Information Systems 000 (2017) 1–14

Contents lists available at ScienceDirect

Information Systems

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

# Root cause analysis in IT infrastructures using ontologies and abduction in Markov Logic Networks☆

Joerg Schoenfisch[a,*], Christian Meilicke[a], Janno von Stülpnagel[b], Jens Ortmann[b], Heiner Stuckenschmidt[a]

[a] Research Group Data and Web Science, University of Mannheim, Germany
[b] Softplant GmbH, Munich, Germany

ARTICLE INFO

*Article history:*
Received 15 December 2016
Revised 10 November 2017
Accepted 12 November 2017
Available online xxx

*Keywords:*
Root cause analysis
IT Infrastructure management
Markov Logic Network
Ontology
Abductive reasoning

ABSTRACT

Information systems play a crucial role in most of today's business operations. High availability and reliability of services and hardware, and, in the case of outages, short response times are essential. Thus, a high amount of tool support and automation in risk management is desirable to decrease downtime.

We propose a new approach for calculating the root cause for an observed failure in an IT infrastructure. Our approach is based on abduction in Markov Logic Networks. Abduction aims to find an explanation for a given observation in the light of some background knowledge. In failure diagnosis, the explanation corresponds to the root cause, the observation to the failure of a component, and the background knowledge to the dependency graph extended by potential risks. We apply a method to extend a Markov Logic Network in order to conduct abductive reasoning, which is not naturally supported in this formalism.

Our approach exhibits a high amount of reusability and facilitates modeling by using ontologies as background knowledge. This enables users without specific knowledge of a concrete infrastructure to gain viable insights in the case of an incident. We implemented the method in a tool and illustrate its suitability for root cause analysis by applying it to a sample scenario and testing its scalability on randomly generated infrastructures.



## 1. Introduction

Root cause analysis (RCA) plays an important part in processes for problem solving in many different settings. Its purpose is to find the underlying source of the observed symptoms of a problem. IT plays an important role in processes in a wide area of business, thus a high availability and short response times to failures (e.g., failing e-mail deliveries, inaccessible websites, or unresponsive accounting systems) are crucial [2]. Today's IT infrastructures are getting increasingly complex with diverse direct and transitive dependencies. This makes root cause analysis a time intensive task as the cause for a problem might be unclear or the most probable cause might not be the most obvious one. Therefore, automating the process of root cause analysis and helping an IT administrator to identify the source of a failure or outage as fast as possible is important to achieve a high service level [3].

In this paper we present our approach to root cause analysis that uses Markov Logic Networks (MLN) and abductive reasoning to enable an engineer to drill down fast on the source of a problem. Markov Logic Networks provide a formalism that combines logical formulas (to describe dependencies) and probabilities (to express various possible risks) in a single representation. We focus on abductive reasoning in MLNs and show how it can be used for the purpose of root cause analysis. To our knowledge, the proposed approach is a novel method to root cause analysis that combines probabilistic and logical aspects in a well-founded framework.

Throughout the paper, we illustrate our approach on a small case study. The IT infrastructure in our settings is comprised of a multifunction office printer that offers – amongst others – printing and scanning services via a network. These services use a mail and

☆ This work has been partially supported by the German Federal Ministry of Economics and Technology (BMWI grant no. KF3099601KM2) in the framework of the Central Innovation Program SME (Zentrales Innovationsprogramm Mittelstand - ZIM) within the project "Risk management tool for complex IT infrastructures".
The most notable extensions compared to the work presented in [1] is the incorporation of ontologies in the modeling step, and the presentation of scalability results on specifically generated datasets.

* Corresponding author.
*E-mail addresses:* joerg@informatik.uni-mannheim.de (J. Schoenfisch), christian@informatik.uni-mannheim.de (C. Meilicke), janno.stuelpnagel@softplant.de (J.v. Stülpnagel), jens.ortmann@softplant.de (J. Ortmann), heiner@informatik.uni-mannheim.de (H. Stuckenschmidt).

https://doi.org/10.1016/j.is.2017.11.003
0306-4379/© 2017 Published by Elsevier Ltd.

indirectly an LDAP service. Everything is dependent on the network and the power supply. This small case study already has dependencies that cross several different levels of infrastructure (services, server hardware, network hardware, power supply). We will expand this setting with possible causes for failure and probabilities for their occurrence. Theses risks are described in the "IT-Grundschutz Catalogue" by the German "Bundesamt für Sicherheit in der Informationstechnik" (Federal Office for Information Security) which is based on the ISO 27001 certification.[1] Furthermore, we evaluate the scalability of the approach on infrastructures generated randomly based on the structure observed in real-world environments.

Within our framework, the IT infrastructure is represented as a logical dependency network that includes various threats to its components. When a problem occurs, available observations are entered into the system which then generates the Markov Logic Network from the available observations, the given dependency network, and the general background knowledge related to the components of the infrastructure. Some of these observations might be specified manually, while other observations can be entered into the system automatically, e.g. via constantly running monitoring software. These observations are typically incomplete in the sense that not all relevant components are monitored, or not all problems are recognized. Thus, taking the given observations into account, there might still be a set of several explanations for the problem that occurred. Under the assumption that the modeled dependency network captures all relations present in the infrastructure and all threats are adequately taken into account, the correct explanation will always be contained in that set.

