006)
Anonymous survey sent to 2821 engineers

- 10% random sampling of all developers, testers, program managers at Microsoft in October 2006

487 valid responses
- 44% developers, 28% testers, 17% program managers

Why?

Many agile approaches exist – what's in it for Microsoft?

Survey – Example (cont'd)

Agile practice penetration at Microsoft

- Team coding standards
- Continuous integration of code
- System metaphor
- Simple design
- Sustainable pace
- User stories
- Small releases
- Direct interaction with customer
- Design improvement
- Collective code ownership
- Acceptance testing
- Whole team daily stand-up meeting
- Test-driven development
- Pair programming

0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

- Yes
- Sometimes
- Planning to
- No
- Never will
Survey – Example (cont'd)

Quantitative Results (Highlights)
- 33% of respondents (spread across divisions) report their team uses Agile methodologies.
- They mainly use Scrum (68%).
- Used for many legacy products.
- Agile usage does not appear to depend on team co-location.
- Test-driven development and pair programming are not very common.

Qualitative Results (Highlights)
- MS engineers who have used Agile like it for their local team, but not necessarily for their organization.
- They worry about scale, overhead, and management buy-in.

<table>
<thead>
<tr>
<th>Perceived benefits (687 comments, 44 themes)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Improved Communication and Coordination</td>
<td>121</td>
</tr>
<tr>
<td>2. Quick Releases</td>
<td>101</td>
</tr>
<tr>
<td>3. Flexibility of Design – Quicker Response</td>
<td>86</td>
</tr>
<tr>
<td>to Changes</td>
<td></td>
</tr>
<tr>
<td>4. More Reasonable Process</td>
<td>65</td>
</tr>
<tr>
<td>5. Increased Quality</td>
<td>62</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Perceived problems (565 comments, 58 themes)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Does not scale to larger projects</td>
<td>52</td>
</tr>
<tr>
<td>2. Too many meetings</td>
<td>44</td>
</tr>
<tr>
<td>3. Management buy-in</td>
<td>37</td>
</tr>
<tr>
<td>4. Unfamiliar with Agile</td>
<td>36</td>
</tr>
<tr>
<td>5. Coordination with other teams</td>
<td>29</td>
</tr>
<tr>
<td>6. Losing sight of the big picture</td>
<td>29</td>
</tr>
</tbody>
</table>
Controlled Experiment – Characterisation

- An investigation of a testable hypothesis where one or more independent variables are manipulated to measure their effect on one or more dependent variables.
- In Software Engineering, typically, experiments require human subjects to perform some task.
Controlled Experiment – Simple Example

- Independent Variable: Tool used (Levels: X and Y)
- Dependent Variable: Design Quality
- Treatments: E = use the new Tool X / C = use the old Tool Y

NB: Design can be *within-subject* or *between-subject*
Randomization is a prerequisite for a controlled experiment!

Controlled Experiment vs. Quasi-Experiment

- **Randomized or true experiment**
  - **is random assignment used?**
    - yes
    - **is there a control group or multiple measures?**
      - yes
      - quasi-experiment
      - no
      - non-experiment
Experiment – Example

What?

Research Question:
- What is best – Pair Programming or Solo Programming?

Why?

Many studies with contradicting results – mostly conducted with students (not with professional developers)

Who, Where, and When?

Norway, 2007

295 junior, intermediate and senior professional Java consultants from 29 companies were paid to participate (one work day)

99 individuals; 98 pairs

The pairs and individuals performed the same Java maintenance tasks on either:
- a "simple" system (centralized control style), or
- a "complex" system (delegated control style)

They measured:
- duration (elapsed time)
- effort (cost)
- quality (correctness) of their solutions

Experiment: Overall Effect of PP

Total Effect of PP

- Duration: -8%
- Effort: 84%
- Correctness: 7%
Experiment: Effect of PP for Juniors

Effect of PP for Juniors

<table>
<thead>
<tr>
<th>Duration</th>
<th>Effort</th>
<th>Correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 %</td>
<td>111 %</td>
<td>73 %</td>
</tr>
</tbody>
</table>
Experiment – Example (cont'd)

