Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 72 (2018) 1057–1062

www.elsevier.com/locate/procedia

51st CIRP Conference on Manufacturing Systems

Root cause analysis of failures and quality deviations in manufacturing using machine learning

Anna Lokrantz[a,*], Emil Gustavsson[a], Mats Jirstrand[a]

[a] *Systems and Data Analysis, Fraunhofer-Chalmers Research Centre for Industrial Mathematics, Chalmers Science Park, SE-412 88, Gothenburg, Sweden*

* Corresponding author.Tel. +46 31 772 40 00; fax: +46 31 772 42 60 *E-mail address:* anna.lokrantz@fcc.chalmers.se

Abstract

Today root causes of failures and quality deviations in manufacturing are usually identified using existing on-site expert knowledge about causal relationships between process steps and the nature of failures and deviations. Automatization of identification and back tracking of root causes for said failures and deviations would benefit companies both in that knowledge can be transferred between factories and that knowledge will be preserved for future use. We propose a machine learning framework using Bayesian networks to model the causal relationships between manufacturing stages using expert knowledge, and demonstrate the usefulness of the framework on two simulated manufacturing processes.

Keywords: Root cause analysis; machine learning; probabilistic graphical models; Bayesian networks

1. Introduction

Quality deviations and failures during manufacturing contribute to large losses in both resources, time, and money [1,2]. Traditionally, root causes of failures and quality deviations in manufacturing have been identified using expert knowledge present at each separate manufacturing site. The experience of the staff at the sites constitutes the basis for describing the causal correlations between different process steps and the output failures or quality deviations, and manual methods are then employed to identify the root causes for the failures. This means that valuable knowledge in general cannot always be transferred between neither people at the site nor between different sites. Another factor to take into account is that the experts might actually be biased in their assessments and therefore inaccurate in their conclusions.

Advances in machine learning and big data analytics have opened the possibilities to create models which can learn the causal relations in production lines and make purely data-driven assessments of the root causes of failures. As more components in the manufacturing sites gets connected, more data can be gathered and utilized for such model development. With a data-driven understanding of the causal relations in the productions lines, one will not rely solely on the expert knowledge and the knowledge can be also be stored for future use, and shared between different manufacturing sites.

The aim of this paper is to describe a developed framework where machine learning techniques such as probabilistic graphical models and Bayesian networks are utilized for learning the causal relations in manufacturing processes. We demonstrate how the framework can be utilized for creating Bayesian network descriptions of manufacturing processes and how to train the models and perform inference regarding the root causes of failures. The main objective of the paper is not to present results for a real world manufacturing site, but to analyze concepts and ideas on simulated processes in order to demonstrate how machine learning and probabilistic graphical models can be utilized for deducing root causes of quality deviations and failures in manufacturing.

The remainder of this paper is organized as follows. Section 2 gives an introduction to some basic concepts regarding root cause analysis is included together with a thorough description of probabilistic graphical models and Bayesian networks. Section 3 describes the developed machine learning framework for root cause analysis in manufacturing and illustrates a simple simulated production line. In Section 4 some computational results regarding how hard probabilistic graphical models are to train and the importance of the inclusion of existing expert knowledge is presented, and in Section 5 we conclude with a discussion regarding applications, extensions, and future work of our framework.

2. Background

In this section some introductory theory regarding the central concepts for the paper is presented. In Section 2.1 an introduction to root cause analysis is included, and in Section 2.2 the basic concepts of machine learning, probabilistic graphical

2212-8271 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 51st CIRP Conference on Manufacturing Systems.
10.1016/j.procir.2018.03.229

models, and Bayesian networks are explained.

