Software error grouping with machine learning

Arpad Tamasi CS 229 (Aut 2019) Final Project

Introduction

Software error messages are generated from template strings, filled with context information. We aim to improve error classification by learning the templates and identify the context information from error messages generated by different applications. We want to be able to group messages like `Cannot read property 'vdata1567' of null` by their template (`Cannot read property '%s' of null`) and property name (`vdata1567`). **Other messages:**

- `Cannot read property 'vdata1234' of null`
- `Cannot read property 'isDestroyed' of undefined`
- `ecommerceData is not defined`
- `The action 'buy_now' could not be found for Api::V1::LotsController`

Data

19.8 million JavaScript errors in September 2019 from 6,300 different projects. **Attributes**:

- class name
- error message
- no labels

Features

- tokens with embedded context: class name and token index (`TypeError|0|Cannot`, `TypeError|1|read` etc)
- features: the tokens projected into a 4096 dimension space using binary term frequency hashing

Agglomerative clustering

In the projected space the static words of templates corresponds a vector. Let k_t be the number of placeholders, w the number of words in the template. The squared euclidean distance between the template and its messages $d(t, m_t^{(i)}) = k_t$. Also, the distance of two messages generated with the same template $\|m_t^{(i)} - m_t^{(j)}\|_2^2 \leq 2^{k_t}$. It is equivalent to the double of the Levehnstein distance if the messages have ot the same number of words: $|m_t^{(i)}| = |m_t^{(j)}|$

- Metric: squared Euclidean distance $\|m^{(i)} - m^{(j)}\|_2^2$
- Linkage criteria:
  - complete linking
  - $\|m^{(i)} - m^{(j)}\|_2^2 \leq max\_dist$ (limit the number of placeholders)
  - $|m^{(i)} = m^{(j)}|$ (same number of words)

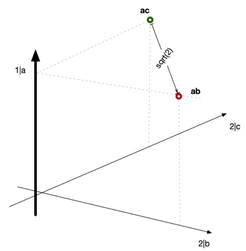

Figure 1: Two messages generated from the a? template.

References

[1] Software error grouping https://youtu.be/kdq7bxekifs.

[2] Wikipedia: Hierarchical clustering.

[3] Apache spark ml guide.

PCA + K-means

Assuming that the messages have been generated by t templates, we apply PCA to transform the data to $k \geq t$ principal components then K-means to find the messages generated by the templates.

The first t components identify the templates, the rest $k - t$ describe the variables in the placeholders.

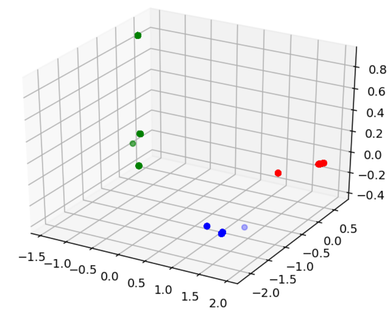

Figure 2: 3 templates represented by 3 components

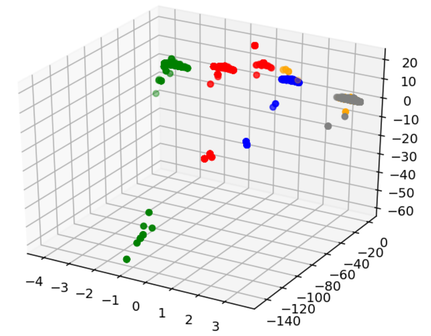

Figure 3: 5 templates represented by 3 components

Results

Agglomerative clustering identified 3599 clusters. 603 of them appeared in multiple projects and covered 86% of the crashes. We generated regular expressions from the messages belonging to the same cluster.

This method was very robust to collisions introduced by hashing and common words, separated clearly the common patterns and outliers.

- `Cannot read property '(\w+)' of null`
- `Cannot read property '(\w+)' of undefined`
- `(\w+) is not defined`
- `The action '\w+' could not be found for (\S+)`

PCA was very sensitive to collisions and outliers so it was practically useless without data cleaning. We used the results of agglomerative clustering for it. We kept the errors that appeared in multiple projects and their identified generated at least 100 distinct messages. This covered 12 patterns. PCA found 3 identical clusters and 9 mixed ones.

Discussion & Future

Unsupervised learning is a great to explore large datasets and understand its structure. The results of **agglomerated clustering** saves hundreds of hours by identifying the recurring templates in messages. It improves the specificity of grouping that 80% of users found helpful.

Using **PCA** for grouping without recovering the templates looks an extremely promising alternative. It lets us easily handle other features like the stack frames. However, it seems that we need much larger dataset and more sophisticated features to overcome the noise.

As an alterenative, the results opf agglomerated clustering can be used to label and augment datasets that can be used to train neural networks.