Effect of PP for Seniors

-9 %
83 %
-8 %
-40 %
-20 %
0 %
20 %
40 %
60 %
80 %
100 %
120 %
140 %
160 %

Duration     Effort     Correctness

Difference from individuals

Duration
Effort
Correctness

-40 %
Experiment: Effect of PP for Juniors taking task complexity under consideration

Moderating Effect of System Complexity for Juniors

<table>
<thead>
<tr>
<th></th>
<th>CC (easy)</th>
<th>DC (complex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>4%</td>
<td>6%</td>
</tr>
<tr>
<td>Effort</td>
<td>109%</td>
<td>112%</td>
</tr>
<tr>
<td>Correctness</td>
<td>32%</td>
<td>149%</td>
</tr>
</tbody>
</table>
Experiment: Effect of PP for Seniors taking task complexity under consideration

Moderating Effect of System Complexity for Seniors

- Duration: -23 %  8 %
- Effort: 115 %  55 %
- Correctness: -13 %  -2 %

Experiment: Effect of PP for Seniors taking task complexity under consideration
So, when should we use PP?

<table>
<thead>
<tr>
<th>Programmer Expertise</th>
<th>Task Complexity</th>
<th>Use PP?</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junior</td>
<td>Easy</td>
<td>Yes</td>
<td>Provided that increased quality is the main goal</td>
</tr>
<tr>
<td></td>
<td>Complex</td>
<td>Yes</td>
<td>Provided that increased quality is the main goal</td>
</tr>
<tr>
<td>Intermediate</td>
<td>Easy</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Complex</td>
<td>Yes</td>
<td>Provided that increased quality is the main goal</td>
</tr>
<tr>
<td>Expert</td>
<td>Easy</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Complex</td>
<td>No</td>
<td>Unless you are sure that the task is too complex to be solved satisfactorily even by solo seniors</td>
</tr>
</tbody>
</table>

The question of whether PP is best, or not, is meaningless!
One should ask: In which situation is PP best to achieve a defined goal?

Importance of Context:
Helps construct/refine theory about when and how to do 'Pair Programming'
Case Study Research
Case Study – Characterisation

- Definition:
  - An empirical enquiry that investigates a contemporary phenomenon within its real-life context (in-vivo=in the living), especially when the boundaries between phenomenon and context are not clearly evident.

- Examples:
  - Investigation on how a company takes advantage of ‘Open Innovation’
  - Investigation on how a company practices mobile app testing
  - Investigation on how and why a company practices TDD

- Characteristics:
  - When to use? --> When 'rich' information is requested
  - Often focus on qualitative data --> allows for better understanding of conditions under which a technique/tool works

- Issues:
  - Important: Proper case selection / clearly stated research question(s) / clearly defined framework for interpreting the observations
  - 'Generalisability' (1 case --> only 1 context)
Case Study – Variants

- **Descriptive Case Study**
  - Purely observational / Focus on “What happens?”

- **Explorative Case Study**
  - Initial investigation of some phenomena to derive new hypotheses and build theories / Focus on “What and Why?”

- **Confirmatory Case Study**
  - Start out with a given theory and try to refute it, ideally with a series of case studies covering various contexts

More on Case Study design (SE Group at Lund University):
http://serg.cs.lth.se/education/case_study_research/
Case Study – Guidelines

• Research questions
• Case and subject selection
• Data collection procedures
• Data Analysis procedures
  – E.g., coding schemes

• Results:
  – Case and subjects description, covering execution, analysis and interpretation issues
  – Evaluation of validity

From events to observations to perceptions to conclusions
Validity & Reliability of Empirical Studies

• **Construct Validity**
  – Concepts being studied are operationalised and measured correctly (do the measures used actually represent the concepts you want to measure?)

• **Internal Validity**
  – Establish a causal relationship and sort out spurious relationships (exclude confounding variables / by: random sampling, blocking, balancing)

• **Conclusion Validity**
  – Do proper statistical inference

• **External Validity**
  – Establish the domain to which a study’s findings can be generalized (precisely describe the population and experimental conditions)

• **Reliability**
  – The study can be repeated (i.e., by other researchers) and yields the same results
  – The measurement instrument is reliable (interrater agreement)
Data Mining of SW Repositories – Why and What?