2.1. Root Cause Analysis

Root cause analysis (RCA) in the context of manufacturing is the process of identifying (causal) factors that cause errors or quality deviations in the manufactured product. A root cause is the most primary reason for, for example, an unwanted quality deviation in a final product or a decrease in overall equipment effectiveness (OEE) for a machine. In many cases, a problem can be solved in the short term by resolving either the symptom or the apparent immediate cause, but it will later reoccur because the treatment was not implemented to address the real root cause. In RCA one challenging issue is the problem of distinguishing between what is a symptom and what is the real root cause. The traditional approach to RCA is to utilize expert knowledge to define the causal factors existing in a manufacturing line. Equipment experts will recognize errors in production or particular deviations in quality and, by using their expert knowledge, draw conclusions on the cause of the problem. This approach has several drawbacks; (i) the expert knowledge can in many cases not be easily transferred between people, (ii) the expert knowledge may not be saved, stored, and transferred to a future workforce, and (iii) there is no guarantee that the knowledge the experts possess is accurate.

Methods for automatic root cause analysis and failure diagnostics have been previously developed and utilized within software and server diagnostics [3]. In these applications the data is already digital, easily accessible, and of large size, so machine learning methods are suitable for the task. These methods have also been extended to be applicable for more analog areas such as the chemical processing industry, [4] and [5]. In production and manufacturing some attempts at implementing machine learning methods for root cause analysis have been performed in [6] and [7].

2.2. Machine learning, Probabilistic Graphical Models, and Bayesian Networks

Machine learning is a subfield of artificial intelligence concerned with the study and development of algorithms for computers to learn from data. Essentially the objective is to train a data driven model from which predictions and decisions can be made. Strong ties exist to other fields, notably mathematical optimization and statistics. Machine learning problems arise in many areas and there are a variety of approaches to these, and among them something called *Probabilistic graphical models*, which is the main scope of this paragraph. For a more thorough introductions to machine learning, see [8] and [9].

Probabilistic graphical models are probabilistic models for which a graph expresses the conditional dependence structure between random variables. A graph based representation enables finding structure in distributions and describe them compactly. In this setting the random variables are modeled as nodes in a graph and the direct probabilistic interactions between these variables are represented by the edges. There are two main families of graphical representations of distributions, *Bayesian networks* that use directed, acyclic graphs and the more general framework *Markov random fields* where undirected graphs are utilized. The model represents a factorization of the joint probability distribution of all variables. If the nodes are $X_1, ..., X_n$ then the joint probability distribution is

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i|p_i) \quad (1)$$

for the set of parent nodes p_i of X_i.

Since we are considering modelling the dependencies between variables in a production or manufacturing process, there exists a clear causal effect between the variables (process steps), so we will focus on the Bayesian network representation. To describe the general concepts of Bayesian networks, we first introduce a simple text book example of a Bayesian network representation in Figure 1. The example in Figure 1 describes a Bayesian network consisting of three binary nodes (Rain, Sprinkler, Wet grass). The edges in the graph described the conditional dependencies between the variables. For example, the probability that the grass is wet depends both on information regarding if it is raining or not, and if the sprinkler is on or not. The probability that the sprinkler is on depends on whether or not it is raining.

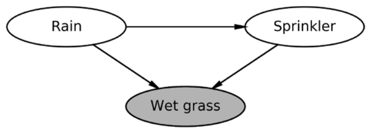


Fig. 1: Example of a simple Bayesian network consisting of three nodes (Rain, Sprinkler, Wet grass.

One often distinguish between two types of nodes in a Bayesian network:

- *Observed variables* are variables for which we can directly observed data. In a manufacturing process the observed variables are either measurements taken from the process or quality deviations and failures detected from the output.
- *Latent variables* or hidden variables are variables which can not be directly measured or observed. In our setting the latent variables represents the failure modes of the process, i.e., the variables we want to infer something about.

A Bayesian network is fully described by the directed edges between the variables (the structure of the network) and the conditional probability distributions of the dependent variables (the parameters of the network). In a manufacturing process the structure of the network describes which causes or failures in the process affects other failures and if they affect the quality of the product or not. The parameters of the network describes how strong the correlations are between the failure nodes and the quality deviations of the products. In general, there are three main tasks associated with fitting and performing inference on Bayesian networks:

- *Structure learning:* It might often be that one does not know the structure of the network, or that only a few edges are known thanks to prior knowledge about dependencies. The aim of structure learning is to deduce the structure of the network from the independencies and dependencies in the data, aided by any prior knowledge as constraints. Examples of algorithms include *constraint based learning*, *score based learning* and *Bayesian averaging*[10,11].

