Engineering Applications of Artificial Intelligence 70 (2018) 149–158

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Data-driven prediction of change propagation using Dependency Network

Jihwan Lee[a], Yoo S. Hong[b,*]

[a] Division of System Management and Engineering, Pukyong National University, 45, Yongso-ro, Nam-Gu. Busan, Republic of Korea
[b] Department of Industrial Engineering, Seoul National University, 1, Gwanak-ro, Gwanak-gu, Seoul, Republic of Korea

ARTICLE INFO

Keywords:
Change propagation
Change management
Change prediction
Probabilistic graphical model
Dependency Network
Stochastic coupling

ABSTRACT

Change propagation is a central aspect of complex system developments. The prediction of change propagation is necessary to prevent further changes and to perform an assessment of the cost of planned changes. Bayesian Network has been applied to extract co-change patterns from the historical change log and to predict the probability of further changes caused by the change of other components. Due to the complexity of the Bayesian Network, however, its application to large scaled system can be limited. Also, Bayesian Network cannot represent the bi-directional relationship between system components. To address these limitations, this article proposes an alternative method using Dependency Network, which is an approximated version of the Bayesian Network. Detailed procedure for learning the DN from the data, as well as probabilistic inference algorithm using DN is explained. To show the feasibility of the model, a case study is conducted with empirical data obtained from the open-sourced software, Azureus. To validate the effectiveness of the proposed method, several experiments incorporating different parameters were conducted. The result confirms that our model can produce reliable and accurate estimation of change propagation probabilities.

1. Introduction

Complex systems are involved in many design changes during the development phase, in response to new requirements or the correction of existing problems. With the growing scale and complexity of the system, a change made in one system part often results in changes of other related parts (Giffin et al., 2009). This phenomenon is referred to as *change propagation*. The risk of change propagation has been noted by the literature. It may introduce many unanticipated changes to the system, which results in the cost overrun, schedule delay and quality problems (Eckert et al., 2004). Therefore, it is crucial to predict the change propagation in advance to prevent further changes and also to perform a pre-assessment of the cost of planned changes.

To predict change propagation, it is necessary to identify design couplings between system parts. To help identify the relevant part of a given system element, an analyser may use a pre-defined parametric relationship (e.g. call or use relationship between source code) or subjective opinion of experts in the relevant parts. However, it is virtually impossible to establish a full set of relationship for each and every design parameter, because they are usually obtained from a wealth of experiments and/or an arduous investigation of complicated constraints imposed on the components.

To augment existing analyses, one promising approach is to leverage historical change records, which are available from the modern management systems such as PDMS (Product Data Management System) or CVS (Concurrent Version System). The change record includes the set of system components that frequently changed together in the past. Existing works in the literature apply several data-mining tools to extract co-change patterns from the change records and to predict future changes.

Bayesian Network (BN) is a graphical formalism for representing uncertain knowledge in probabilistic system (Nielsen and Jensen, 2009). BN represents the probabilistic relationship between large numbers of random variables with both graph and conditional probability distributions (Kjaerulff and Madsen, 2008). The cause–effect analysis of change coupling also coincides with the framework of BN. In this regard, several works proposed BN-based models to predict change propagation with probabilistic measure (Mirarab et al., 2007; Lee and Hong, 2017; Zhou et al., 2008). In these studies, system components are represented with the set of random variables, and the uncertain coupling between components are represented with the conditional probability distribution. After BN is constructed, probabilistic inference algorithm is deployed to infer the probability of further changes caused by the change of other parts in the system. The main advantage of BN-based approach is that it can provide a probabilistic viewpoint for modelling

[*] Corresponding author.
E-mail addresses: jihwan@pknu.ac.kr (J. Lee), yhong@snu.ac.kr (Y.S. Hong).

https://doi.org/10.1016/j.engappai.2018.02.001
Received 22 March 2017; Received in revised form 18 January 2018; Accepted 1 February 2018

uncertain coupling between components. Thus, compared with other approaches such as association rule mining or sequential mining that divide the elements into binary classes (whether change or not), BN can provide a quantitative measure to reflect the probability that an element will change.

However, BN also has several limitations in modelling change propagation. First, BN is computationally expensive. Since the learning and inference algorithm of BN are known as NP-complete problem (Cooper, 1990), its application to large-scaled complex systems may be limited. Second, BN has a limitation in modelling dependency between system elements. For example, it is not possible with BN to represent the mutual dependency between system element. Thus, if there is an edge from component A to component B ($A \rightarrow B$), then the edge with an opposite direction ($B \rightarrow A$) cannot be represented with BN. Moreover, cycle structure is not allowed in BN.

To overcome these limitations, this article proposes an alternative probabilistic graphical model, Dependency Network (DN). DN is an alternative version of BN which approximates the full joint probability distribution over a set of random variables (Heckerman et al., 2001). DN provides a straightforward and computationally efficient algorithm for both learning and probabilistic inference, which make it applicable to large-scaled system. Also, it is possible with DN to represent mutual dependency or cycles among system elements. Thus, DN can overcome the limitation of BN in modelling dependency between system elements. This article proposes a systematic framework for modelling change propagation using DN. The procedure for learning from historical change records as well as inferring change propagation probabilities is depicted. To show the feasibility of our model, a case study conducted with empirical data obtained from the historical change records of Azureus, which is an open-sourced software (Azureus, 2016). To validate the effectiveness of the proposed method, several experiments incorporating different parameters were conducted.