• To support decision making at all stages of the software development process

• To complement other sources of evidence
  – Surveys, Case Studies, Experiments
Data Mining in SW Engineering: Application Examples

• Journal: EMSE’16
  http://www.springer.com/computer/swe/journal/10664

• Conferences:
  – MSR’16: http://thomas-zimmermann.com/2016/01/msr-2016/
  – ESEM’16: http://alarcos.esi.uclm.es/eseiw2016/esem
  – EASE’16: http://ease2016.lero.ie
  – PROMISE’16: http://promisedata.org/2016/
# Data Mining in SW Engineering (2016)

<table>
<thead>
<tr>
<th></th>
<th>EMSE’16</th>
<th>MSR’16</th>
<th>ESEM’16</th>
<th>EASE’16</th>
<th>PROMISE’16</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>RE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Arch/Des</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Progr/Evol</td>
<td>11</td>
<td>15</td>
<td>4</td>
<td></td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>Testing/QA</td>
<td>3</td>
<td>6</td>
<td></td>
<td></td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>ProjM</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>21</td>
</tr>
<tr>
<td>ProdM</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>QM</td>
<td>12</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>36</strong></td>
<td><strong>43</strong></td>
<td><strong>20</strong></td>
<td><strong>6</strong></td>
<td><strong>8</strong></td>
<td><strong>113</strong></td>
</tr>
</tbody>
</table>

![Pie chart showing distribution of topics](chart.png)
SE Data Repositories

- App stores (Google Play, etc.)
- Q/A web-pages (e.g., StackOverflow)
- Crash report repositories (e.g., Ubuntu’s repository)
- YouTube tutorials (e.g., tool tutorials)
- ELFF dataset at Brunel: https://github.com/tjshippey/ESEM2016
- Data Showcases at MSR’16
- Industry data: ISBSG repository, Finnish dataset
- Issue Trackers -> e.g. JIRA
- Version Control Systems -> e.g., Git
GitHub & GHTorrent

- https://en.wikipedia.org/wiki/GitHub
- API’s for Java, Ruby, Python, etc.

Articles:

“The GHTorrent Dataset and Tool Suite” (2013)

“Lean GHTorrent: GitHub data on demand“ (2014)
GHTorrent

Data scheme
## GHTorrent

<table>
<thead>
<tr>
<th>Entity</th>
<th>Description</th>
<th>Raw data entity</th>
<th>Num Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>__</td>
<td></td>
<td>__</td>
<td>__</td>
</tr>
<tr>
<td>projects</td>
<td>Project repositories.</td>
<td>repos</td>
<td>__</td>
</tr>
<tr>
<td>users</td>
<td>Github users.</td>
<td>users</td>
<td>793,855</td>
</tr>
<tr>
<td>project_members</td>
<td>Users with commit access to the referenced project.</td>
<td>repo_collabs</td>
<td>983,629</td>
</tr>
<tr>
<td>organization_members</td>
<td>List of members in an organization.</td>
<td>org_members</td>
<td>34,924</td>
</tr>
<tr>
<td>commits</td>
<td>A list of all commits on Github. The project_id field refers to the first project this commit has been added to.</td>
<td>commits</td>
<td>29,978,291</td>
</tr>
<tr>
<td>project_commits</td>
<td>List of all commits to a project.</td>
<td>commit_comments</td>
<td>126,697</td>
</tr>
<tr>
<td>commit_parents</td>
<td>Commits that are parents to a commit.</td>
<td>watchers</td>
<td>7,744,619</td>
</tr>
<tr>
<td>commit_comments</td>
<td>Code review comments for a commit.</td>
<td>followers</td>
<td>1,797,343</td>
</tr>
<tr>
<td>watchers</td>
<td>Users that have starred (was watched) a project</td>
<td>issues</td>
<td>2,326,069</td>
</tr>
<tr>
<td>followers</td>
<td>Users that are following another user.</td>
<td>issue_events</td>
<td>4,085,294</td>
</tr>
<tr>
<td>issues</td>
<td>Issues that have been recorded for a project</td>
<td>issue_comments</td>
<td>2,886,006</td>
</tr>
<tr>
<td>issue_events</td>
<td>Chronologically ordered list of events on an issue.</td>
<td>pull_requests</td>
<td>1,144,251</td>
</tr>
<tr>
<td>issue_comments</td>
<td>Discussion comments on an issue.</td>
<td>pullrequest_comments</td>
<td>2,228,894</td>
</tr>
<tr>
<td>pull_requests</td>
<td>List of pull requests for base_repo. Requests originate at head head_repo/commit and are created by user_id</td>
<td>__</td>
<td>__</td>
</tr>
<tr>
<td>pull_request_comments</td>
<td>Discussion comments on a pull_request</td>
<td>__</td>
<td>__</td>
</tr>
<tr>
<td>pull_request_history</td>
<td>Chronologically ordered list of events on a pull_request</td>
<td>__</td>
<td>__</td>
</tr>
</tbody>
</table>