- *Parameter learning:* Assuming the structure of the Bayesian network, learning the parameters of a Bayesian network constitutes of learning the conditional probability distributions associated with each node in the network from recorded data. There are two methods for parameter learning: *Maximum likelihood* and *Bayesian parameter estimation* [10,11].
- *Inference:* The most common query type one would make in a Bayesian network or a graphical model is a conditional probability query, that is the probability of some subset of variables in the network given some evidence in the form of the instantiation of some (other) subset of random variables. One should note that the inference problem is computationally difficult even for simple inference tasks. So for a large number of models exact inference is intractable, and approximations are needed. Algorithms for inference include *Variable elimination*, which is an algorithm for exact inference, and *Belief propagation*, which is an algorithm for approximate inference [10,11].

For more thorough descriptions of probabilistic graphical models and Bayesian networks, see [10,11], and [12,13], respectively. Examples of applications where Bayesian networks and Bayesian inference is utilized for root cause analysis can be found in [14–16].

3. RCA framework

In the following section we describe the developed framework where Bayesian networks are utilized for constructing models which can perform inference and draw conclusions regarding the root causes of quality deviations. A manufacturing process can in many cases be described as a sequence of consecutive process steps that are performed on some input in order to produce an output. In Figure 2 a schematic description of a manufacturing line consisting of six process steps (P1-P6) is illustrated. At the output of the process four different failures/quality deviations may be detected (A-D). These failures are caused by properties and characteristics of the preceding process steps (P1-P6). The causal correlations, i.e., how the process steps affect each other and how these are correlated with the quality deviations at the output, are represented by the dashed arrows. For example, the quality deviation D may stem from a property of process step P6 as the causal correlation suggests, but it may also stem from process steps earlier in the manufacturing process since these affect the properties of P6.

The natural division of a manufacturing process into consecutive steps lets us define networks that are layered, i.e., where the nodes in the graph can be divided into separate layers corresponding to the different process steps. In each layer, except the output layer, there exist so called *causal nodes* and/or *measurement nodes*. The causal nodes represent causes of faults and quality deviations in the production process, and the measurement nodes represent observable information one can obtain from measurements performed at the different steps, for example, vibration measurements of the machine in a process step. The output layer consists of *failure nodes* representing quality deviations or failures of the finished product. In general, the measurement nodes and the failure nodes are observable variables, while the causal nodes are latent/hidden.

The developed framework for root cause analysis is implemented in Python and utilized the library `pgmpy`, see [17], with which it is possible to build Bayesian networks. Below is a pseudo-code example of how to initialize a model consisting of two process steps (`P1` and `P2`) with two and one causal nodes, respectively, and one output step (`output`) with tree possible quality deviations/failures. In each process a measurement node is also included.

```
P1     = [causal = C1, C2; measurement = M1]
P2     = [causal = C3; measurement =  M2]
output = [failures = F1, F2, F3]
layers = [P1, P2, output]

model  = RCAmodel(layers)
```

This overall structure of the process needs to be specified by expert knowledge, i.e., one needs to define the different process steps and the corresponding causes that can occur, and also the failures/quality deviations that can happen during the production. Calling `RCAModel` initiates a Bayesian model as with a call to the `pgmpy` class for creating Bayesian networks and adds the nodes in the different layers to the model. If there exists expert knowledge regarding correlations between causal nodes and failures nodes, this can easily be added. In this simple example, assume that from historic data one can deduce that the causal node C1 affects the quality deviation F2, and that the causal node C3 affects F3. Then this type of expert knowledge can easily be added to the model by the following:

```
edges = [C1 -> F2, C3 -> F3]
model.addExpertEdges(edges)
```

Once the model is initialized with or without some expert knowledge, the next step is to infer the structure and parameters of the network from previously recorded data, i.e. data where failures have occured and where the root cause is known. Theses are separate learning tasks, as described in 2.2, and are performed independently. The `pgmpy` library provides various algorithms for structure learning as well as parameter learning and inference. The structure learning creates a suggestion for a model, and the edges are then added to the model if the condition that no loops should exist is fulfilled. Loops would raise an error in the `BayesianModel` class. In this case, constraint based structure learning, Bayesian parameter learning and Variable elimination are the algorithms are default for learning and inference.

```
data = ...
model.structureLearning(data, algorithm =...)
model.parameterLearning(data, algorithm=...)
```

