Using Bayesian Networks in an Industrial Setting: Making Printing Systems Adaptive

Arjen Hommersom and **Peter J.F. Lucas**[1]

Abstract. Control engineering is a field of major industrial importance as it offers principles for engineering controllable physical devices, such as cell phones, television sets, and printing systems. Control engineering techniques assume that a physical system's dynamic behaviour can be completely described by means of a set of equations. However, as modern systems are often of high complexity, drafting such equations has become more and more difficult. Moreover, to dynamically adapt the system's behaviour to a changing environment, observations obtained from sensors at runtime need to be taken into account. However, such observations give an incomplete picture of the system's behaviour; when combined with the incompletely understood complexity of the device, control engineering solutions increasingly fall short. Probabilistic reasoning would allow one to deal with these sources of incompleteness, yet in the area of control engineering such AI solutions are rare. When using a Bayesian network in this context the required model can be learnt, and tuned, from data, uncertainty can be handled, and the model can be subsequently used for stochastic control of the system's behaviour. In this paper we discuss industrial research in which Bayesian networks were successfully used to control complex printing systems.

1 INTRODUCTION

Many complex physical systems are required to make dynamic trade-offs between the various characteristics of operation, which can be viewed as the capability to *adapt* to a changing environment. For example, in printing systems such characteristics include power division and consumption, the speed of printing, and the quality of the print product. Such trade-offs heavily depend on the system's environment determined by humidity, temperature, and available power. Failure to adapt adequately to the environment may result in faults or suboptimal behaviour, resulting, for example, in low quality print products or low throughput of paper.

The problem of adaptability concerns taking actions based on available runtime information, which we call *making decisions*. As defined above it has two main features. First, making decisions is typically required at a low frequency: it is not necessary and not even desirable to change the speed or energy usage of an engine many times per second. Second, there is a lot of uncertainty involved when making decisions, in particular about the environment, the state of the machine, and also about the dynamics of the system. Complex systems usually cannot be modelled accurately, whereas adaptability requires one to make system-wide, complex, decisions. In order to deal with these uncertainties, techniques where probability distributions can be learnt from available data seem most appropriate.

[1] Radboud University Nijmegen, Institute for Computing and Information Sciences, The Netherlands, email: {arjenh,peterl}@cs.ru.nl

In this paper, we propose to use Bayesian networks [17] to deal with the control of such complex systems. The formalism possesses the unique quality of being both an AI-like and statistical knowledge-representation formalism. Nowadays, Bayesian networks take a central role for dealing with uncertainty in AI and have been successfully applied in many fields, such as medicine and finance. The control of physical systems, on the other hand, is largely done using traditional methods from control theory.

One of the attractive features of Bayesian networks is that they contain a qualitative part, which can be constructed using expert knowledge, normally yielding an understandable, white-box model. Moreover, the quantitative parameters of a Bayesian network can be learnt from data. Other AI learning techniques, such as neural networks, resist providing insight into why a machine changes its behaviour, as they are black-box models. Furthermore, rules—possibly fuzzy—are difficult to obtain and require extensive testing in order to check whether they handle all the relevant situations.

The present paper summarises our successful effort in using Bayesian-network based controllers in the industrial design of adaptive printing systems, which can be looked upon as special stochastic controllers. In our view, as systems get more and more complex, the embedded software will need to be equipped with such AI reasoning capabilities to render the design of adaptive industrial systems feasible.

2 BAYESIAN NETWORKS FOR CONTROL

We first offer some background about Bayesian networks and discuss needed assumptions for modelling and reasoning about dynamic systems using Bayesian networks.