Fig. 1. Schema entities, their description, the corresponding raw data entities and the number of raw data items (Feb 15, 2013).
GHTorrent – Data-on-demand Service

Figure 1: Architecture of the GHTorrent data-on-demand service.
GHTorrent – Database Dumps

What am I downloading?

- The MySQL dump is a full, up to date database dump. You can use it for querying the latest available data.
- The MongoDB dumps are incremental. They are provided mostly for reference and backup purposes, as they may contain duplicates. The reasons for this are the following:
  - When refreshing project data, old records are deleted and new are added. This cannot be reflected in the dumps (it is not practical to regenerate all dumps every time).
  - The dumps have already been restored once, hence the dump dates do not represent the actual data generation dates.

For these reasons, we recommend using the MongoDB data through our query service.

Which is the applicable license?

See here

MySQL database dumps

New: As of MySQL dump `mysql-2015-09-25`, we are distributing CSV files (one file per table) instead of `mysqldump` based backups. The provided archive expands to a directory including a restore script and instructions on how to do the restore. See more information here.

You can also query MySQL. It is always loaded with the latest dump.

- 2017-05-01 (55047 MB)
- 2017-04-01 (33516 MB)
- 2017-03-01 (50466 MB)
- ...

http://ghtorrent.org/downloads.html
GHTorrent – DB Dumps: Limitations

- Dumps contain only the first order dependencies
  - e.g., contributors to a repository and their followers, but not followers of these followers
- Creating the dumps can be a lengthy process, potentially requiring several days to complete
- No recovery actions in case of errors are currently implemented, potentially leading to incomplete dumps
  - e.g., if GitHub fails to answer an API request
- Requests to lean GHTorrent should not exceed 1000 repositories
  - This is to limit the load on GHTorrent servers
Getting data from GitHub repositories using the Java API

(by Ezequiel Scott and Didar Al-Alam)
Example: GitHubDataExtractor

We can use the GitHubDataExtractor project to retrieve data from Github repositories.

- The project relies on the Github API for Java
- You can download the GitHubDataExtractor from here
  - import the project into your favorite Java IDE (e.g. Eclipse) and then
  - add the required libraries to the build path

Links at: https://courses.cs.ut.ee/2017/dm/spring/Main/Links
What data can be extracted?

- Commits
- Pull requests
- Issues...
About the project

• There are two important classes:
  – **RRCalc** – just the main class
  – **CommitDataCollection** – the class in charge of collecting the commit data, it does the hard job

• In **RRCalc**, we set up important data such as the username, repository, the credentials, dates, etc.

• In **CommitDataCollection**, we use the Github API to connect with the Github services and obtain data from the repository
How does it work?

First, we have to create an object for the repository and set the credentials up:

```java
RepositoryService repservice = new RepositoryService();
repservice.getClient()
    .setCredentials(GitCredits[0], GitCredits[1]);

RepositoryId repo = new RepositoryId(repoOwner, repoName);
```
How does it work? (cont.)

Then, we can use different services for retrieving the data from the repository.

There are three services available: **Commit**, **Issue**, and **Pull**. All of them require credentials.

```java
// For downloading commits
CommitService commitservice =
    new CommitService();
commitservice.getClient()
    .setCredentials(GitCredits[0], GitCredits[1]);

// For downloading pull requests
PullRequestService pullservice =
    new PullRequestService();
pullservice.getClient()
    .setCredentials(GitCredits[0], GitCredits[1]);
```
How does it work? (cont.)

Finally, we can retrieve all the data from each service and store it in List objects. It makes finding elements easier to do.