When the model is trained on historic data, it can be used for performing inference on the causal nodes and determining the root causes of new process failures that occur. For example, assume that the quality deviation or failure F3 occur, and we need to infer from the model which causal effect was the most probable. The result of inference is given in the form of conditional probabilities of the desired variables conditioned on the evidence, as exampled by the following pseudo code. In this case only the value of failure nodes are given as evidence.

```
evidence = [F3 is "True"]
```

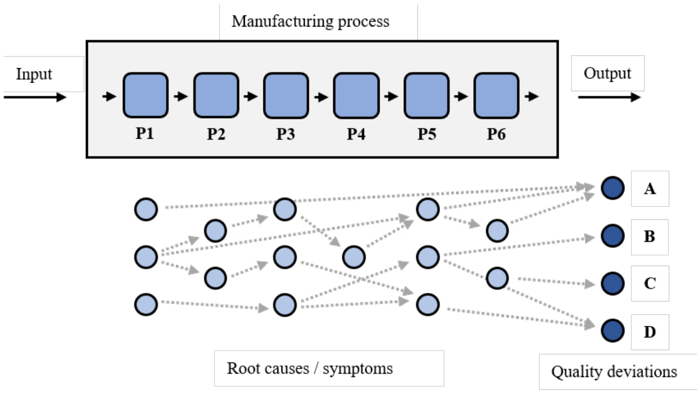


Fig. 2: A schematic illustration of a manufacturing line consisting of six process steps (P1-P6). Four quality deviations/failures (A-D) are illustrated with possible root causes and symptoms that have causal correlations with the quality deviations.

```
model.inference(evidence, algorithm=...)
 >> Probability C1 = 0.01
 >> Probability C2 = 0.04
 >> Probability C3 = 0.76
```

For this simple example, the inference results states that, given the evidence, there is a 76% probability that the causal factor C3 was the cause for the failure. Note that the probabilities do not sum to 1, meaning that there is always the probability that the failure occurred without C1-C3 being the causes of the failure. This can happen when some unmodeled phenomena was the cause of the failures. Since in this simple example we also have access to measurement nodes at each process step, we can utilize this information when performing the inference, i.e., we can give the measurements as evidence for the inference.

```
evidence = [F3 is "True", values of M1, M2]
model.inference(evidence, algorithm =...)
 >> Probability C1 = 0.00
 >> Probability C2 = 0.02
 >> Probability C3 = 0.91
```

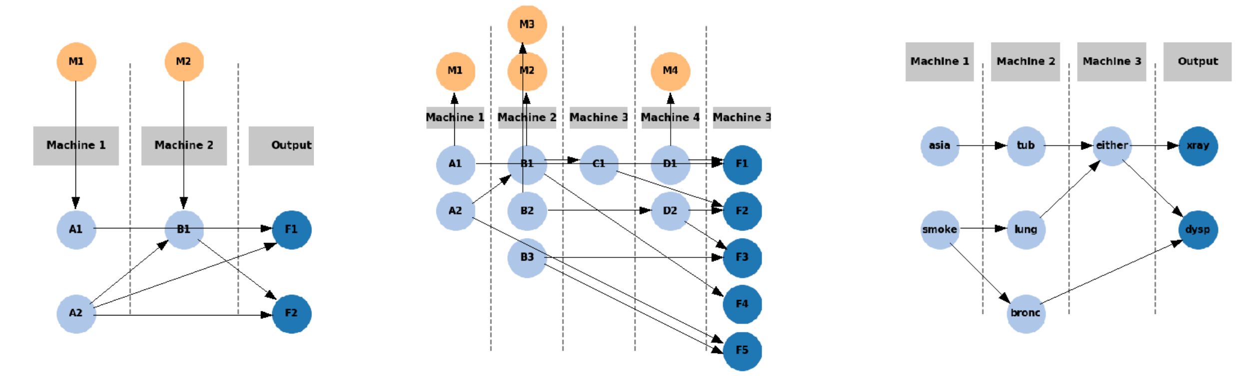
Here we see that including more evidence, that is the measurements at the process steps, gives a stronger inference regarding which causal node was the cause of the failure. Now the probability that C3 was the cause of the failure is increased to 91%. So, by including measurements in the process steps we seem to have captured some more aspects that were not included before and thus improved the accuracy with which we can conclude the cause of this particular failure. This is also in much what is going to be the scope of the succeeding investigation in this paper, how expert knowledge about causal links and the existence of measurement nodes affect the result of inference.

