Decision Support Systems 56 (2013) 439–449

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Software project risk analysis using Bayesian networks with causality constraints

Yong Hu [a,*], Xiangzhou Zhang [b], E.W.T. Ngai [c], Ruichu Cai [d], Mei Liu [e]

[a] Institute of Business Intelligence and Knowledge Discovery, Guangdong University of Foreign Studies, Sun Yat-sen University, Guangzhou 510006, PR China
[b] School of Business, Sun Yat-sen University, Guangzhou 510006, PR China
[c] Department of Management and Marketing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, PR China
[d] Department of Computer Science, Guangdong University of Technology, Guangzhou, PR China
[e] Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA

ARTICLE INFO

Available online 8 November 2012

Keywords:
Software project risk analysis
Bayesian networks
Causality analysis
Knowledge discovery
Expert knowledge constraint

ABSTRACT

Many risks are involved in software development and risk management has become one of the key activities in software development. Bayesian networks (BNs) have been explored as a tool for various risk management practices, including the risk management of software development projects. However, much of the present research on software risk analysis focuses on finding the correlation between risk factors and project outcome. Software project failures are often a result of insufficient and ineffective risk management. To obtain proper and effective risk control, risk planning should be performed based on risk causality which can provide more risk information for decision making. In this study, we propose a model using BNs with causality constraints (BNCC) for risk analysis of software development projects. Through unrestricted automatic causality learning from 302 collected software project data, we demonstrated that the proposed model can not only discover causalities in accordance with the expert knowledge but also perform better in prediction than other algorithms, such as logistic regression, C4.5, Naïve Bayes, and general BNs. This research presents the first causal discovery framework for risk causality analysis of software projects and develops a model using BNCC for application in software project risk management.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

The software industry has become one of the fastest-growing industries. The global software market is estimated to have a value of US$330 billion in 2014, an increase of 36.1% since 2009 (US$ 242.4 billion) [43]. However, software development is yet a high-risk activity. The "CHAOS Summary 2009" from the Standish Group reported that the success rate of global (mainly U.S. and European) software projects is only 32% [55]. Much previous research has shown that the most important problem in software engineering is risk management, whereas technical issues are only secondary. For example, the Standish Group's report "EXTREME CHAOS" [54] summarized the recipe for software project success, that is, the CHAO 10, most of which are non-technical factors. Risk management is critical to project management; it is one of the 9 knowledge areas in project management as defined in the Project Management Body of Knowledge (PMBOK) [42] and is one of the 25 key process areas as defined in the Capability Maturity Model Integration (CMMI) [9]. McConnell believes that to obtain a 50–70% chance of avoiding time overrun, risk management only requires 5% of the total project budget [31]. These reasons highlight the urgency and feasibility of software project risk management.

In the current practice, subjective analysis or expert judgment is one of the methods often used in project risk management [15]. It is based on the experience of an expert and is thus inevitably human-intensive and obscure [16]; likewise, it generally lacks repeatability as experience is not readily shared among different teams within an organization [35]. Therefore, it is crucial to develop intelligent modeling techniques that can provide more objective, repeatable, and visible decision-making support for risk management. Among various existing intelligent modeling techniques, the Bayesian network (BN) has attracted much attention, such as those presented in refs. [1, 16, 28], due to its excellent ability in representing and reasoning with uncertainties.

Most research on software project risk analysis focuses on the discovery of correlations between risk factors and project outcomes [13, 24, 60]. At present, studies on BN-based risk analysis of software projects involve two ways of network construction: (1) experts manually specify the network to reflect expert knowledge [14, 16], and (2) automatically learn the network from observational data [27]. Since the manual method is not based on observational data, it will certainly contain expert subjective bias. The existing automatic methods for BN network learning cannot distinguish correlation from causality. For instance, the edge orientation does not necessarily indicate which risk should be controlled to change another risk. However

* Corresponding author.
E-mail addresses: henryhu200211@163.com (Y. Hu), zhxzhou@mail2.sysu.edu.cn (X. Zhang), mswtngai@inet.polyu.edu.hk (E.W.T. Ngai), cairuichu@gmail.com (R. Cai), mei.liu@njit.edu (M. Liu).

0167-9236/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dss.2012.11.001

this limitation in existing algorithms is usually neglected. Such research models are not suitable for direct risk control.

Software project practitioners have long complained about the difficulty in determining the real and direct risks to guide the allocation of time and resources. Thus causality, rather than correlation, is of greater interest to industry experts in software project risk planning because it can determine the causal factors that directly affect project outcomes. For example, the risk of "project involving the use of new technology" may be correlated with "immature technology" because new technology is probably underdeveloped due to its unidentified bugs. Nevertheless, a new technology does not necessarily mean an immature technology. Whether we can mitigate the former risk by only focusing on the latter is not certain, and vice versa. Actually, we are advised to reduce the risks of using a new technology by referring to pilot investigations, preparing alternative technology, training of team members. National Aeronautics and Space Administration (NASA) considers that risk planning should first "make sure that the consequences and the sources of the risk are known" and "plan important risks first" [45]. The Software Engineering Institute of Carnegie Mellon University (CMU/SEI) requires the risk analysis process to satisfy the goal of "determining the source of risk", i.e., "the root causes of the risk" [18]. Hence, in risk planning, analyses of the consequences and risk sources are very important.