The reminder of this paper is organized as follows. Section 2 introduces the literature related to prediction of change propagation. Section 3 motivates our approach throughout the problem definition, and also the formal definition of BN and DN is illustrated. Section 4 describes our approach. Section 5 presents the case study of Azureus. Section 6 presents the validation of our approach. Finally, Section 7 draws conclusions and anticipates future work.

2. Literature review

Several change-propagation prediction methods have been proposed by the literature. They are classified by the approach by which the dependency information is obtained. In product design domain, the expert-based method has been widely adopted. The expert-based methods utilize the subjective opinions of experts to measure inter-component dependency. Cohen et al. (2000), proposes an C-FAR (change favourable representation) as a model for the dependency representation. In C-FAR, expert opinion on attribute interaction is then measured on a qualitative scale (high, medium, and low), after which the result is translated to matrix form. This matrix is then used to calculate the consequence of a change of a source entity to a target entity. Clarkson et al. (2004) proposed an alternative, quantitative index for prediction of change propagation. Utilizing Design Structure Matrix (DSM), a square matrix representing inter-component interaction, they quantify both the likelihoods and impacts of changes between adjacent components with numerical values between 0 and 1. Parameter-based methods utilizes a predefined mathematical function between components. For example, Yang and Duan (2012) proposed the use of the mathematical constraint relationship between design parameters as a measure of inter-component dependency, which relationship is then used to find the optimal design path maximally mitigating the risk of change propagation. Hamraz et al. (2013) utilized a change propagation probability distribution obtained with reference to the tolerance ranges identified between subsystems.

In software systems, there has been extensive body of works that utilizes the historical change records in identification of source code dependencies. Several data-mining techniques has been applied to predict the change propagation by mining historical change logs. Gall et al. (1998) tried to develop a rule-based model that identifies the logical coupling between classes, files and functions with the source code release information. Mockus and Weiss (2000) and Silva et al. (2014) propose a clustering-based methodology to identify the group of source codes that have changed together, which is called as chunks. Hassan and Holt (2004) proposes several heuristics for identifying source code dependencies. They also proposed several metrics that can be used to evaluate the change propagation. Zimmermann et al. (2005) proposes an association rule mining approach. In these approaches, several association rules among source files were extracted from the historical change logs of the CVS, and the mined patterns were used in prediction of changes given queries. Similarly, Ying et al. (2004) utilized frequent pattern mining algorithm in prediction of change propagation. Finlay et al. (2014) proposes decision tree-based model to predict whether the specific code will change or not. Sun et al. (2012) proposes to use Formal Concept Analysis (FCA) to develop the concept lattice of changes which represents the hierarchical order of software changes.

Time series analysis have been applied to capture the interdependencies between multiple time series of changes. For example, Ceccarelli et al. (2010) applies multivariate time series analysis to test whether past changes of a software element are statistically related to changes of the other software element. Similarly, Canfora et al. (2010) proposes to combine association rule-based technique with multi-variate time series in predicting changes.

Information Retrieval (IR) techniques also has been applied to examine the word usage patterns of source codes and to identify the conceptual coupling among source files. For example, by analysing identifiers and comments used in source codes, Poshyvanyk et al. (2009) measured the conceptual coupling between different software classes. Kagdi et al. (2010) combined Latent Semantic Indexing (LSI) with association rule mining to examine the conceptual coupling as well as evolutionally coupling between software elements. Among IR techniques, topic model, a generative probabilistic model to detect several topics from documents, was also widely adopted to detect cohesion groups of software elements. Hindle et al. (2009) and Gethers and Poshyvanyk (2010) propose Latent Dirichlet Allocation (LDA) in grouping classes by measuring the conceptual cohesion of classes. Linstead et al. (2008) and Thomas et al. (2014) also utilizes LDA in examine the topic evolution of class groups.

Recently, probability theory was applied to quantify the impact of change propagation with probability measure. Tsantalis et al. (2005) proposed the probability measure of change propagation based on the structural relations between software elements. Wong and Cai (2011) proposed a Markov chain model to model the cascading effect of change propagation between software elements. Similarly, Ferreira et al. (2017) proposed stochastic process of change propagation to estimate the number of change steps during a modification process.

Bayesian Network (BN) is one of the probabilistic model that was used to estimate the probability of change propagation. Abdi et al. (2009), proposed a Bayesian network with structural coupling metric that is obtained from assessment of the expert's opinion. Lee and Hong (2017) proposed a Dynamic Bayesian Network (DBN) model that addresses the cascading effect between system elements. However, their work does not provide an algorithm for learning graph structure and parameters from the dataset. In Mirarab et al. (2007), BN was used to learn both graph structure and parameters of BN from the historical change data. However, as previously noted, BN is not scalable to large and complex system as well as it cannot represent the cyclic dependency between elements. This paper proposes a Dependency Network to overcome the limitation of BN in predicting change propagation.