Reliability Engineering and System Safety 121 (2014) 61–71

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

A classification-based approach to monitoring the safety of dynamic systems

Shengtong Zhong [a], Helge Langseth [a,*], Thomas Dyhre Nielsen [b]

[a] Department of Computer and Information Science, The Norwegian University of Science and Technology, Trondheim, Norway
[b] Department of Computer Science, Aalborg University, Aalborg, Denmark

ARTICLE INFO

Article history:
Received 5 December 2012
Received in revised form
16 May 2013
Accepted 31 July 2013
Available online 8 August 2013

Keywords:
Monitoring
Anomaly detection
Dynamic classification
Hidden Markov models
Approximate inference

ABSTRACT

Monitoring a complex process often involves keeping an eye on hundreds or thousands of sensors to determine whether or not the process is stable. We have been working with dynamic data from an oil production facility in the North sea, where unstable situations should be identified as soon as possible. Motivated by this problem setting, we propose a general model for classification in dynamic domains, and exemplify its use by showing how it can be employed for *activity detection*. We construct our model by using well known statistical techniques as building-blocks, and evaluate each step in the model-building process empirically. Exact inference in the proposed model is intractable, so in this paper we experiment with an approximate inference scheme.

1. Introduction

A typical task for the risk and reliability engineer is to monitor the status of a dynamic system, like, e.g., a chemical process. Doing so will often mean tending to a large number of sensors, each of them updating their readings on a regular basis. Real-life processes have their own natural dynamics when everything is running according to plan; "outliers" may on the other hand be seen as indications that the process is leaving its stable state, and thereby becoming more dangerous. Thus, the engineer would like to know if the system is unstable in order to ensure that the proper corrective actions are implemented as soon as the system becomes unsafe. Unfortunately, it may be difficult to measure the status of the system directly, and one will typically only have access to indirect status indicators, which need to be analyzed and combined in a statistical model. Formally, detecting the instantaneous status of a system described by a collection $\boldsymbol{Y} = \{Y_1, Y_2, \ldots, Y_n\}$ of random variables is identical to *classification*, where an object described by a value assignment $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$ is mapped to one of a set of possible labels (or classes). The labels for an object are represented by a class variable C, and are denoted $sp(C)$. We will focus on real-valued attributes in this paper, meaning that $\boldsymbol{y} \in \mathbb{R}^n$. In a probabilistic framework, it is well-known that the optimal classifier will label an object $\boldsymbol{y}$ by the class label $\hat{c}$, where

$$\hat{c} = \arg \min_{c \in sp(C)} \sum_{c' \in sp(C)} L(c, c')P(c' | \boldsymbol{y}) \qquad (1)$$

and $L(c, c')$ is the loss-function encoding the cost of misclassification. Learning a classifier therefore amounts to estimating the probability distribution $P(C = c | \boldsymbol{y})$.

The engineer may not only want to assess the *instantaneous* status of a system, but rather to detect if the system is *about to become* unstable (that is, to predict future problems). This would give a system operator the chance to implement countermeasures before anyone is exposed to an increased level of risk. Classifiers that fail to take the dynamic aspect of a process into account will not be able to make accurate predictions, and will therefore not be able to recognize a problem under development. In *dynamic classification*, the task is to assign a class label to an object at each time step. To support the classification, objects are characterized by a new observation at each time step as well. We use $\boldsymbol{Y}^t = \{Y_1^t, Y_2^t, \ldots, Y_n^t\}$ to denote the random variables describing the object at time t, where $\boldsymbol{y}^t = \{y_1^t, y_2^t, \ldots, y_n^t\}$ is a specific value assignment to these variables. The collected observations from time $t=1$ and up to time t is denoted as $\boldsymbol{y}^{1:t}$. The set of possible labels (or classes) for the time series at time t is represented by a class variable C^t and denoted $sp(C^t)$. With the observations $\boldsymbol{y}^{1:t}$ from time step 1 to t, the optimal classifier will label $\boldsymbol{y}^{1:t}$ by the class label $\hat{c}^t$ at time t, where

$$\hat{c}^t = \arg \min_{c^t \in sp(C^t)} \sum_{c' \in sp(C^t)} L(c^t, c')P(c' | \boldsymbol{y}^{1:t});$$

confer also Eq. (1).

[*] Corresponding author. Tel.: +47 73596488; fax: +47 73594466.
E-mail addresses: shket@idi.ntnu.no (S. Zhong), helgel@idi.ntnu.no (H. Langseth), tdn@cs.aau.dk (T.D. Nielsen).

0951-8320/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ress.2013.07.016

In a risk and reliability setting, the desire to build efficient statistical models that are flexible yet easy to understand for domain experts has led to reduced focus on traditional frameworks like fault trees. On the other hand, the Bayesian network (BN) framework [28,17] has received increased attention from the community over the last decade [22], partly because BNs have proven to be an attractive alternative to classical reliability formalisms, see e.g., [33,18]. BNs have also been used extensively for classification [8,21,35].

