An Investigation of the Use of BBNs for Project Management

Daniel Rodriguez Rachel Harrison Manoranjan Satpathy

Department of Computer Science, University of Reading, Reading RG6 6AY, UK

{d.rodriguez-garcia, Rachel.Harrison, M.Satpathy}@reading.ac.uk

Abstract

Dynamic models such as Bayesian Networks are of late becoming increasingly popular among the Software Engineering research community. Such models take into account the causal relationship between the variables of a domain, and therefore may provide better solutions to many of the problems in this area. This paper concerns the use of Bayesian Belief Networks (BBNs) in project management. Creation of a BBN to reflect a software engineering problem is a challenging task. In this paper, we will concentrate on the creation of a BBN by using domain data. There are many semi-automatic approaches available in literature but there are also many open issues in these approaches. We will discuss the step by step construction of a BBN using the dataset from a Web-related academic projects involving 37 Web sites.

Key Words*: Bayesian Networks, Project Management, Estimates*.

1 Introduction

A Bayesian Belief Network (BBN) [4] is a directed graph in which the nodes represent uncertain variables and the arcs represent the causal relationship between the variables. Each node has a probability table, which stores the conditional probabilities for each possible state of the node variable in relation to each combination of its parent state values. For a node without any parents, such a table stores the marginal probabilities for each possible state of that node. If the state of a certain node is known then its probability table is altered to reflect this knowledge. Such knowledge is then propagated to determine the changed probabilities of all possible values associated with other nodes.

2 Mining Software Engineering Data using BBNs

In this section we will show how data mining techniques and BBNs can help us to acquire knowledge about the software engineering process of an organization. In general, data mining techniques try to extract in an automatic way the information useful for decision support or exploration of the data source [3]. Since data may not be organised in a way that facilitates the extraction of useful information, typical KDD processes are composed of the following steps: (i) Data preparation. The data is formatted in a way that tools can manipulate it. (ii) Data selection and cleaning. There may be missing, noisy and uncertain data in the raw dataset. (iii) Data mining. It is in this step when the automated extraction of knowledge from the data is carried out. (iv) Proper interpretation of the results, including the use of visualization techniques. (v) Assimilation of the results.

2.1 Learning BBNs from Data

When data is available, automated learning methods have been developed for learning, i.e. obtaining, BBNs from data. The process is composed of two main tasks: (i) induction of the best matching qualitative dependency model from data and prior knowledge; and (ii) estimation of the local probabilities.

2.2 The Process of Creating BBNs from Data

We will illustrate this process following an example in the context of estimation in Web Engineering using the dataset provided by Mendes *et al* [5].

Data Pre-processing:

Our dataset is composed of 37 Web applications (3 were outliers) which were created by final year undergraduate students. From this dataset, the authors identified 8 variables that characterise a Web hypermedia application and its development process.

Metric	***Description***
Page Count (PaC)	Number of HTML or SHTML files used in the application
Media Count (MeC)	Number of Media files used in the application
Program Count (PRC)	Number of JavaScript files and Java applets used in the application
Reused Media Count (RMC)	Number of Reused/modified media files
Reused Program Count (RPC)	Number of Reused/modified programs
Connectivity Density (CoD)	Total number of internal links divided by *Page Count*
Total Page Complexity (TPC)	Average number of internal links divided by *Page Count*
Total Effort (TE)	Effort in person hours to design and author the application

Table 1 Size and Complexity Measures [5]

In order to create a BBN from data, it needs to be pre-processed. Typically, this process consists of discretizing data and dealing with missing values. Table 2 partially shows the pre-processed dataset where new columns with discretized data have been added. Columns that were already discrete and new columns with discretized data will be used in the learning process.

ID	***TE***	***PaC***	***MeC***	***PRC***	***…***	***TE_d***	***PaC_d***	***MeC_d***
1	79.13	43	0	0	…	<92.175	<50.5	<0.5
2	133.1	53	53	1	…	>128.45<150.1	>50.5<53.5	>27.5<78
…	…	…	…	…	…	…	…	…
34	141.4	52	48	5	…	>128.45<150.1	>50.5<53.5	>27.5<78

Table 2 Pre-processed data

Learning process:

This process consists of learning the network structure and generating the node probability table. In order to learn the structure, we need to apply domain knowledge both before and after the execution of the learning process. Before executing the learning process, it may be necessary to totally or partially define the structure of the network (node ordering for known causal relationships, forbidden links and root/leaf nodes) and some parameters such as thresholds for the accepted dependencies. After executing the algorithm, we may need to edit the network to add, remove or label arcs if it was impossible to do this from the data.

There are several projects under the GNU license that implement well-known *search and scoring* and *dependency analysis* algorithms in Java. However, we found that using wizard-like GUI interfaces to define the structure facilitates the above task. Among these software packages we used BNPC tool [1] and Bayesware Discoverer [6]. Figure 1 shows a possible BBN generated after running the BBN Power Constructor tool.