2.1 Background

A *Bayesian network* $B = (G, P)$ consists of a directed acyclic graph $G = (V, E)$, where V is a set of vertices and $E \subseteq V \times V$ is a set of directed edges or arcs, and P is a joint probability distribution associated with a set of random variables X that correspond one-to-one to the vertices of G, i.e., to each vertex $v \in V$ corresponds exactly one random variable X_v and vice versa. As the joint probability distribution P of the Bayesian network is always factored in accordance to the structure of the graph G, it holds that:

$$P(X) = \prod_{v \in V} P(X_v \mid X_{\pi(v)}),$$

where $\pi(v)$ is the set of parents of v. Thus, P can be defined as a family of local conditional probability distributions $P(X_v \mid X_{\pi(v)})$,

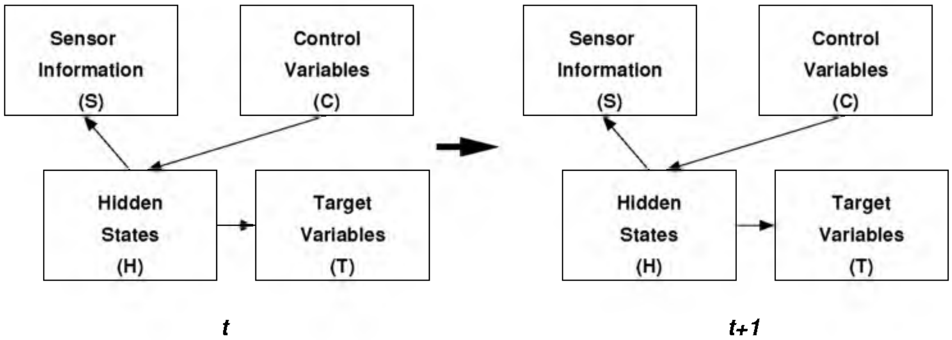

Figure 1. A temporal Bayesian network structure for modelling dynamic systems.

for each vertex $v \in V$. Bayesian networks can encode various probability distributions. Most often the variables are either all discrete or all continuous. Hybrid Bayesian networks, however, containing both discrete and continuous conditional probability distributions are also possible. A commonly used type of hybrid Bayesian network is the conditional linear Gaussian model [3, 8]. Efficient exact and approximate algorithms have been developed to infer probabilities from such networks [10, 2, 12]. Also important in the context of embedded systems is that real-time probabilistic inference can be done using anytime algorithms [6].

A Bayesian network can be constructed with the help of one or more domain experts. However, building Bayesian networks using expert knowledge, although by now known to be feasible for some domains, can be very tedious and time consuming. Learning a Bayesian network from data is also possible, a task which can be separated into two subtasks: (1) structure learning, i.e., identifying the topology of the network, and (2) parameter learning, i.e., determining the associated joint probability distribution, P, for a given network topology. In this paper, we employ parameter learning. This is typically done by computing the maximum likelihood estimates of the parameters, i.e., the conditional probability distributions, associated to the networks structure given data [9].

Temporal Bayesian networks are Bayesian network where the vertices of the graph are indexed with (discrete) time. All vertices with the same time index form a so-called *time slice*. Each time slice consists of a static Bayesian network and the time slices are linked to represent the relationships between states in time. If the structure and parameters of the static Bayesian network are the same at every time slice (with the exception of the first), one speaks of a *dynamic* Bayesian network, as such networks can be unrolled (cf. [16] for an overview).

2.2 Bayesian-network modelling of a dynamic system

Common assumptions in the modelling of dynamic physical systems are that the system is Markovian and stationary (e.g., [11]), i.e., the system state at time $t + 1$ is only dependent on the system state at time t, and the probabilistic dependencies are the same at each time t. Stationarity is an assumption too strong for the work discussed below; however, it is assumed that the network *structure* is the same for every t. In case a particular dependence is absent in a time-slices, then such independence will be reflected in the conditional probability distributions rather than in the structure.

Four different types of vertices were distinguished in developing Bayesian networks for stochastic control:

- **Control variables** C act as input to the physical system's control system, such as the car engine's throttle position.
- **Hidden state variables** H determine the *unobservable* state of the system, such as the engine's speed.
- **Sensor information** S provides observations about the (unobservable) state of the machine (here engine), for example by a measurement of the speed of the car.
- **Target variables** T act as reference values or set-points of the system. It is the purpose of a Bayesian network to control these variables.

A schematic representation of such a network is shown in Figure 1. Given n time slices, a Bayesian network will have an associated joint probability distribution of the following form:

$$P(S_1, C_1, H_1, T_1, \ldots, S_n, C_n, H_n, T_n)$$

The chosen representation closely fits the concepts of traditional control theory. A typical feedback controller influences the system (H) through a system's input (C); it does so by comparing the sensed data (S) with a reference value (T). A feed-forward controller is similar in this view, except that the sensor variables are missing or cannot be observed.