We calculate, via abduction, the most probable cause for the current problem, which is then presented to the user, e.g., the administrator of the IT infrastructure. The user can then investigate if it is indeed the source of the problem. This might require to manually check the availability of some component or to analyze a log file. If the proposed explanation is correct, counter-measures can be introduced immediately. If the additional observations revealed that the calculated explanation is wrong, those new observations are entered into the system as additional evidence and a better explanation is computed. This iterative, dialog-based process is a practicable approach to quickly narrow down on a root cause.

In our approach, we represent the given infrastructure and the possible risks as ontology. This allows us to automatically infer that certain threats are relevant for certain infrastructure components, or add logical constraints ensuring consistency. Relevant background knowledge can easily be maintained and used to generate the Markov Logic Network. Moreover, our approach can take into account known probabilities of risks and failures. These probabilities are derived from expert judgment or statistical data. Instead of computing multiple candidate explanations, which is possible in purely logic based approaches, we are able to generate the most probable explanation with our approach, while still leveraging the full power of an expressive, declarative framework.

This paper is structured as follows. First, we present the theoretical underpinnings of our approach. In Section 2, we give a brief description to Markov Logic, introduce the general notion of abduction, and explain how abduction can be realized in the context of Markov Logic Networks. Furthermore, we give a short introduction to ontologies and their benefit in modeling IT infrastructures. In Section 3, we first present a typical scenario for root cause analysis. Then, we show how to model this scenario in our framework and describe how to apply abductive reasoning to find the most probable root cause. We present a workflow that illustrates how our approach is used in the context of a dialog-based process in Section 4. Furthermore, we conduct the evaluation of the scalability of the approach in Section 5. A tool we implemented to support the user in modeling the infrastructure and running a root cause analysis is presented in Section 6. In Section 7, we show how our approach is related to other works. Finally, we discuss the drawbacks and benefits of our approach, and point out directions for future work in Section 8.

[1] https://www.bsi.bund.de/EN/Topics/ITGrundschutz/itgrundschutz_node.html.

## 2. Theoretical background

This section first describes First-Order Logic and Markov Logic Networks. Then, we explain abduction and its concrete implementation in the context of Markov Logic Networks.

### 2.1. First-order logic

First-order logic (FOL) is used do describe and reason in a domain of discourse. Syntactically, it consists of: *constants* $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$, *variables* $\mathcal{V} = \{v_1, \ldots, v_{|\mathcal{V}|}\}$, *predicates* $\mathcal{R} = \{r_1, \ldots, r_{|\mathcal{R}|}\}$, *functions* $\mathcal{F} = \{f_1, \ldots, f_{|\mathcal{F}|}\}$, and *logical operators* ($\forall$, $\exists$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\neg$, (, ), $\equiv$). A *term* $t$ can either be a variable $v$, a constant $c$, or a function of terms $f(t_i, \ldots, t_j)$. An *atomic formula* (or simply atom) is a formula that contains no logical connective, i.e. it consists of a single predicate. A general formula is created by connecting multiple atoms. If an atom contains no variables we call it a *ground atom*; if a formula contains no free variables (i.e. only constants and variables bound by $\forall$ or $\exists$), we call it a *ground formula*.

The semantics of a first-order logic is given by an *interpretation*. Intuitively, an interpretation assigns real-world objects present in the domain to the syntactic constructs described above. This way, every term is also assigned a truth value, e.g. an atomic formula is true if a relation as identified by the predicate exists (or can exist) between the specified constants and variables.

A small example FOL model is the following simple description of relations between persons and their hobbies:

$$\begin{gathered} \text{person}(Alice) \quad \text{person}(Bob) \quad \text{person}(Eve) \\ \text{friends}(Alice, Bob) \\ \text{hasHobby}(Alice, Football) \\ \text{friends}(x, y) \wedge \text{hasHobby}(x, z) \rightarrow \text{hasHobby}(y, z) \end{gathered} \tag{1}$$

The model contains the constants *Alice, Bob* and *Eve*, variables *x, y* and *z*, the predicates *friends* and *hasHobby* and the logical operators $\wedge$ and $\rightarrow$. It states that *Alice* and *Bob* are both persons and friends, *Eve* is a person, *Alice* has *Football* as a hobby, and that individuals who are friends have the same hobbies. From this model it can be inferred that *Bob* also has the hobby *Football*.

### 2.2. Markov Logic Networks

Markov Logic Networks (MLN) generalize first-order logic and probabilistic graphical models by allowing hard and soft first-order formulas [4]. Hard formulas are regular first-order formulas, which have to be fulfilled by every interpretation. An interpretation is also referred to as a possible world. Soft formulas have weights that support (in case of positive weights) or penalize (in case of negative weights) worlds in which they are satisfied. The probability of a possible world, one that satisfies all hard formulas, is proportional to the exponential sum of the weights of the soft formulas that are satisfied in that world. This corresponds to the common understanding of Markov Networks as log-linear probabilistic models [4].

MLNs are a template for constructing Markov Networks. A formula is called a grounded formula if all variables have been replaced by constants. Given a set of constants, a Markov Network