Technical Report

Xunhui Zhang, Yue Yu, Tao Wang, Ayushi Rastogi, Huaimin Wang

1. Factor correlation heatmaps

Figures 1, 2 and 3 show the heatmaps of correlation between factors, representing the correlation between continuous factors, the correlation between categorical factors and the correlation between continuous and categorical factors.

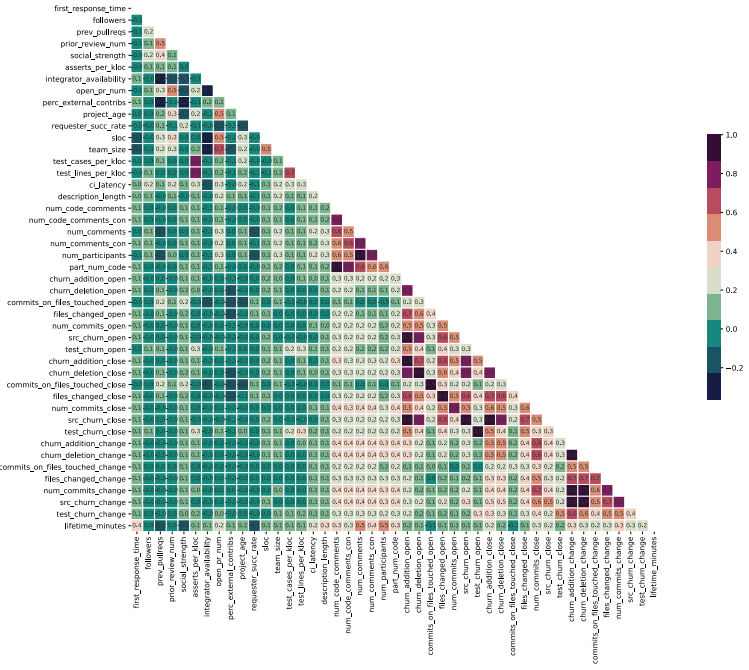


Fig 1. Heatmap of Spearman correlation coefficients

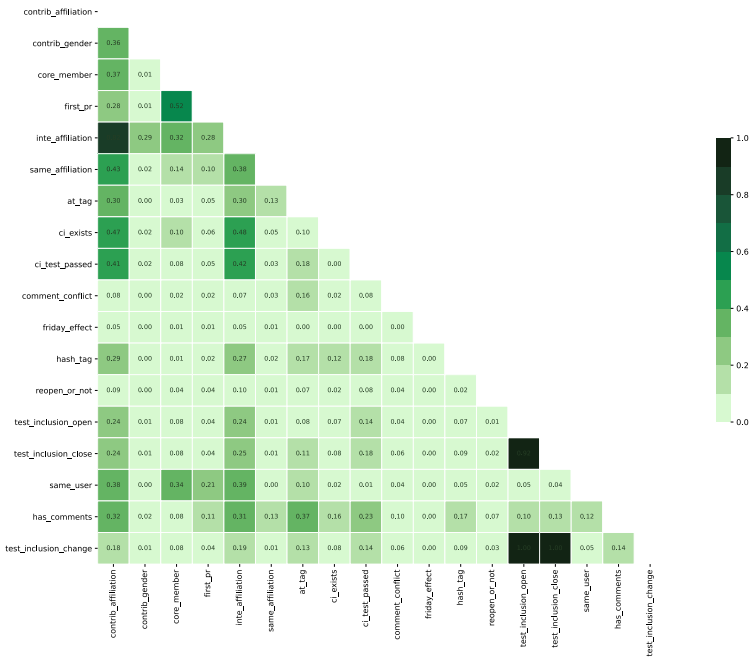


Fig 2. Heatmap of Cramer's V value

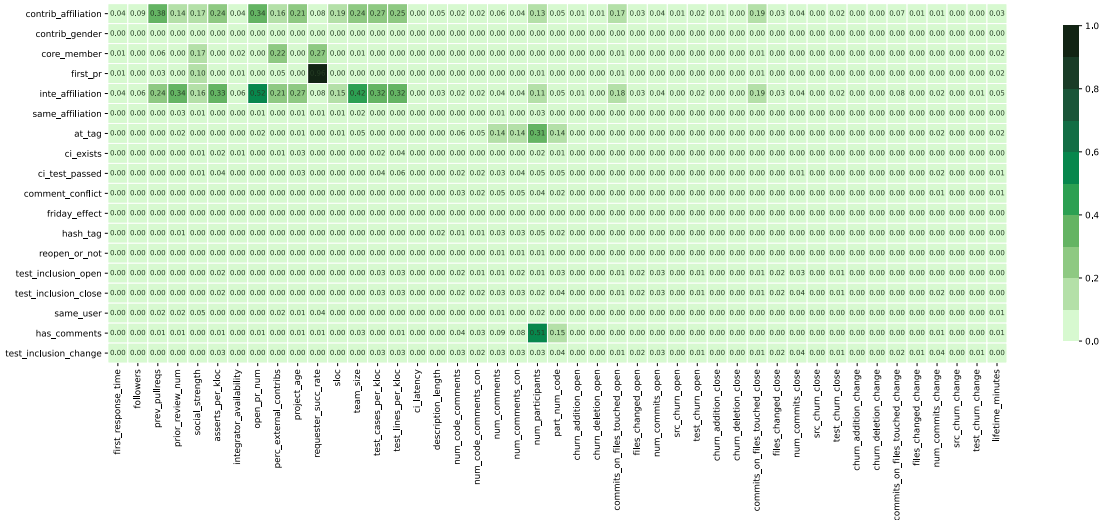
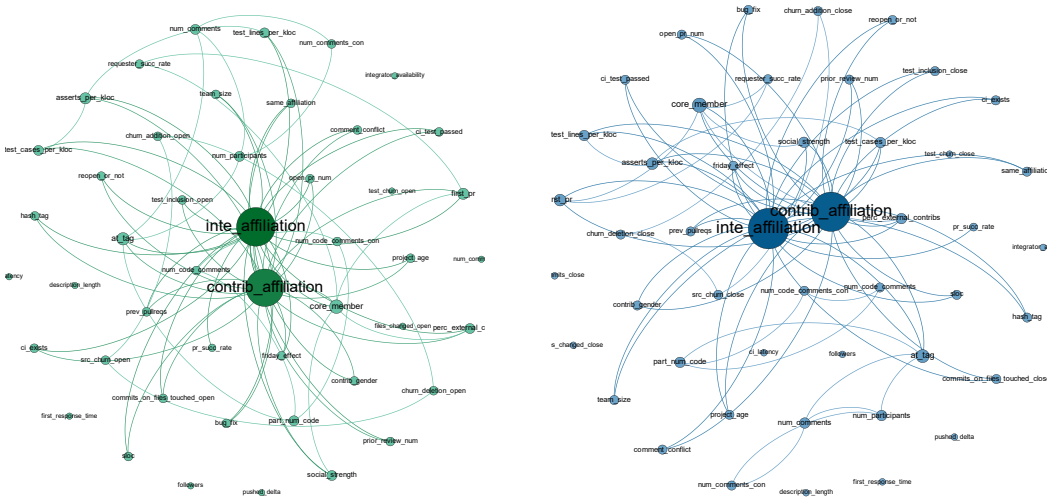


Fig 3. Heatmap of partial Eta-square value

2. Strong correlation network

Figure 4 presents the correlation network among strongly correlated factors, where node size and text size positively correlate with node's degree. So does the node colour depth.



4a. At pull request open time 4b. At pull request close time

Fig 4. Correlation network of strongly correlated factors

3. Factor selection workflow

After calculating factor correlation, we removed the strongly correlated factors according to the workflow shown in Figure 5.

The basic principle is to keep popular factors while removing factors with the strongest correlations.

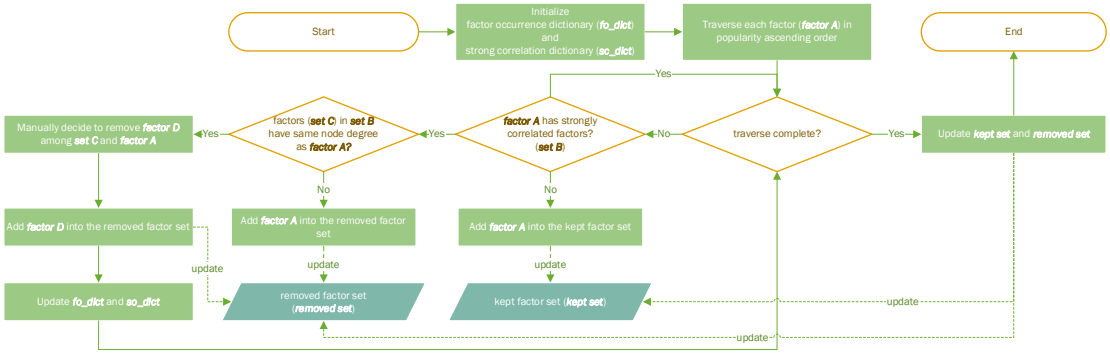


Fig 5. Workflow of factor selection

1. We calculated the number of related papers (popularity) for each factor and formed it into the factor occurrence dictionary (fo_dict). Also, we initialize the strong correlation dictionary (sc_dict) according to a strong correlation network.
2. Then we traverse each factor (factor A) in popularity ascending order.
3. We add factor A into kept factor set if it does not have strongly correlated factors.
4. If factor A has strong correlations, find strongly correlated factors with the same popularity (set C) as factor A.
5. If no factor is found (set C is empty), we remove the currently least popular factor A by adding it to the removed set.
6. If we find factors with the same popularity and strongly correlate with factor A, we manually decide which factor (factor D) to remove first and add it to the removed set.
7. Then we update fo_dict and sc_dict by deleting the information related to the removed factor for the next traversal.
8. When we finish the traversal, we need to update the kept set and removed set by checking removed factors that do not correlate strongly with those in the kept set. E.g., the factor contrib_gender is removed because it is strongly correlated with contrib_affiliation but less popular. However, after removing contrib_affiliation, no strong correlation for contrib_gender exists anymore.

When making manual decisions in step 6, we remove churn_deletion_open instead of churn_addition_open, as churn_addition significantly influences pull request latency. We remove num_code_comments_con instead of num_code_comments, as num_code_comments is more general. We remove inte_affiliation instead of contrib_affiliation, as one previous study found that contrib_affiliation is significantly important.