Bayesian Network Conflict Detection for Normative Monitoring of Black-Box Systems

Annet Onnes, Mehdi Dastani, Silja Renooij
Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
{a.t.onnes, m.m.dastani, s.renooij}@uu.nl

Abstract

Bayesian networks are interpretable probabilistic models that can be constructed from both data and domain knowledge. They are applied in various domains and for different tasks, including that of anomaly detection, for which an easy to compute measure of data conflict exists. In this paper we consider the use of Bayesian networks to monitor input-output pairs of a black-box AI system, to establish whether the output is acceptable in the current context in which the AI system operates. A Bayesian network-based prescriptive, or normative, model is assumed that includes context variables relevant for deciding what is or is not acceptable. We analyse and adjust the conflict measure to make it applicable to our new type of monitoring setting.

Introduction

Ever since humans have engaged with technology, we have monitored operations to ensure that the technology is safe and reliable. The demand for inspecting and controlling omnipresent black-box AI systems as they take over decision-making and operation in increasingly more critical situations is therefore not surprising. Even when such an AI system is developed with matters as safety and reliability in mind, it can still be a black-box when deployed. As a result, it is difficult to guarantee that the system's behaviour is as it ought to be, given the specific context in which it is operating. When the AI system is developed any general constraints can be taken into account through system requirements; however, context-specific constraints only become clear when the system is in use in that context. Take for example a medical decision-support system, designed to be used in multiple hospitals. Even if the system is considered generally accurate, when used in a specific hospital for a specific patient, the additional context provided by e.g. local hospital protocols or patient-specific information, may call for a different decision than suggested by the system. To detect this, we in essence need a glass-box that can constrain the behaviour of the black-box in a transparent way (Aler Tubella et al. 2019).

We will offer a first step towards a technical implementation of a glass-box for the purpose of monitoring black-box AI systems. We propose to use Bayesian networks (BNs) as a prescriptive, or normative, model of the context-specific acceptable behaviour. BNs are probabilistic models that can both handle uncertainty and are known for their interpretability and transparency. Additionally, to detect deviations from acceptable behaviour, we take inspiration from the field of anomaly detection. The field of anomaly detection (AD) studies how to detect when the behaviour of a system, or a real-life process, deviates from what is considered normal, typically through modelling the normal behaviour. This setting differs from our current setting in two important ways. First, a model of normal behaviour as used in AD is a descriptive model rather than a prescriptive one. Secondly, our setting adds an additional layer of uncertainty and complexity by including the AI system that in itself is a model of real-world processes. As a result, existing techniques from AD cannot be directly employed for the purpose of monitoring AI systems.

This paper contributes the following. We introduce the novel setting of monitoring under uncertainty of black-box AI systems using normative models of context-specific behaviour; demonstrate that existing AD techniques need adjustment to be used in this setting; and illustrate the aforementioned for an existing Bayesian network conflict measure. After reviewing existing work on AD and BNs, we introduce and formalise our new normative monitoring setting. We then analyse the conflict measure for AD using BNs and adjust it to fit the new setting.

Preliminaries

In this section we briefly review AD methods, the BNs in AD, and introduce our notations.

Anomaly Detection

The aim of AD is to identify data patterns, known as anomalies, that deviate from normal behaviour (Chandola, Banerjee, and Kumar 2009). Anomaly detection can be used for fraud, intrusion or fault detection. AD approaches generally consist of two steps. The first is to construct or train a model of normal behaviour and the second is to use this model to detect anomalies at run-time. Figure 1a presents a schematic overview of the general AD setting. The real world process or system that is being monitored for anomalies is the *target system*, from which we can typically observe only partial, indirect, and hence uncertain information. The target system

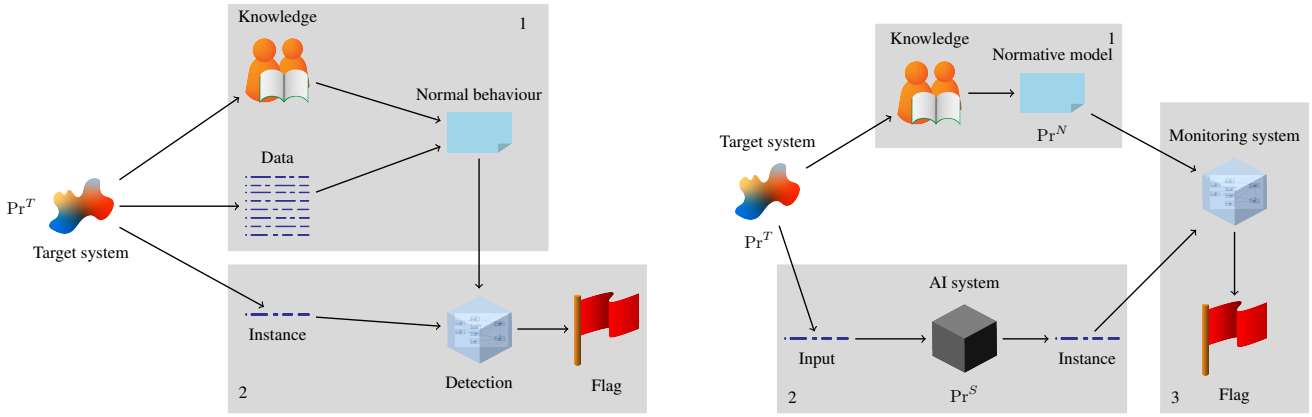

Figure 1: Overview of (a) Anomaly Detection with step 1 modelling normal behaviour and step 2 detecting anomalies and of (b) normative monitoring setting with step 1 constructing the normative model, step 2 running the AI system and step 3 monitoring an input-output pair.

is therefore taken to generate data from some partially observable distribution $\Pr^T$.