// For downloading commits
List<RepositoryCommit> commitList = commitservice.getCommits(repo);

// For downloading issues
List<RepositoryIssue> issueList = issueservice.getIssues();

// For downloading pulls
List<PullRequest> pullList = pullservice.getPullRequests(repo, "closed");
How does it work? (cont.)

Once we have obtained the lists with the data, we can retrieve all the info from the commit/issue/pull objects.

```java
// Getting the SHA key from the i-commit
String sha = commitList.get(i).getSha();

// Getting the author from the i-commit
String author = commitList.get(i).getCommit().getAuthor().getName();

// Getting the message from the i-commit
String message = commitList.get(i).getCommit().getMessage();
```

...
Mining Software Repositories: Application Examples
Application Examples – Overview

- Ex1 – Release Readiness – RAISE 2016 (PhD)
- Ex2 – Issue RT (a) – MSR 2016 (PhD)
- Ex3 – Issue RT (b) – EASE 2016 (MSc)
- Ex4 – App Reviews – WAMA 2016 (MSc/PhD)
- More (ongoing PhDs):
  - Green Software
  - Open Innovation (RE)
  - ...
- Many MSc thesis topics
Application Example 1

- RAISE 2016

- Comparative Analysis of Predictive Techniques for Release Readiness Classification

- Slides: Didar Al-Alam
Application Example 2

- MSR 2016

- Using Dynamic and Contextual Features to Predict Issue Lifetime in GitHub Projects

- Slides: Riivo Kikas
Application Example 3

• EASE 2016

• Improving Expert Prediction of Issue Resolution Time
Predicting Issue Resolution Time – Why & How?

Why?

• Maintenance/Evolution is consuming a major share of the development effort

• Knowing the probable issue resolution time helps in the planning of resource allocation

How?

• Manually done by experts

• Automatically done by models
Motivation of Study

• Many attempts have been made to predict issue resolution time
  • Published work shows mixed results with regards to performance

• Availability of a case Company:
  • Expert estimates
  • Plan and actual data available
  • Question: Would automatic prediction outperform experts?
Related Work

• Little industry data available regarding expert estimates

• Several studies on automatic prediction (> 2006):
  • Usually using OSS data with actual IR times
  • Several methods used:
    • kNN, $\alpha$-kNN, (simple) k-means clustering, Naïve Bayes Classifier, C4.5 Decision Tree, Random Forest, and Logistic Regression
  • Different performance measures used:
    • MMRE, Pred_rel(25%), classification accuracy, AUC
  • High variation in performance / Unclear whether experts are outperformed

10 studies found
Research Goals

(1) To compare the prediction quality of expert-based IRT prediction in a software company in Estonia with that of various fully automated IRT prediction approaches proposed/used by other researchers

- including k-means clustering, k-nearest neighbor classification, Naïve Bayes classification, decision trees, random forest (RF) and ordered logistic regression (OLR)

(2) To improve the current IRT prediction quality in the company at hand

IRT = Issue Resolution Time
Approach

• Establish baseline (expert data in Company)
• Apply automatic prediction methods found in the literature to Company data
• Apply enhanced versions of the found prediction methods to Company data
• Compare results (using 4 performance measures)
Company Baseline

Dataset:
- IRs must be written in English
- IRs must be 'closed'
- IRs must have both 'estimated' and 'actual' resolution times

2125 IRs in total
894 IRs used

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Char</td>
<td>Text</td>
</tr>
<tr>
<td>Description</td>
<td>Char</td>
<td>Text</td>
</tr>
<tr>
<td>Reporter</td>
<td>ENUM</td>
<td>70 different values</td>
</tr>
<tr>
<td>Project name</td>
<td>ENUM</td>
<td>11 different values</td>
</tr>
<tr>
<td>Type</td>
<td>ENUM</td>
<td>Bug, Epic, Gw-issue, Improvement, Incident,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Investigation, New Feature, Project, Story,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sub-task, Task, Technical task</td>
</tr>
<tr>
<td>Priority</td>
<td>ENUM</td>
<td>Blocker, Critical, High, Immediate, Low, Normal</td>
</tr>
<tr>
<td>Creation date</td>
<td>Integer</td>
<td>Continuous values: April 2011 – January 2015</td>
</tr>
<tr>
<td>Label</td>
<td>List of strings</td>
<td>39 different values</td>
</tr>
<tr>
<td>Actual IRT</td>
<td>Integer (seconds)</td>
<td>Recorded actual time spent</td>
</tr>
<tr>
<td>Estimated IRT (by experts)</td>
<td>Integer (seconds)</td>
<td>Estimated time required for resolving the issue</td>
</tr>
</tbody>
</table>
Company Baseline