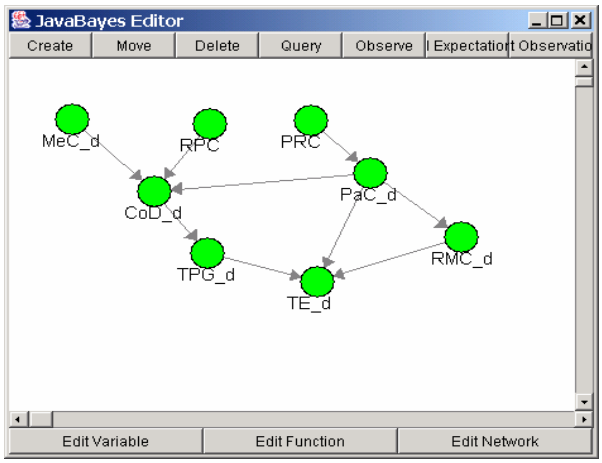

Figure 1 BBN generated from Mendes' dataset

At this stage the resulting network can be used by Bayesian network inference tools; in our case, we used USEGESOFT tool which includes a modified version of the JavaBayes tool [2]. Figure 1 shows the final network from which we should be able to show how predictions might be made and historical results can be explained more clearly (Figure 2).

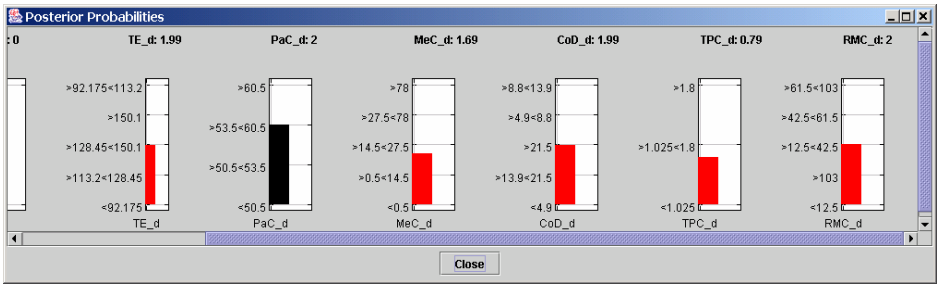

Figure 2 Probabilities Window of a BBN for Web Authoring Estimates

Validation of the BBN:

Presently, we will compare the results obtained from this network with the actual dataset from which we obtained the network. Table 3 shows the maximum, minimum and the mean values of the 8 variables in the original dataset. We use the network to obtain the value of TE (total effort) by feeding to the network the values of the remaining 7 variables. The results that we obtain from the BBN are shown in Figure 3. For the minimum values of the said 7 variables, Table 6.4 says that TE has a value of 58.36. The network shows (refer to Figure 6.9) that the probability that the value of TE remains less than 92 is 55%, and the other possible values of TE are very low. When the maximum values are considered, the table shows that value of TE is 153.78, whereas the Figure shows that the probability of TE being 150 is maximum (33%). So far as the mean values are considered, the table shows that TE has a value of 111.89, whereas the network shows that the probability of the TE value remaining between 92 and 128 is high. What we can say from these results is that the values obtained by the network to a large extent approximates the actual values, although they are not very accurate. One reason is that the

number of cases considered in the dataset is low (34). Furthermore, the network construction process uses expert knowledge at intermediate stages, which might have not been perfect in our case.

Metric	*Mean*	*Min*	*Max*
PaC	55.21	33	100
MeC	24.82	0	126
PRC	0.41	0	5
RMC	42.06	0	112
RPC	0.24	0	8
COD	10.44	1.69	23.3
TPC	1.16	0	2.51
TE	111.89	58.36	153.78

Table 3 Summary Statistics

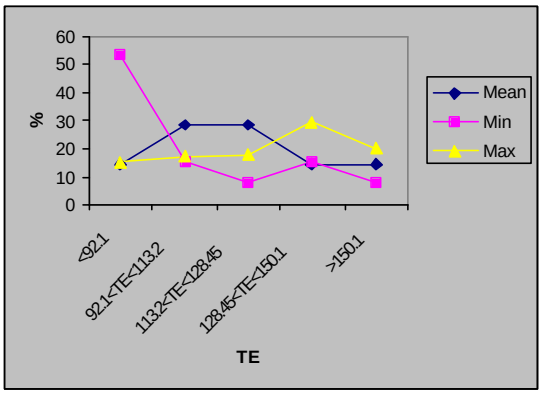

Figure 3 Output Probabilities for the Variable *Total Effort* (TE)

3 Conclusion

Computation with BBNs can be a very powerful technique in software engineering problems like estimation and risk analysis because: (i) they take into account the cause-effect relationships of the variables in the problem domain, and (ii) the backwards propagation of the probabilities in the BBN can help in taking managerial decisions that might not be possible using static models.

Construction of BBNs is an intellectual task. We have discussed a semi-automatic approach which uses many prototype tools for constructing BBNs. We have also discussed a practical example and constructed a BBN in relation to the given dataset. Research on validation of BBNs is an important part of our future work.

References

[1] J. Cheng, "Belief Network (BN) Power Constructor", 2.2 Beta ed, 2001.

[2] F. G. Cozman, "JavaBayes 0.346", 0.346 ed, 2001.

[3] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD Process for Extracting Useful Knowledge From Volumes of Data", in *Communications of the ACM*, vol. 39, 1996, pp. 27-34.

[4] F. V. Jensen, *An Introduction to Bayesian Networks*. London: UCL Press, 1996.

[5] E. Mendes, I. Watson, C. Triggs, N. Mosley, and S. Counsell, "A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications", 8th IEEE Metrics Symposium, Ottawa, Canada, 2002.

[6] M. Ramoni and P. Sebastiani, "Bayesware Discoverer", 1.0 ed, 2002.