A Machine Learning-based Impact Analysis Tool and its Improvement Using Co-occurrence Relationships

Teppei Kawabata, Tsuyoshi Nakajima, Shuichi Tokumoto, Ryota Tsukamoto, Kazuko Takahashi

Shibaura Institute of Technology, Information Technology R&D Center, Mitsubishi Electric Corporation
Table of Contents

1. Conventional impact analysis methods and their problems
2. Proposed impact analysis method using machine learning
3. Four proposed algorithms in machine learning considering multilabel classification
4. For a comparative evaluation of the above four algorithms
Background: Importance of impact analysis

Software change impact analysis plays an important role in controlling software evolution in the maintenance of continuous software development.

- It is important to improve the accuracy and efficiency to obtain modification candidates. This is because it is difficult to automate determining whether a modification candidate is really a modification target or not, requiring a lot of efforts.
- However, the problem is that it depends on the amount of developer's knowledge about the source code base.
Conventional method: Impact analysis with traceability

Traceability: established linkage between multiple deliverables in the development process[1]

Problem 1: Need to establish traceability links in advance

Problem 2: It is not applicable against new change requests that are not related to existing requests

Problem 3: Too many modification candidates to review in reasonable amount of time

Traceability links: information that shows the relationship between specific artifacts

Proposed method: Learning from change histories.

Our method is:

- To learn from a large number of change histories from past projects, and
- To create modification candidates from a change request.

Frontend of the system:

```
<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Request (Japanese)</td>
<td>List of modification candidates</td>
</tr>
<tr>
<td>Text vectorization + machine learning</td>
<td></td>
</tr>
<tr>
<td>Text vectorization + machine learning</td>
<td>1 component 5</td>
</tr>
<tr>
<td></td>
<td>2 component 3</td>
</tr>
<tr>
<td></td>
<td>3 component 28</td>
</tr>
</tbody>
</table>
```

PROBLEMS TO BE SOLVED:

- Not necessary to establish links in advance
- Applicable to new change requests
- Create modification candidates directly
Proposed method: composition of the algorithm

Change design document

modified targets

Change Request (text)
Component 2
Component 4

extract

Change Request (text)

Vectorization

Machine-Learning Component

Component vector (when ML:0,1/an inferring score)

Modification targets

0 1 0 1 0 0 0 0
Proposed method: How to implement sentence vectorization

Vectorizing steps

1. Word extraction
   - Extract nouns only
   - Selection by developer (Weighting)

2. Word vectorization
   - word2vec

3. Vector association
   - simple average
   - weighted average
   - doc2vec

Three implementations were evaluated

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation 1</td>
<td>noun only</td>
<td>word2vec</td>
<td>simple average</td>
</tr>
<tr>
<td>Implementation 2</td>
<td>All</td>
<td>word2vec</td>
<td>doc2vec</td>
</tr>
<tr>
<td>Implementation 3</td>
<td>noun only</td>
<td>word2vec</td>
<td>doc2vec</td>
</tr>
</tbody>
</table>
Previous study: Neural Network as the machine learning component

Configuration of the NN

The source code base used in our experiments has 32 components

Hyper Parameters
- Number of studies performed: 50
- Batch size: 50
- Learning rate: 0.1
- Loss function: binary cross-entropy error
- Weight parameter update method: SGD

Vectorized sentence (100) → ReLu (1000) → ReLu (500) → ReLu (300) → ReLu (100) → sigmoid → output (32)

[2] Y. Iwasaki, Proposal of a system that recommends candidate program changes from requirement text by learning past change, 2020
Evaluation Methods and the results of the previous study

We defined three indexes for the given threshold of Sigmoid value.

A) Candidate Range ratio

\[ A = \frac{S}{W} \]

B) Accuracy in the candidate range

\[ B = \frac{(S \cap T)}{S} \]

C) Missing rate

\[ C = \frac{(W - S) \cap T}{T} \]

The results of the previous study.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>A) Accuracy in the candidate range</th>
<th>B) Percentage of correct answer</th>
<th>C) Missing rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.06</td>
<td>30.0%</td>
<td>35.0%</td>
<td>23.0%</td>
</tr>
</tbody>
</table>

Missing modification targets has serious consequences.
Our idea to reduce missing rate

Hypothesis

A specific change pattern may cause modification of the same combination of components

Rationale

From the architectural point of view, some components may use common resources, or some call relationships exist between layers.

Idea for improvement

Adopting multi-label classifiers that model the co-occurrence relationship

Dependencies arising from architecture

Layer A
Component A1  Component A2  Component A3  Component A4

Layer B
Component B1  Component B2  Component B3  Component B4

Common Resources

call relationship
The four algorithms implementation to be evaluated

• Previous study
  ➢ Neural Network (NN)

• Basic Methods for Handling Multilabel Classification
  ➢ Binary Relevance (BR) method

• Methods modeling co-occurrence relationships
  ➢ Label Powerset (LP) method
  ➢ Random k-Labelsets (RAkEL) methods
Binary Relevance (BR) method

- Binary Relevance (BR) is a multilabel classification method, which learns a binary model for each label independently of the rest.
- This method does not model the co-occurrence relationships.
The four methods evaluated in this paper

• Previous study
  ➢ Neural Network (NN)

• Basic Methods for Handling Multilabel Classification
  ➢ Binary Relevance (BR) method

• Methods using co-occurrence relationships
  ➢ Label Powerset (LP) method
  ➢ Random k-Labelsets (RAkEL) methods
Algorithm 1 for modeling the co-occurrence relationship

Label Powerset (LP) method

LP is a multilabel classification method that models the co-occurrence relationship, considering all distinct combinations of labels as a different class and conducting a single-label classification for each.