4. Computational results

In the following section some computational results regarding structure learning and inference of the Bayesian networks described by manufacturing processes is presented. We introduce two simulated manufacturing processes consisting of three and five process steps (layers), respectively. The processes are illustrated in Figures 3a and 3b together with the true dependencies between the nodes. The small model has eight direct dependencies and the large model has fifteen direct dependencies. The data we use for analyzing the structure learning, parameter learning, and inference on the models are simulated from the true models where all the dependencies and parameters are known. We limit ourselves to one structure learning algorithm, constraint based structure learning, one parameter learning algorithm, Bayesian parameter learning, and one inference algorithm, variable elimination, as the purpose is more to show the difference between the proposed cases to be considered rather than between different algorithms. Additionally we also consider applying our framework to some benchmark data. In this case from the so called *Asia network*, which is a Bayesian network that describes the process of diagnosing a patient who just came back from at trip to Asia and shows signs of dyspnoea (laboured breathing) [18,19]. In the case of the Asia model, no measurement nodes are considered. The network defined by the Asia model is illustrated in Figure 3c.

4.1. Structure learning

A first analysis is to investigate how good the implemented structure learning methods used in our framework are to find the correct dependencies between the variables (the structure of the network) depending on how much training data they are given. In Figures 4a, 4b, and 4c the average number of correct direct dependencies found by the structure learning algorithm are illustrated for different sizes of training sets given. When expert edges are considered, three links are used for the small model as well as for the Asia model and six for the large. No links to measurement nodes are considered as we want to use the same expert links for versions of the model with and without measurement nodes. The expert links are added to the model prior to the structure learning, then the algorithm deduces a structure from the data and the links that are not the already provided expert links are added to the model. The models are initialized without expert knowledge about the causal links, so no prior information regarding the process is included. From the figures it is apparent that the task of structure learning is very difficult if no expert knowledge is added. For the small model it requires a training set of more than 15,000 data points in order to find six out of seven correct edges of the model. For the large model it requires more than 200,000 data points to find eleven or twelve of the fifteen correct direct dependencies. When expert knowledge is added initially, the stable point is reached much quicker for the smaller model but no additional edges are found. For the



(a) Small network (b) Large network (c) Asia network

Fig. 3: The rca networks and the asia network.

larger model the result improves significantly with an average of fourteen out of sixteen edged found. The same applies to the Asia model, which has eight direct dependencies, as to be expected. To be noted is that these expert edges are added without any knowledge about what edges might be found or not during structure learning, thus improvement depend somewhat on the edges provided.

4.2. Inference

The objective of the Bayesian network approach to root cause analysis is to utilize the models for performing inference, and for deducing root causes of quality deviations and failures. For the small and large process models we analyze three different Bayesian network models:

- *No expert knowledge*: No initial dependencies between the variables are given, and structure learning is first applied to the model, followed by parameter learning.
- *Some expert knowledge*: Some initial direct dependencies between variables are given to the initial model. Structure learning is still performed to find the entire structure, followed by parameter learning.
- *Full model structure*: The entire structure of the model is given initially, and only parameter learning is performed for the model.