After t time steps, the probability distribution can be updated with the observations $S_1, \ldots, S_t$ and earlier control choices $C_1, \ldots, C_t$ to a probability distribution over the remaining variables:

$$P(H_1, \ldots, H_n, T_1, \ldots, T_n, S_{t+1}, \ldots, S_n, C_{t+1}, \ldots, C_n \\ \mid S_1, \ldots, S_t, C_1, \ldots, C_t)$$

In the following, this conditional probability distribution is abbreviated to

$$P_t(H_1, \ldots, H_n, T_1, \ldots, T_n, S_{t+1}, \ldots, S_n, C_{t+1}, \ldots, C_n) \quad (1)$$

A common question in control is to provide an estimation of the target variable, i.e., to compute $P_t(T_k)$ for some k from the conditional probability distribution (1). If this can be done reliably, it offers the possibility to exploit the network for control. The controller is able to decide what to do in the future by reasoning about the target of control in the future $T_f = T_{t+1}, \ldots T_{t+m}, t + m \leq n$ given a possible choice of control $C_f = C_{t+1}, \ldots T_{t+p}, t + p \leq n$. Both m as well as p can be tuned to domain-specific requirements.

Let $U : T_f \to \mathbb{R}$ be a utility function defined for the target variables T_f. The expected utility for controlling the machine by

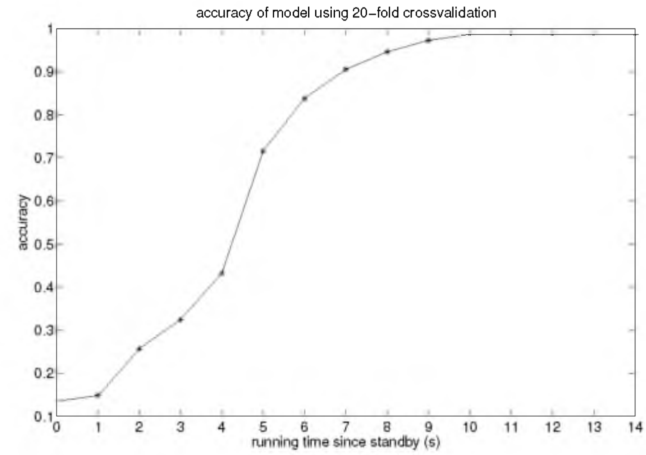

Figure 2. This graph shows the classification accuracy of the Bayesian network plotted as a function of time after the start of a print job.

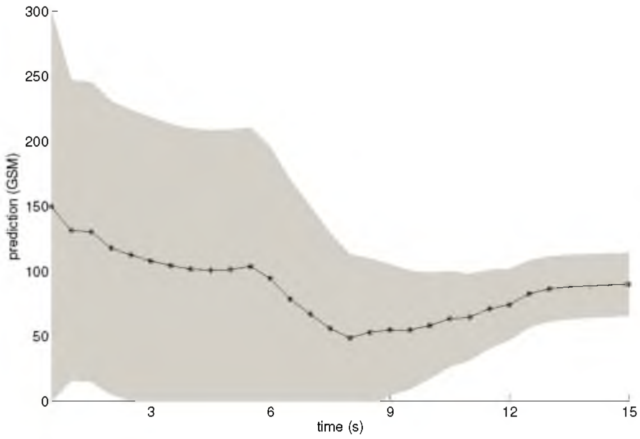

Figure 3. The estimation of paper weight plotted in time. The solid line denotes the mean media weight estimation and the gray area visualises three standard deviations from the mean.

$C_f = c_f$, $eu(c_f)$ is then equal to:

$$eu(c_f) = \sum_{t_f} P_t(t_f|c_f)U(t_f)$$

This approach can also be adapted to continuous variables by integrating over the domain of T_f. A control strategy c_f^* with maximal expected utility yields a maximal value for $eu(c_f)$.

$$c_f^* = \underset{c_f}{\operatorname{argmax}}\, eu(c_f)$$

In Sections 3 and 4, we present research in which we have explored the theory summarised above with the aim of making an industrial printing system adaptive to its environment.