• Experts’ performance: predicted versus actual

![Diagram showing distribution of actual and predicted resolution time (RT)]

- Number of issues in interval according to estimate (black)
- Number of issues in interval actually (gray)

Intervals in days (8 hours):
- [0, 0.5] - (0.5, 1] - (1, 3] - (3, 6] - (6, 11] – (20, 40] - (40, ...)

Classes of Resolution Time

Number of issues in interval according to estimate (black)
Number of issues in interval actually (gray)
Company Baseline

- Experts’ performance

<table>
<thead>
<tr>
<th></th>
<th>Pred_abs (0.5h)</th>
<th>Pred_abs (1h)</th>
<th>Pred_rel (10%)</th>
<th>Pred_rel (25%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>0.668</td>
<td>0.727</td>
<td>0.501</td>
<td>0.578</td>
</tr>
</tbody>
</table>

\[
difference(irt_p, irt_a) = \begin{cases} 
0, & \text{if } irt_a \in [irt_{p_{min}}, irt_{p_{max}}] \\
\min(\{|irt_{p_{min}} - irt_a|, |irt_{p_{max}} - irt_a|\}), & \text{otherwise}
\end{cases}
\]

where \( irt_p \) is the predicted interval of IRT and \( irt_a \) is the actual IRT in hours.

Magnitude of Relative Error (MRE) is defined as the division of MAE by the actual IRT: \( MRE = \frac{MAE}{irt_a} \).

Predictive Quality using MAE can now be calculated as the percentage of issues with MAE ≤ X as follows:

\[
Pred_{abs}(X) = \frac{|MAE(d_i) \leq X|}{|D|}
\]

X is a threshold for MAE and D is the set of issue reports \( d_i \in D \).

Similarly, Predictive Quality using MRE can now be calculated as the percentage of issues with MRE ≤ X as follows:

\[
Pred_{rel}(X) = \frac{|MRE(d_i) \leq X|}{|D|}
\]

where X is a threshold for MRE and D is again the set of issue reports \( d_i \in D \).
Automatic Prediction

• Using methods as published
  - Naïve Bayes
  - C4.5 Decision Tree
  - Random Forest
  - RF (enhanced)
  - Ordered Logistic Regression
  - OLR (enhanced)
  - $\alpha$-kNN ($k=3; \alpha=0.3$)
  - Spherical k-means (Last 50; Title)

• Using enhanced methods
  - Outlier removal
  - Advanced k-means
Automatic Prediction (as published)

- Using methods as published
  - Naïve Bayes
  - C4.5 Decision Tree
  - Random Forest
  - RF (enhanced)
  - Ordered Logistic Regression
  - OLR (enhanced)
  - $\alpha$-kNN ($k=3; \alpha=0.3$)
  - Spherical k-means (Last 50; Title)

- Using enhanced methods
  - Outlier removal
  - Advanced k-means
Automatic Prediction (enhanced)

• Using methods as published

<table>
<thead>
<tr>
<th>Method</th>
<th>Naïve Bayes</th>
<th>C4.5 Decision Tree</th>
<th>Random Forest</th>
<th>RF (enhanced)</th>
<th>Ordered Logistic Regression</th>
<th>OLR (enhanced)</th>
<th>α-kNN (k=3; α=0.3)</th>
<th>Spherical k-means (Last 50; Title)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.138</td>
<td>0.171</td>
<td>0.121</td>
<td>0.138</td>
<td>0.561</td>
<td>0.605</td>
<td>0.429</td>
<td>0.512</td>
</tr>
<tr>
<td></td>
<td>0.460</td>
<td>0.566</td>
<td>0.378</td>
<td>0.439</td>
<td>0.579</td>
<td>0.676</td>
<td>0.457</td>
<td>0.541</td>
</tr>
<tr>
<td></td>
<td>0.533</td>
<td>0.643</td>
<td>0.439</td>
<td>0.512</td>
<td>0.579</td>
<td>0.676</td>
<td>0.302</td>
<td>0.387</td>
</tr>
<tr>
<td></td>
<td>0.572</td>
<td>0.666</td>
<td>0.449</td>
<td>0.535</td>
<td>0.579</td>
<td>0.676</td>
<td>0.541</td>
<td>0.656</td>
</tr>
</tbody>
</table>