The inference error of a model is measured as the average difference (in L2-norm) between the true probability distribution of the correct model and the model found by structure and parameter learning for a test set of values. For each of these models we measure the inference error given two different types of evidences; (F) when only the failure that has occured is given as evidence, and (F+M) when both the failure and the values of the measurement nodes are given as evidence. In Figure 5a and 5b the inference errors of the Bayesian network models are illustrated for the small and large model, respectively. For both the small and the large model one can see that the inference error decreases with the size of the training set. All models where both failure nodes and measurement nodes perform better than their counter parts with only failure nodes known, however in none of the cases is the case of knowing the full model structure and having measurement nodes reached. The difference is less for the smaller of the two models though, not surprisingly.

5. Discussion and future research

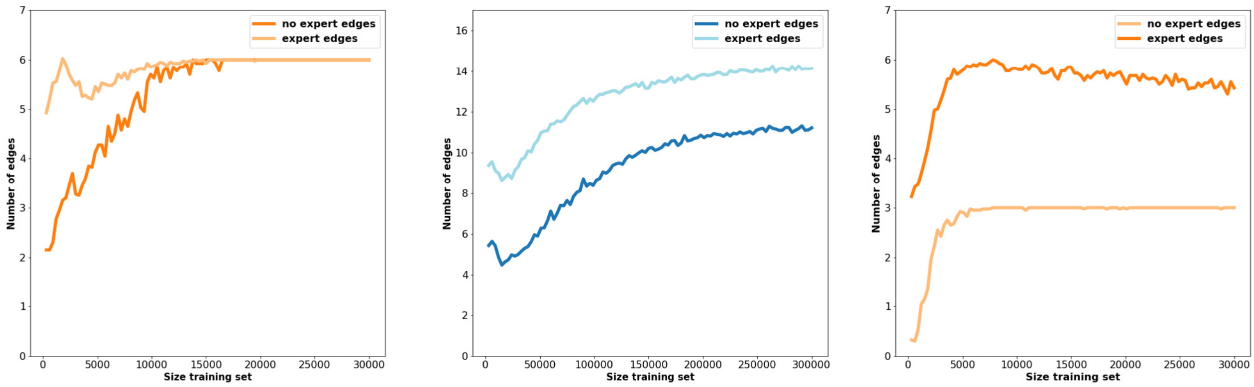
We have in this paper demonstrated how probabilistic graphical models can be used in manufacturing processes for training models where inference can be drawn about root causes of failures or deviations in production.

The main findings are how knowledge from equipment experts can be used to pose a manufacturing process as a Bayesian network and in particular how knowledge about links in this network as well as measurement nodes may improve the model. Structure learning is a difficult task, as shown in Section 4.1 and in particular for a larger model. More knowledge about the structure also improves the accuracy in inference, as does further knowledge about each layer in the form of measurement nodes. However, the combination gives the best result in all cases considered as shown in Section 4.2.

The next steps in the development of the framework will be to try it on a problem where we have continuous data and to use it for real world applications. Other thoughts are to consider a different approach for doing inference, involving training a neural network to perform inference in the RCA network.