In this paper, we propose a novel framework for software project risk management using BNs with causality constraints (BNCC). Our primary objective is to perform a causality analysis between risk factors and project outcomes to achieve more effective risk control. Specifically, the analysis involves (1) introducing a new modeling framework for risk causality analysis to discover new causal relationships and validate existing ones (i.e., practical and/or academic expert knowledge) between risk factors and project outcomes based on historical data; and (2) constructing an empirical BN software project risk analysis model based on the framework, which can be readily used in risk planning.

Compared with other modeling algorithms such as C4.5 and Naïve Bayes, the proposed BNCC-based model has the following advantages: (1) strong interpretability — the constructed BN combines data with expert knowledge, depicts causal relationships between variables, and helps obtain better project outcomes or higher probability of project success; and (2) acceptable predictive accuracy — the final model in this study has better predictive power compared with other modeling algorithms, making the model suitable for capturing the statistical relationships between risk factors and project outcomes.

This study makes two important contributions. First, it proposes the first causal discovery framework for risk management of software projects, which builds an empirical model from real data and incorporates the causal discovery technique and expert knowledge. This risk modeling framework can be widely applied to other related domains. Second, it provides a BNCC model for risk analysis based on data from real industry software projects. The network has strong interpretability and can provide explicit knowledge (causal relationships between risk factors and project outcomes) of software projects. Subsequently, such knowledge can help in conducting effective risk analysis and further risk planning, which will result in a better implementation of software project risk management.

This paper is organized as follows. Section 2 provides a review of related literature. Section 3 describes the proposed risk model and the modeling concept. Section 4 presents the experimental results. Finally, Section 5 concludes and discusses limitations of the study.

2. Review of literature

2.1. Risk management of software projects

Risk management was first introduced to software project management by Boehm [3] and Charette [6]. According to the "IEEE Standard for Software Project Management Plans" [22], a software project is defined as a series of technical and managerial work activities that should meet the terms and conditions listed in the project agreement. Successful software project usually means that the project can be completed within the budget and given time, and meet the customers' demand for high-quality and high-performance. Wallace et al. [60] defined software project risk as a series of factors or circumstances that will be a threat to the successful completion of a software project.

Boehm [3] summarized the risk management process into two steps: risk assessment and risk control. Risk assessment involves three subsidiary steps: risk identification, risk analysis, and risk prioritization. Risk control also consists of three subsidiary steps: risk-management planning, risk resolution, and risk monitoring. Risk analysis mainly focuses on the relationships between risk factors and project outcomes, and is to prepare for further risk control. In NASA, risk analysis is the process of determining the extent of the risks, their relationships with each other, and the most important risks [45].

2.2. Risk analysis of software projects

Numerous statistical and data mining methods have been used to analyze the relationships between variables. For intelligent risk analysis of software projects, many works have employed these methods, including regression analysis [23], association rules [33], decision trees [63], fuzzy logic [62], clustering analysis [61], and neural networks [35]. Jiang and Klein [23], for instance, used multiple regression analysis to explore the various risks that significantly affect the multidimensional success of information system development. Moreno-García et al. [33] used association rules to estimate the influence of certain management policies on the software project output attributes, which include product quality, time spent, and effort exerted on the project. However, their methods have only been applied to data generated by a Software Project Simulator rather than real project data. Xu et al. [63] introduced a hybrid learning method that combines genetic algorithm and decision trees to derive optimal subsets of software metrics for risk prediction. Moreover, Xu et al. [62] developed a fuzzy expert system and illustrated how to infer the rules about software development in the early phase of its life cycle. Wallace et al. [61] performed *k*-means clustering analysis to explore the trends in risk dimensions across three clusters (i.e., low-, medium-, and high-risk projects), and then examined the influence of project characteristics (e.g., project scope, sourcing practices, and strategic orientation) on project risk dimensions. Neumann [35] combined principal component analysis with neural networks to perform software risk classification and to discriminate high-risk projects with imbalanced data sets.

Each method has its unique advantages. Regression analysis can establish the dependence between variables and can be used for prediction. Association rules can find rules that can satisfy user-specified minimum support and confidence based on (conditional) frequency counting. Decision trees are simple and easy to understand, while neural networks can capture the non-linear interdependence among variables. Fuzzy logic can aggregate the scores of risk factors into an overall project risk score based on fuzzy set theory, which is suitable for inexact risk assessment. Clustering analysis groups a set of observations into subsets based on the mutual similarity/dissimilarity of observations, without manually pre-defining specific categories. Unfortunately, none of these methods were developed to capture the causality relationships in the form of "A influences B." These methods may (unintentionally) discover some genuine cause–effect relationships, but they are unable to distinguish causality from correlation.

2.3. BN-based project risk management

BNs have a wide range of real world applications such as in diagnosis, forecasting, automated vision, sensor fusion, manufacturing control, transportation, ecosystem and environmental management [20, 56, 57].

Download English Version:

https://daneshyari.com/en/article/6948633

Download Persian Version:

https://daneshyari.com/article/6948633

Daneshyari.com