### Estimation Results for each label in the class

<table>
<thead>
<tr>
<th>class</th>
<th>Probability of occurrence of class</th>
<th>Estimation Results for each label in the class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>set1</td>
</tr>
<tr>
<td>set[1,2]</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>set[31,32]</td>
<td>0.2</td>
<td>-</td>
</tr>
<tr>
<td>set[1,2...,32]</td>
<td>0.3</td>
<td>1</td>
</tr>
</tbody>
</table>

Total evaluation: \(0.1 \times 1 + 0.3 \times 1 = 0.2 \times 1 + 0.3 \times 1\)

Disadvantage: large amount of calculation and over-learning
The four methods evaluated in this paper

• Previous study
  ➢ Neural Network ( NN )

• Basic Methods for Handling Multilabel Classification
  ➢ Binary Relevance ( BR ) method

• Methods using co-occurrence relationships
  ➢ Label Powerset ( LP ) method
  ➢ Random k-Labelsets ( RAKEL ) methods
Algorithm 2 for modeling co-occurrence relationships

Random k-Labelsets (RAkEL) methods

- RAkEL is a multilabel classification method that models the co-occurrence relationship, breaking the initial set of labels into a number of small random subsets, called labelsets and employing LP to train a corresponding classifier.

![Diagram of RAkEL method]

<table>
<thead>
<tr>
<th>Class</th>
<th>Estimation Results for each Label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>label1</td>
</tr>
<tr>
<td>set[1,31]</td>
<td>1</td>
</tr>
<tr>
<td>set[2,31]</td>
<td>-</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>set[1,2,...32]</td>
<td>0</td>
</tr>
</tbody>
</table>

| Total evaluation | $T_1/M_1$ | $T_2/M_2$ | ... | $T_{31}/M_{31}$ | $T_{32}/M_{32}$ |

$T_1$: Number of cells whose estimated result is 1, $M_i$: Number of cells with estimated results
Purpose of experiment

To investigate whether the LP and RAkEL methods, which model the co-occurrence relationship, improve accuracy or not.

The four methods were evaluated using the same field data.

<table>
<thead>
<tr>
<th>Multi-label classification method</th>
<th>Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1: Neural Network (NN)</td>
<td>SVM</td>
</tr>
<tr>
<td>M2: BR method</td>
<td>SVM</td>
</tr>
<tr>
<td>M3: LP method</td>
<td>SVM</td>
</tr>
<tr>
<td>M4: RAkEL method</td>
<td>SVM</td>
</tr>
</tbody>
</table>
Data used in the experiments

Total data is 405.

- Change design specifications
- Request sentence
- Component list
- Training data (324)

Training data (324)
- Request sentence
- Component list

Test data (81)
- Request sentence

Multiple classification Method
- Modification candidate

(study)
Results of the experiment

The threshold was set so that the candidate range ratio is around 30 percent.

<table>
<thead>
<tr>
<th>Method</th>
<th>Candidate Range ratio</th>
<th>Accuracy in the candidate range</th>
<th>Missing rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1:NN</td>
<td>30.00%(0.06)</td>
<td>18.00%</td>
<td>23.00%</td>
</tr>
<tr>
<td>M2:BR+SVM</td>
<td>29.10%(0.06)</td>
<td>19.10%</td>
<td>17.10%</td>
</tr>
<tr>
<td>M3:LP+SVM</td>
<td>29.70%(0.06)</td>
<td>22.20%</td>
<td>23.00%</td>
</tr>
<tr>
<td>M4:RAkEL+SVM</td>
<td>29.50%(0.07)</td>
<td>24.50%</td>
<td>15.60%</td>
</tr>
</tbody>
</table>

Result

1. M2 is more accurate than M1 → SVM is an excellent classifier
2. M3 is 5.9% worse than M2 → Small number of data could have caused overlearning.
3. M4 is the most accurate one.

RAkEL provides the best results, meaning to model the co-occurrence relationship has a good effect to reduce missing rate. However their missing rates are not at enough level for practical use.
Summary and Future Issues

Summary

• We proposed an impact analysis method that learn change histories to directly create modification candidates.

• To improve the previous study, which use NN as the machine-learning component, we proposed a multi-label classification method considering the co-occurrence relationship.

• The effectiveness of this method was confirmed by an experiment using BR, LP, and RAKEL methods.

Future Issues

• Application of an improved algorithm for the RAKEL method

• Validation by using the other data set (from OSS)
Supplementary data: Reasons for determining target values

Utilizes standard deviation (σ), a value often used in quality control

• ± σ (σ interval): 68.3%
• ± 2σ (2σ interval): 95.4%
• ± 3σ (3σ interval): 99.7%

z-distribution diagram
Supplementary material: Target projects used for the study

- Each project modifies the program matrix for multiple change requests
- Create a change design document for each change request

(annual) Project 1  Project 2  Project 3  ...  Project 30
(average) change requests(10) change requests(10) change requests(10) change requests(10)

Approximately 300 in total

same program entity
Improved machine learning implementation methods.

Apply and evaluate machine learning methods that consider co-occurrence relationships to reduce the hazard rate that has been the subject of previous research.

research goal

Performance targets, taking into account the extent to which this is possible in terms of actual audits:

Candidate Range ratio $\leq 30\%$ and, Missing rate $\leq 5\%$