• Using enhanced methods
  • Outlier removal
  • Advanced k-means
Comparison: Expert vs. Model

<table>
<thead>
<tr>
<th>Prediction Technique</th>
<th>Pred_abs (0.5h)</th>
<th>Pred_abs (1h)</th>
<th>Pred_rel (10%)</th>
<th>Pred_rel (25%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert estimation</td>
<td>0.668</td>
<td>0.727</td>
<td>0.501</td>
<td>0.578</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>0.138</td>
<td>0.171</td>
<td>0.121</td>
<td>0.138</td>
</tr>
<tr>
<td>C4.5 Decision Tree</td>
<td>0.460</td>
<td>0.566</td>
<td>0.378</td>
<td>0.439</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.533</td>
<td>0.643</td>
<td>0.439</td>
<td>0.512</td>
</tr>
<tr>
<td>RF (enhanced)</td>
<td>0.572</td>
<td>0.666</td>
<td>0.449</td>
<td>0.535</td>
</tr>
<tr>
<td>Ordered Logistic Regression</td>
<td>0.561</td>
<td>0.665</td>
<td>0.429</td>
<td>0.512</td>
</tr>
<tr>
<td>OLR (enhanced)</td>
<td>0.579</td>
<td>0.676</td>
<td>0.457</td>
<td>0.541</td>
</tr>
<tr>
<td>α-kNN (k=3; α=0.3)</td>
<td>0.409</td>
<td>0.520</td>
<td>0.302</td>
<td>0.387</td>
</tr>
<tr>
<td>Spherical k-means (Last 50; Title)</td>
<td>0.643</td>
<td><strong>0.732</strong></td>
<td><strong>0.603</strong></td>
<td><strong>0.656</strong></td>
</tr>
</tbody>
</table>
Results Summary

• RQ 1: Comparison Company vs. Published Models
  • Experts outperform published models

• RQ 2: Enhance Company’s Performance
  • Spherical k-means applied to Title only and with using only last 50 reported issues is for 3 out of 4 performance measures (slightly) better than experts
Discussion

The good news:

• Automatic prediction is roughly as good as experts and thus might be used instead of them

The interesting news:

• Experts and models might complement each other
Limitations – Threats to Validity

• External validity
  • Only one case with a relatively small data set

• Internal validity
  • The fact that the case company was recording plan/actual expert data might mean that they are relatively mature in this particular aspect (i.e., estimating IRT) and thus the comparison with automatic methods might be unfair

• Conclusion validity
  • Choice of performance measure
Application Example 4

- WAMA 2016
- Feature-Based Evaluation of Competing Apps
- Slides: Faiz Ali Shah
Motivation

- User feedback could help developers improve the quality of their app by comparing it with other similar apps.

More precisely:

- To identify sets of app features loved by users in other apps but missing in company’s own app.

- To identify app features which are perceived negatively by its users and need improvement.
App Reviews Dataset

- We used app reviews dataset openly available on the website of Swinburne University of Technology.

![Bar Chart](https://example.com/bar-chart.png)

**Figure 1. Number of reviews in each app**

Approach

Figure 2. Overview of the approach
Tool Prototype: Show List of Apps and Select Base App

<table>
<thead>
<tr>
<th>Application (click to find competitors)</th>
<th>Link to iTunes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period Diary</td>
<td><a href="http://itunes.apple.com/us/app/id436762566">http://itunes.apple.com/us/app/id436762566</a></td>
</tr>
</tbody>
</table>
Feature list of base app “Calorie Counter” with minimum support count = 22
Tool Prototype: Present competing Apps

Competing apps based on selected features of base app “Calorie Counter”

