Paper Nine Summary

Matthew Neal, Joseph Sankar, and Alexander Sobran

December 12, 2015

Reference

Nam et al. [1] listed below.

Important Keywords

Transfer defect learning Extracting common knowledge from one task domain and transferring it to another. The transferred knowledge is then used to train a prediction model.

TCA+ An improvement on Transfer Component Analysis. TCA can map the data of the source and target projects on a latent feature space, but is sensitive to normalization. TCA+ selects a proper normalization to yield better prediction performance.

Data set Characteristic Vector A vector of six elements each relating to the distance between pairs of instances of data. DCVs are used to see how similar two projects are (see below).

Similarity vector Represents the difference between two projects: a source and a target. Examples include "much more", "less", or "same". The values in the DCVs are used to calculate the similarity vectors.

Feature Extraction

Motivational statements Cross-project defect prediction is often necessary for new projects, but doesn't always yield good results. Therefore, the authors hope to employ transfer defect learning to improve the performance of these models.

Data The authors have provided the data used in their experiments at https://sites.google.com/site/transferdefect/.

Future work The authors were looking into transferring knowledge across entire domains as an extension of transfer learning. They were also interested in seeing which other prediction and recommendation systems might benefit from transfer learning.

Statistical Tests The authors use the F-measure as a metric to determine the performance of TCA vs TCA+ vs within target cross prediction. They state that because of the trade off between precision and recall, providing F-measure is a better metric to provide overall.

Possible Improvements

- The authors make a statement on page five in the section on expirimental design that is unfortunately false. They state that 10-fold cross validation essetially uses 90 percent of the training data and 10 percent of the test data as a downside of 10-fold cross validation. They neglect to state that it does that 10 times and actually goes through the entireity of the 10 possible permutations of the data and calculates its error based on that. It may have been enough to state that random split was used widely in the literature and leave out the statement on 10-fold.
- There is not a future work section that is actually specifically split out in the paper, and what they do provide is really sparse. A bit of expansion on that would have been extremely helpful.
- Tables X and XI could be improved to be a bit simpler to understand. The within target $\rightarrow$ target column is a bit confusing, and furthermore the reasoning for only providing (N4) and (N2) normalization techniques in X and XI respectively is a bit confusing as well.

Connection to Other Papers

- There is a direct connection with [2] as the author discusses his paper as evidence that cross-project defect prediction works as well as within-project prediction in terms of cost effectiveness.

References

[1] Jaechang Nam, Sinno Jialin Pan, and Sunghun Kim. Transfer defect learning. In *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, pages 382–391, Piscataway, NJ, USA, 2013. IEEE Press.

[2] Foyzur Rahman, Daryl Posnett, and Premkumar Devanbu. Recalling the "imprecision" of cross-project defect prediction. In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, FSE '12, pages 61:1–61:11, New York, NY, USA, 2012. ACM.