3 ESTIMATION OF MEDIA TYPE

Printing systems contain many challenging control problems. As a first study, we consider the problem of establishing the media weight during a run, which can be exploited during the control of many different parts of the printer. For example, if the media weight, here interpreted as paper type, is known, then we can (i) avoid bad print quality, (ii) avoid engine pollution in several parts of the printer, and (iii) help service technician at service call diagnosis, i.e., to avoid blocking for specific media under specific circumstances. Given the mechanical space limitations in printers, it is non-trivial to design a printer that measures paper properties directly nor is it desirable to ask the user to supply this information. We therefore investigated whether the available sensor data could be used to estimate these properties. This sensor data mainly consists of temperatures and voltages that are required during the regular control of the printer and is available without any extra cost.

The data that is available consists of logging data of runs from stand-by position with a warm machine. In order to vary different conditions, the duration of stand-by prior to the run was deliberately varied. This ensures a realistic variation of temperatures inside the engine. Moreover, the paper type was varied, namely the data contains runs with 70 gsm ($n = 35$), 100 gsm ($n = 10$), 120 gsm ($n = 24$), 140 gsm ($n = 10$), and 200 gsm ($n = 29$) paper.

With the help of the domain experts we designed a Bayesian network structure with 8 vertices at each time-slice. Logging data of an industrial printing system were obtained at a frequency of 2Hz for 15 second, which would yield a total of 240 vertices in the temporal Bayesian network if represented explicitly. All variables were modelled as Gaussian random variables. To convert the Bayesian network into a classifier, the estimations of the model were mapped to a number of classes, corresponding to the distinguished media types.

The plot shown in Figure 2 indicates that it takes some time before the Bayesian network is able to reliably distinguish between the media types based on the sensor information available. After about 6 seconds the classification reaches a performance that is seen as sufficiently reliable for many applications. However, for high-speed printing systems a higher reliability may be required. As the plot shows, on the whole there is a gradual, and robust increase in performance with a form that looks like a sigmoid learning curve. However, note that the only thing that changes in time are the data: the nature of the data is such that in time it becomes easier for the Bayesian network to distinguish between various media types. Further evidence of the robustness of the approach is obtained by computing the confidence intervals of the weight estimates. As shown in Figure 3, the confidence intervals become smaller in time, and conclusions about media type are therefore also more reliable. Hence, it is fair to conclude that the model was able to derive useful information about media type by using sensor information, here about temperature and voltage usage, that is not immediately related to media type.

In case there is reasonable confidence in the estimation, decisions can be made to adapt the system's behaviour. Our work on such adaptation is presented in the next section.

4 CONTROL OF ENGINE SPEED

4.1 Description of the problem

The productivity of printers is limited by the amount of power available, in particular in countries or regions with weak mains. If there is insufficient power available, then temperature setpoints cannot be reached, which causes bad print quality. To overcome this problem, it is either possible to decide to always print at lower speeds or to adapt to the available power dynamically. In this section, we explore the latter option by a dynamic speed adjustment using a Bayesian network.

4.2 Approach

The block diagram in Figure 4 offers an overview of this approach. In this schema, 'sensors' are put on low-level controllers and signal the high-level controller with requests. The network then reasons about appropriate setpoints of the low-level controller. In this problem setting, the high-level controller decides on a specific velocity of the engine based on requested power by a lower-level controller.

For this problem, we look at the model of a part of the printer in more detail. The structure of the model at each time slice is shown in Figure 5. The requested power available is an observable variable that depends on low-level controllers that aim at maintaining the right setpoint for reaching a good print quality. The error variable models the deviation of the actual temperature from the ideal temperature, which can be established in a laboratory situation, but not during runtime. If this exceeds a certain threshold, then the print quality will be below a norm that has been determined by the printer manufacturer.

Both velocity and available power influence the power that is or can be requested by the low-level controllers. Furthermore, the combination of the available power and the requested power is a good predictor of the error according to the domain experts. To model the dynamics, we use two time slices with the interconnections between the available power – which models that the power supply on different time slices is not independent – and requested power, which models the state of the machine that influences the requested power.

