Using Bayesian data analysis techniques to predict change sets in IT systems
CS886 - Fall 2010
Sarah Nadi

1 Introduction

For the class project, I propose doing an empirical evaluation of different Bayesian data analysis techniques to predict change sets in an IT system (Option B). Before applying a change in an IT system, we need to check if there are any other components that need to be changed for the change to be complete, and that we will not negatively impact any other components in the system. In this domain (Enterprise IT management), a system is composed of both software and hardware components, and each component is called a configuration item (CI).

In previous work [1], we tried to predict the change set (given the initial CI that will be changed) using historical data. To do that, we kept track how many times each pair of CIs changed together. Let us assume that we want to change A, then we would suggest that B should be changed as well if it has changed frequently enough with A in the past. This is a very high level view of the approach. We got very high recall and precision in our results. However, there are still some limitations. Let us assume we want to change A and B, and we want to check if there are other CIs we should change. The algorithm would separately check the CIs that changed with A *alone* frequently and suggest them, and do the same for B while what we really want are the CIs that changed with *both* A and B frequently. The current algorithm cannot do that since we only look at pairs of CIs to construct our model.

Accordingly, we want some way to capture the fact that changing A alone versus changing both A and B together may alter our belief that C should also change. Bayesian networks seem to be a good candidate for this kind of information. This would be a good chance to experiment with different Bayesian data analysis techniques.

2 Proposed techniques

Using probability theory to predict changes in a software system has been extensively employed on the source code level. I plan to investigate if these same techniques would work in the domain of IT systems in general. For example, Mirarab et al. [2] use Bayesian Belief Networks to model the relationship between different source code artifacts (these could be packages, files, or methods depending on the level of granularity required). To build their model they use structural learning, and use a noisy-OR assumption to assume that all parents are independent in terms of their influence on the child. The algorithm they use to learn the structure of the network is presented in the paper so I need to examine this in details. To address the problem I stated in the introduction, I'm not sure if this is an assumption I would want to make or not so this could be a point of investigation. However, the general learning technique used in this paper could be a point of reference. Zhou et al. [3] address the problem similarly, but use a supervised learning approach to construct the network using the K2 algorithm. This is another algorithm I will need to examine.

Additionally, I will look at equivalence classes [4] to see if I can benefit from them to model the data I have. Other papers I may benefit from include that by Tang et al. [5]. Finally, in the initial work we did, we used exponential forgetting to give more weight to more recent information, but found out that it did not affect the results (implying that the system is more or less stable) It would be interesting to employ stochastic dependencies as explained by Wong et al. [6] to see if the system is indeed stable or not.

References

[1] S. Nadi, R. Holt, and S. Mankovskii. Does the past say it all? Using history to predict change sets in a CMDB. In *CSMR'10: Proceedings of the 14th European Conference on Software Maintenance and Reengineering*, Madrid, Spain, 2010.

[2] S. Mirarab, A. Hassouna, and L. Tahvildari. Using bayesian belief networks to predict change propagation in software systems. pages 177 –188, jun. 2007.

[3] Yu Zhou, Michael Würsch, Emanuel Giger, Harald C. Gall, and Jian L? A bayesian network based approach for change coupling prediction. *Reverse Engineering, Working Conference on*, 0:27–36, 2008.

[4] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

[5] A. Tang, A. Nicholson, Y. Jin, and J. Han. Using Bayesian belief networks for change impact analysis in architecture design. *Journal of Systems and Software*, 80(1):127–148, 2007.

[6] S. Wong, Y. Cai, and M. Dalton. Change Impact Analysis with Stochastic Dependencies. *submitted to ICSE 2011.*