The dynamic Bayesian network framework [12] supports the specification of dynamic processes, and has already found numerous applications in reliability engineering, see, e.g., [20,27]. A simple instance of this framework is the *hidden Markov model* (HMM), which has also been considered for classification purposes [15,31,6]; to this end the "hidden" node in the HMM is used as the classification node, and the attributes at time t are assumed to be independent of those at time $t+1$ given the class label at either of the two points in time. Further simplification can be obtained by assuming that all attributes at one time step are conditionally independent given the class label at that time step; the resulting model by [26] is known as a dynamic naïve Bayes (dNB) classifier. The dNB models can be efficiently estimated from data due to the relatively small number of parameters required to specify them.

To the best of our knowledge, there has been no systematic investigation into the properties of probabilistic classifiers and their applicability to real-life dynamic data. In this paper we will take a step in that direction by examining the underlying assumptions of some well-known probabilistic classifiers and their natural extensions to dynamic domains. We do so by carefully linking our analysis back to a real-life dataset, and the result of this analysis is a classification model, which can be used to, e.g., help prevent unwanted events by automatically analyzing a data stream and raise an alert if the process is entering an unstable state. For the discussions to be concrete, we will tie the model development to the task of *activity recognition* in offshore oil drilling; this is further described in Section 2. In Section 3 we give a general overview of the dynamic classification scheme, and we also propose a specific classification model called a *dynamic latent classification model* (abbreviated to dLCM). Next, we look at inference and learning in dLCMs (Section 4), before reporting on their classification accuracy in Section 5. Finally, in Section 6 we conclude and give directions for future research.

2. The domain and the dataset

Offshore oil drilling is a complex process, potentially with major risks to the safety of the operators involved (see, e.g., [34]). Further, the drilling process in itself is extremely expensive, leading to a focus on cost efficient operation, including high demands wrt. the reliability of the equipment employed. This has again resulted in a plethora of data being collected – either for real-time analysis of the state of the ongoing operations or to enable investigations after an event has occurred. We will consider one such dataset from an oil production installation in the North Sea. Data, consisting of 62 variables, is captured every 5 s. The data is monitored in real time by experienced engineers, who have a number of tasks to perform ranging from understanding the situation on the platform, in order to avoid a number of either dangerous or costly situations, to optimization of the drilling operation. The variables that are collected in this dataset cover measurements taken both topside (like flow rates) and down-hole (like, for instance, the gamma rate).

The overall drilling process can be broken down into a series of *activities* that are performed iteratively as the depth of the well increases. Recognizing which activity is performed at a given point in time is called *activity recognition*, and is the focus of the present paper. Out of the 62 attributes that are collected, domain experts have selected the following 9 attributes as the most important for activity detection: *Depth Bit Measured*, *Depth Hole Measured*, *Block Position*, *Hookload*, *Weight On Bit*, *Revolutions Per Minute*, *Torque*, *Mud Flow In*, and *Standpipe Pressure*.

In the *Wellsite Information Transfer Specification* (WITS), a total of 34 different activities with associated activity codes are defined. Each activity has its separate purpose and consists of a set of actions. Out of the 34 different drilling activities in total, only a handful are really important to recognize. The important activities in our analysis, which roughly correspond to those that constitute most of the total well drilling time, are described next

WITS2 – *Drilling:* The activity occurs when the well is gaining depth by crushing rock at the bottom of the hole and removing the crushed pieces (*cuttings*) out of the wellbore. Thus, the drill string is rotating during this activity, and mud is circulated at low speed to transport out the cuttings. The activity is interrupted by other activities, but continues until the well reaches the reservoir and production of oil may commence.

WITS3 – *Connection*: This activity involves changing the length of the drill-string, by either adding or removing pieces of drill-pipe.

WITS8 – *Tripping in*: This is the act of running the drill string into the well hole.

WITS9 – *Tripping out*: Tripping out means pulling the drill string out of the well bore.

It what follows, the remaining activities will collectively be grouped under the label *Others*.

Knowing which activity is performed at any point in time is important in several contexts: firstly, the operation of an offshore installation can be monitored by groups of experts located elsewhere (typically in on-shore control-rooms). These experts are shielded from the offshore-operation in that they only observe visualizations of streams of data. Important aggregations, like which activity is performed, helps them better understand the situation on-site.

Secondly, operators are consistently looking for more cost-efficient ways of drilling, and the sequencing of activities during an operation is important for hunting down potential time-sinks.

Thirdly, some undesired events can only happen during specific activities, and knowing the current activity is therefore of high importance. For instance, apparent early warnings of undesired events can be given more credence if that event can actually occur during the current activity and no weight if the event is impossible. From a safety perspective, this allows for a better early warning system with a lower rate of false alarms.

Finally, it is worth mentioning that activity recognition is a task that also finds applications in areas as diverse as health care [32] and video analysis [23]. In this paper we develop a model for dynamic classification and exemplify the process in the oil drilling domain, but other safety and reliability applications of the developed model are readily available.

3. From static to dynamic Bayesian classifiers

In this section we develop a general framework for performing dynamic classification. The framework will be specified incrementally by examining its expressivity relative to the oil production data. In Section 5 we further justify the framework by setting up an empirical study using the oil production data. In the study we analyze the accuracy results for the sequence of models that are