We again considered to model all the variables as Gaussian distributed random variables. This seemed reasonable, as most variables were Gaussian distributed, however with the exception of the available power (see Figure 6). Fitting a Gaussian distribution to such a distribution will typically lead to insufficient accuracy. To improve this, this variable was modelled as a mixture of two Gaussian distributed variables, one with mean $\mu^{\text{low}}_{\text{Power}}$ and one with mean $\mu^{\text{high}}_{\text{Power}}$ with a small variance. Such a distribution can be modelled using a hybrid network as follows. The network is augmented with an additional (binary) parent vertex S with values 'high' and 'low' for the requested power variable. For both states of this node, a normal distribution is associated to this variable. The marginal distribution of requested power is obtained by basic probability theory by

$$P(\text{P}_{\text{req}}) = \sum_S P(\text{P}_{\text{req}} \mid S)P(S).$$

4.3 Error estimation

The main reasoning tasks of the network is to estimate the error, i.e., the deviation from the ideal temperature, given a certain velocity and observations. This problem could be considered as a classification task, i.e., the print quality is *bad* or *good*. The advantage is that this

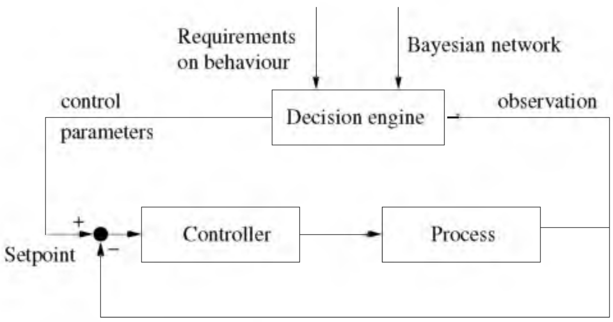

Figure 4. Architecture of an adaptive controller using a Bayesian network.

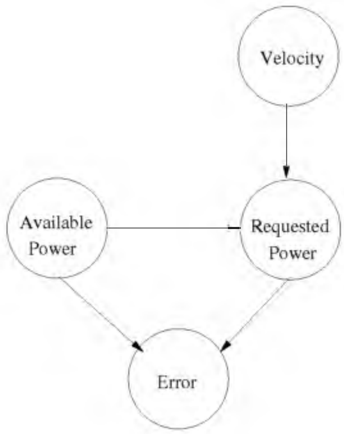

Figure 5. Structure of the Bayesian network of each time slice.

provides means to compare different models and see how well it performs at distinguishing between these two possibilities. A standard method to visualise and quantify this is by means of a Receiver Operating Characteristic (ROC) curve, which shows the relation between the false positive ratio and the true positive ratio (sensitivity). The area under the curve is a measure for its classification performance.

We have compared three models, i.e., a discrete model, a fully continuous model and a hybrid model for modelling the distribution of the requested power with two normally distributed random variables. The classification performance is outlined in Figure 7. As expected, the fully continuous model performs worse, whereas the hybrid and discrete show a similar trend. The advantage of the discrete version is that the probability distribution can easily be inspected and it has no underlying assumptions about the distribution, which makes it easier to use in practice. The hybrid version however allows for more efficient computation as we need a large number of discrete values to describe the conditional distributions. For this reason, we have used the hybrid version in the following.

4.4 Decision making for control

As the error information is not observable during runtime, the marginal probability distribution of the error in the next time slice is computed using the information about the power available and power requested. This error is a normal random variable with mean μ and standard deviation σ. The maximum error that we allow in this domain is denoted by $E_{\max}$ and we define a random variable for print quality Q_k, which is true if $\mu + k\sigma < E_{\max}$, where k is a con-

Figure 6. Distribution of requested power.