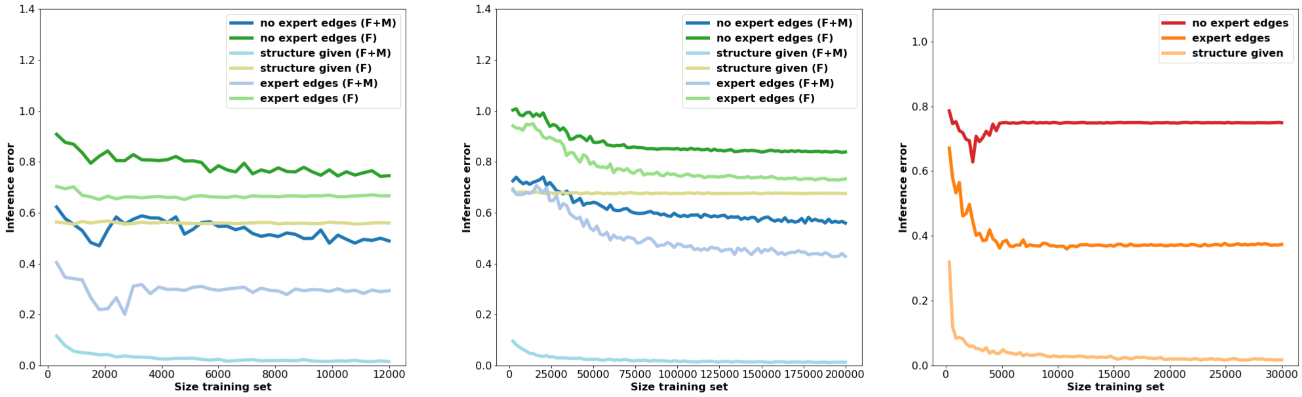
Acknowledgements

This research was supported by the project Root Cause Analysis of Quality Deviations in Manufacturing using Machine Learning (RCA-ML) in the funding program The smart digital factory (DNR 2016-04472), administered by VINNOVA, the Swedish Government Agency for Innovation Systems.



(a) Small model (b) Large model (c) Asia model

Fig. 4: Structure learning with and without expert links.



(a) Small model (b) Large model (c) Asia model

Fig. 5: Structure learning with and without expert links.

References

[1] S. K. Krishnan, A. Agus, N. Husain, Cost of quality: the hidden costs, Total Quality Management 11 (4-6) (2000) 844–848.
[2] J. Campanella, Principles of quality costs: Principles, implementation, and use, in: ASQ World Conference on Quality and Improvement Proceedings, American Society for Quality, 1999, p. 507.
[3] M. Chen, A. X. Zheng, J. Lloyd, M. I. Jordan, E. Brewer, Failure diagnosis using decision trees, in: Autonomic Computing, 2004. Proceedings. International Conference on, IEEE, 2004, pp. 36–43.
[4] J. Yu, M. M. Rashid, A novel dynamic bayesian network-based networked process monitoring approach for fault detection, propagation identification, and root cause diagnosis, AIChE Journal 59 (7) (2013) 2348–2365.
[5] L. H. Chiang, M. E. Kotanchek, A. K. Kordon, Fault diagnosis based on fisher discriminant analysis and support vector machines, Computers & chemical engineering 28 (8) (2004) 1389–1401.
[6] S. Dey, J. Stori, A bayesian network approach to root cause diagnosis of process variations, International Journal of Machine Tools and Manufacture 45 (1) (2005) 75–91.
[7] F. Tsung, Statistical monitoring and diagnosis of automatic controlled processes using dynamic pca, International Journal of Production Research 38 (3) (2000) 625–637.
[8] S. Rogers, M. Girolami, A first course in machine learning, CRC Press, 2016.
[9] R. S. Michalski, J. G. Carbonell, T. M. Mitchell, Machine learning: An artificial intelligence approach, Springer Science & Business Media, 2013.
[10] D. Koller, N. Friedman, L. Getoor, B. Taskar, Graphical models in a nutshell, Introduction to statistical relational learning (2007) 13–55.
[11] D. Koller, N. Friedman, Probabilistic graphical models: principles and techniques, MIT press, 2009.
[12] G. E. Box, G. C. Tiao, Bayesian inference in statistical analysis, Vol. 40, John Wiley & Sons, 2011.
[13] F. V. Jensen, An introduction to Bayesian networks, Vol. 210, UCL press London, 1996.
[14] A. Alaeddini, I. Dogan, Using bayesian networks for root cause analysis in statistical process control, Expert Systems with Applications 38 (9) (2011) 11230–11243.
[15] S. Pradhan, R. Singh, K. Kachru, S. Narasimhamurthy, A bayesian network based approach for root-cause-analysis in manufacturing process, in: Computational Intelligence and Security, 2007 International Conference on, IEEE, 2007, pp. 10–14.
[16] G. Weidl, A. L. Madsen, S. Israelson, Applications of object-oriented bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes, Computers & chemical engineering 29 (9) (2005) 1996–2009.
[17] A. Ankan, pgmpy 0.1.2 documentation (2016).
URL `http://pgmpy.org/`
[18] M. Scutari, bnlearn - bayesian network repository (2017).
URL `http://www.bnlearn.com/bnrepository/`
[19] S. L. Lauritzen, D. J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, Journal of the Royal Statistical Society. Series B (Methodological) (1988) 157–224.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.