<table>
<thead>
<tr>
<th>App Name</th>
<th>Features Matched</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nike+ Running</td>
<td>5/5</td>
</tr>
<tr>
<td>Nike+ Training Club</td>
<td>5/5</td>
</tr>
<tr>
<td>Map My Fitness</td>
<td>5/5</td>
</tr>
<tr>
<td>Calorie Counter (Base app)</td>
<td>5/5</td>
</tr>
<tr>
<td>Strava Running and Cycling</td>
<td>4/5</td>
</tr>
<tr>
<td>Runkeeper</td>
<td>4/5</td>
</tr>
<tr>
<td>Map My Ride</td>
<td>3/5</td>
</tr>
</tbody>
</table>

Base app selected features:
- track calorie
- calorie counter
- track weight
- workout tracker
- exercise activity
Tool Prototype: Evaluation of Competing Apps

Base app selected features
- track calorie
- calorie counter
- track weight
- workout tracker
- exercise activity

Feature Categorization
By sentiment score
- [0.5,2.5] -> Positive
- [-0.5,-2.5] -> Negative
- Otherwise -> Neutral
Result 1:
Feature-based comparison of the base app “Calorie Counter” with competing app “Map My Fitness”

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Neutral</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calorie Counter (Base App)</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Neutral</td>
<td>2</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Negative</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Sentiment distance**

Positive – Positive -> 0 ; positive – Neutral -> 1; Positive – Negative -> 2
Neutral – positive -> -1; Neutral – Neutral -> 0; Neutral – Negative -> 1
Negative – Positive -> -2; Negative – Neutral -> -1 ; Negative -> Negative -> 0

**Calorie Counter vs. Map My Fitness Score**

\[
\text{Calorie Counter vs. Map My Fitness Score} = \frac{\text{SUM}(\text{table\_cell\_count}[i] \times \text{sentiment distance})}{\text{feature\_count}}
\]

\[
= \frac{[1 \times 0 + 1 \times 1 + 0 \times 2 + 2 \times (-1) + 1 \times 0 + 0 \times 1 + 0 \times (-2) + 0 \times (-1) + 0 \times 0]}{5}
\]

\[
= \frac{[1 - 2]}{5} = -0.2
\]

**Competing app is slightly better than the base app**
Result 2:
Feature-based comparison of the base app “Calorie Counter” with competing app “Run Keeper”

<table>
<thead>
<tr>
<th>Calorie Counter (Base App)</th>
<th>Positive</th>
<th>Neutral</th>
<th>Negative</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>Neutral</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Negative</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Missing</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Calorie Counter vs. Run Keeper = \[ \frac{\text{SUM}(\text{table\_cell\_count}[i] \times \text{sentiment distance})}{\text{feature\_count}} \]

= \[ \frac{[1 \times 0 + 0 \times 1 + 0 \times 2 + 0 \times (-1) + 3 \times 0 + 0 \times 1 + 0 \times (-2) + 0 \times (-1) + 0 \times 0]}{4} \]

= 0
Result 3:
Feature-based comparison of the base app “Calorie Counter” with competing app “Strava Running and Cycling”

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Neutral</th>
<th>Negative</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calorie Counter (Base App)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Neutral</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Negative</td>
<td></td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missing</td>
<td></td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Calorie Counter vs. Strava Running and Cycling = \[ \text{SUM}(\text{table_cell_count}[i] \times \text{sentiment distance}) / \text{feature_count} \]
= \[ [0 \times 0 + 1 \times 1 + 1 \times 2 + 1 \times (-1) + 1 \times 0 + 0 \times 1 + 0 \times (-2) + 0 \times (-1) + 0 \times 0] / 4 \]
= 0.5

Overall score of the base app compared to the competing app is positive

Competing app misses a feature perceived neutrally by the base app users
Application Examples – Summary

• Ex1 – Release Readiness – RAISE 2016 (PhD)
• Ex2 – Issue RT (a) – MSR 2016 (PhD)
• Ex3 – Issue RT (b) – EASE 2016 (MSc)
• Ex4 – App Reviews – WAMA 2016 (MSc/PhD)
• More (ongoing PhDs):
  – Green Software
  – Open Innovation (RE)
  – …
• Many MSc thesis topics
Thank You!