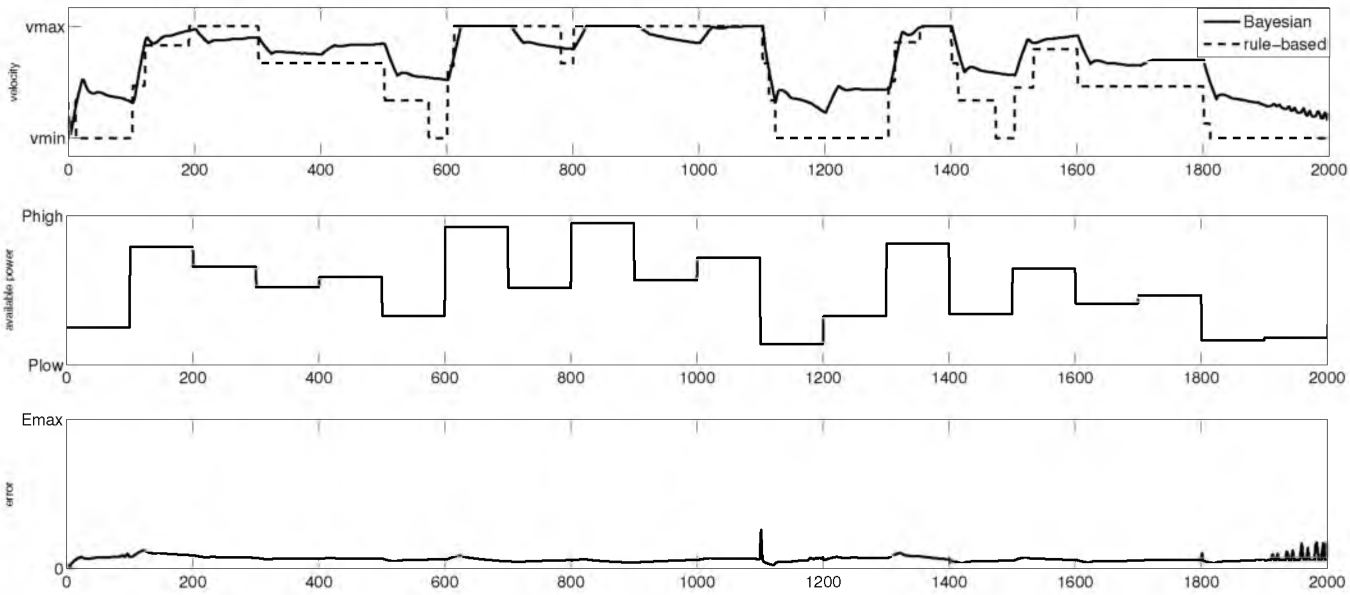

Figure 8. In the centre figure, the available power is plotted, which is fluctuating. at the top, we compare the velocity of the engine which is controlled by a rule-based system and by a Bayesian network. Below, we present the error that the controller based on the Bayesian network yields, which is within the required limits.

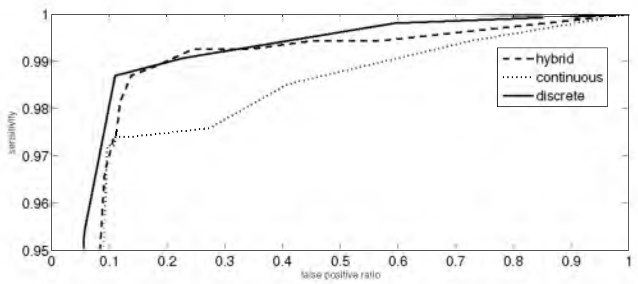

Figure 7. ROC curves of the three Bayesian networks. The hybrid and discrete versions show the best classification performance.

stant. Different values of k correspond to different points on the ROC curve as depicted in Figure 7. For a normal random variable, more than 99.73% of the real value of the error will be within three standard deviations of the mean, so for example $k = 3$ would imply that $P(\text{Error}_{t+1} < E_{\max}) > 99.87\%$. The target variables of our control are the print quality, modelled by Q_k, and the velocity V. Define a utility function $U : \{Q_k, V\} \to \mathbb{R}$ as:

$$U(q, v) = \begin{cases} -1 & \text{if } q = \bot \\ v & \text{otherwise} \end{cases}$$

and apply the maximal expected utility criterion as discussed in Section 2. This implies that the expected utility of a velocity v, $eu(v)$ equals -1 or v depending on the risk of having bad print quality. In effect, we choose the highest velocity v such that $Q_k = \top$.

In order to evaluate the approach, we compared the productivity of the resulting network with the *rule-based method* (implemented in terms of repetitive and conditional statements of a programming language for use in the actual control engine) that incorporates some heuristics for choosing the right velocity. The productivity is defined here simply as $\int_0^\tau v(t)dt$, where τ is the simulation time. In order to smooth the signal that the network produces, we employ a a FIR (Finite Impulse Response) filter in which we employ a moving average of 10 decisions. The resulting behaviour was simulated and is presented in Figure 8 (with $k = 3$). Compared to the rule-based approach, we improve roughly 9% in productivity while keeping the error within an acceptable range. While it could certainly be the case that the rules could be improved and optimised, the point is that the logic underlying the controller does not have to be designed. What is required is a qualitative model, data, and a probabilistic criterion that can be inferred.

5 RELATED WORK

Bayesian inference is well-known for the inference of hidden states in a dynamic model. Typical applications are filtering – i.e., inferring the current hidden state given the observations in the past – and smoothing where past states are inferred. For example, the Kalman filter [7] is well-known in stochastic control theory (see e.g., [1]) and is a special case of a dynamic Bayesian networks, where the model is the linear Gaussian variant of a hidden Markov model, i.e., it describes a Markov process with noise parameters on the input and output variables. Non-linear variants, such as the extended Kalman filter or the unscented Kalman filter (see e.g., [15]) are approximate inference algorithms for non-linear Gaussian models by linearisation of the model. More recently, particle filters [13], have been proposed as an alternative, which relies on sampling to approximate the posterior distribution.

The difference with these filtering approaches is that for Bayesian networks there is an underlying domain model which is understandable. As Bayesian networks are general formalisms, they could also be used or re-used for diagnostic purposes, where it is typically required that a diagnosis can be represented in a human-understandable way so that proper action can be taken (e.g., [19] in the printing domain). Furthermore, it is well-known that the structure of the graphical part of a Bayesian network facilitates the assessment of probabilities, even to the extent that reliable probabilistic information can be obtained from experts (see [14]). One other advantage compared to black-box models is that the modelled probability distribution can be exploited for decision making using decision theory. This is particularly important if one wants to make real trade-offs such as between productivity and energy consumption.

With respect to decision making, adaptive controllers using explicit Bayesian networks have not been extensively investigated. The most closely related work is by Deventer [4], who investigated the use of dynamic Bayesian networks for controlling linear and non-linear systems. The premise of this work is that the parameters of a Bayesian network can be estimated from a deterministic physical model of the system. In contrast, we aim at using models that were learnt from data. Such data can be obtained from measurements during design time or during runtime of the system.

Several approaches for traditional adaptive control already exists. First, model-reference adaptive control uses a reference model that reflects the desired behaviour of the system. On the basis of the observed output and of the reference model, the system is tuned. The second type of adaptive controllers are so called self-tuning controllers, which estimate the correct parameters of the system based on observations and tunes the control accordingly. Our approach employs a mixture of the two, where a reference model is given by a Bayesian network and tunes other parts of the system accordingly. In the last few decades, also techniques from the area of artificial intelligence, such as rule-based systems, fuzzy logic, neural networks, evolutionary algorithms, etc. have been used in order to determine optimal values for control parameters (see e.g. [5]). The work presented in this paper extends these approaches using human-readable Bayesian networks.

6 CONCLUSIONS

In embedded software, there is an increasing trend to apply and verify new software methods in an industrial context, i.e., the *industry-as-laboratory* paradigm [18]. This means that concrete cases are studied in their industrial context to promote the applicability and scalability of solution strategies under the relevant practical constraints. Much of the current AI research, on the other hand, is done in theory using standard benchmark problems and data sets. It poses a number of challenges if one wishes to apply an AI technique such as Bayesian networks to industrial practice. First, there is little support for modelling systems in an industrial context. Bayesian networks are expressive formalisms and little guidance is given to the construction of networks that can be employed in such an industrial setting. Moreover, there seems to be little theory of using Bayesian networks in these areas. For example, while there is a lot of research in the area of stochastic control, it is unclear how these results carry over to Bayesian networks. Similarly, techniques developed in context of Bayesian networks do not carry over to the problem of control.

Bayesian networks have drawn attention in many different research areas, such as AI, mathematics and statistics. In this paper, we have explored the use of Bayesian networks for designing an adaptable printing systems. We have shown that the approach is feasible and can act as a basis for designing an intelligent printing system. This suggests that Bayesian networks can have a much wider application in the engineering sciences, in particular for control and fault detection. With the increasing complexity of systems, there is little doubt that these AI techniques will play a pivotal role in industry.

ACKNOWLEDGEMENTS

This work has been carried out as part of the OCTOPUS project under the responsibility of the Embedded Systems Institute. This project is partially supported by the Netherlands Ministry of Economic Affairs under the Embedded Systems Institute program. We thank Marcel van Gerven for making his Bayesian network toolbox available to us.

REFERENCES

[1] K.J. Åström, *Introduction to Stochastic Control Theory*, Academic Press, 1970.

[2] G. Casella and C. Robert, *Monte Carlo Statistical Methods*, Springer-Verlag, 1999.

[3] R.G. Cowell, A.P. Dawid, S.L. Lauritzen, and D.J. Spiegelhalter, *Probabilistic Networks and Expert Systems*, Springer, 1999.

[4] R. Deventer, *Modeling and Control of Static and Dynamic Systems with Bayesian Networks*, Ph.D. dissertation, University Erlangen-Nürnberg, Chair for Pattern recognition, 2004.

[5] J.A. Farrell and M.M. Polycarpou, *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*, Adaptive and Learning Systems for Signal Processing, Communications and Control Series, Wiley-Interscience, 2006.

[6] H. Guo and W.H. Hsu, 'A survey of algorithms for real-time Bayesian network inference', in *AAAI/KDD/UAI02 Joint Workshop on Real-Time Decision Support and Diagnosis Systems*, eds., A. Darwiche and N. Friedman, Edmonton, Canada, (2002).

[7] R.E. Kalman, 'A new approach to linear filtering and prediction problems', *Journal of Basic Engineering*, **82**(1), 35–45, (1960).

[8] S.L. Lauritzen, 'Propagation of probabilities, means and variances in mixed graphical association models', *Journal of the American Statistical Association*, **87**, 1098–1108, (1992).

[9] S.L. Lauritzen, 'The EM algorithm for graphical association models with missing data', *Computational Statistics and Analysis*, **19**, 191–201, (1995).

[10] S.L. Lauritzen and D.J. Spiegelhalter, 'Local computations with probabilities on graphical structures and their application to expert systems', *Journal of the Royal Statistical Society*, **50**, 157–224, (1988).

[11] U. Lerner, B. Moses, M. Scott, S. Mcilraith, and S. Koller, 'Monitoring a complex physical system using a hybrid dynamic Bayes net', in *Proceedings of the UAI*, (2002).

[12] U. Lerner and R. Parr, 'Inference in hybrid networks: theoretical limits and practical algorithms', in *Proceedings of the UAI*, eds., J. Breese and D. Koller, volume 17, pp. 310–318, San Francisco, CA, (2001). Morgan Kaufmann.

[13] J.S. Liu and R. Chen, 'Sequential Monte Carlo methods for dynamic systems', *Journal of the American Statistical Association*, **93**, 1032–1044, (1998).

[14] P.J.F. Lucas, H. Boot, and B.G. Taal, 'Computer-based decision-support in the management of primary gastric non-Hodgkin lymphoma', *Meth Inform Med*, **37**, 206–219, (1998).

[15] P.S. Maybeck, *Stochastic models, estimation, and control*, Academic Press, 1979.

[16] K.P. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. dissertation, UC Berkeley, 2002.

[17] J. Pearl, *Probabilistic Reasoning in Inteligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[18] C. Potts, 'Software-engineering research revisited', *IEEE Software*, **19**(9), 19–28, (1993).

[19] Claus Skaanning, Finn V. Jensen, and Uffe Kjærulff, 'Printer troubleshooting using Bayesian networks', in *Proceedings of IEA/AIE*, pp. 367–379. Springer-Verlag New York, Inc., (2000).