Human experts can observe the target system and establish (uncertain) knowledge about how the real world process normally behaves. For the purpose of AD, knowledge and data are used to construct a *descriptive* model of normal behaviour. An AD system is now tasked with detecting whether a newly observed data pattern from the target system is an anomaly and should be flagged. To this end it is compared against the model of normal behaviour using a suitable measure.

Bayesian Networks in Anomaly Detection

Among the available methods used for representing normal behaviour in the context of AD are Bayesian networks (BNs) (Nielsen and Jensen 2009). A Bayesian network $\mathfrak{B} = (G, \Pr)$ is a representation of a joint probability distribution $\Pr$ over a set of discrete random variables $\mathbf{V}$ that exploits the independencies among the variables as portrayed in the acyclic directed graph G. We use capital letters to denote variables, bold-faced in case of sets. Each variable $V \in \mathbf{V}$ can be assigned a value $v \in \Omega(V)$; a joint value assignment (or configuration) $v_1 \wedge \ldots \wedge v_n$ to a set of variables $\mathbf{V} = \{V_1, \ldots, V_n\}$ is denoted by $\mathbf{v}$. Such a joint assignment can for example describe an *instance*, or data pattern in AD.

The joint distribution $\Pr(\mathbf{V})$ factorises over local distributions specified for each variable, conditional on its parents in the graph. This allows for efficient computation of any prior or posterior probabilities of interest.

Figure 2 shows the graph of a small BN, with $\mathbf{V} = \{I_1, I_2, A, O\}$. In a strongly simplified manner, it represents the diagnosis (O) for a patient, two possible symptoms (I_1 and I_2) and some additional contextual information (A).

BNs are used in AD to model normal behaviour and the associated uncertainty. Further, a method is required to establish whether an instance deviates from normal behaviour. Nielsen and Jensen (2007) demonstrate the use of a conflict measure introduced by Jensen et al. (1990), to detect abnormal behaviour in production plants using instances consisting of sensor readings. In case of normal behaviour, captured by a BN, the sensor readings should be positively correlated. An instance is flagged as anomalous when this is not the case.

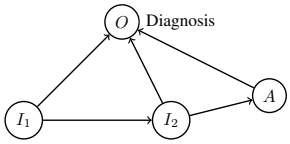

Figure 2: Graph of a small diagnostic BN.

Normative Monitoring

Compared to the AD setting sketched in Figure 1a, where the target system is directly monitored, in the normative setting (Figure 1b) an AI system is monitored in order to decide whether the input-output pairs from the AI system are (un)acceptable according to some context-specific constraints. These constraints capture the norms specified by human experts. We need *normative models* to represent these norms prescribed to the AI system in a particular context.

Normative Models

In regular AD the model used for detection approximates normal behaviour of (part of) the target system and therefore is a *descriptive* model, as it describes normal behaviour. Descriptive models are often created using data, which is generated by the target system specifically under normal circumstances. For normative monitoring we require a *prescriptive* model, as it prescribes what behaviour is expected of the AI system. We emphasize that the prescriptive model is not aimed at monitoring the performance of the AI system, but rather its adherence to norms. Moreover, when this prescrip-

tive normative model is transparent, it can operate as a glass-box for monitoring a black-box AI system.

Norms that can be captured in the normative model are rules and principles that can enact a value (Aler Tubella et al. 2019) and that must be accepted in the context that the monitoring system is designed for, and are therefore accepted in a specific context, i.e. by a particular community (stakeholders) at a particular time (Brennan et al. 2013). Note that we do not hold norms to be statistical patterns that *describe* what the norm is in a context, we hold norms to be *prescriptive*, as they prescribe expected and accepted behaviour. In some cases the two notions coincide: when everyone starts to follow a rule, it also becomes a statistical norm. In our medical decision-making example, the normative model can capture norms that are specified by medical experts and laid down in treatment protocols of a specific hospital, and consider additional information about the patient relevant to such protocols.

Using norms in monitoring is not new, nor is modelling uncertainty for anomaly detection. Various monitoring approaches overlook uncertainty by using rule-based systems to model norms (Dastani, Torroni, and Yorke-Smith 2018). In this paper we focus on uncertainty in the normative model and therefore opt for a BN-based normative model. As discussed, in the standard AD setting, BNs have been used as models of normal behaviour, often learned from data. As such they capture stochastic uncertainty of the real world in addition to uncertainty introduced by the modelling itself. Rather than modelling descriptive norms based on data, in the normative setting the BN is used to capture prescriptive norms, based on human (expert) knowledge. BNs are generally known for being interpretable and can be handcrafted using knowledge elicited from stakeholders (Kjaerulff and Madsen 2013). BNs have for example been used in the medical domain to model protocols as prescriptive norms elicited from expert knowledge (Zheng, Kang, and Kim 2006). Further discussion on how to construct normative BNs, or normative models in general, is beyond the scope of this paper.

Model Formalisation

Our normative monitoring setting builds on an AI system and a normative model, where we assume that both represent a probability distribution. Here we will formally define these models and their relation.

Definition 1 *We define the following models:*

- *A* normative model *represents a joint distribution* $\Pr^N(\mathbf{V}^N)$ *over a set of variables* $\mathbf{V}^N$.
- *An* AI system *represents a joint distribution* $\Pr^S(\mathbf{V}^S)$ *over a set of variables* $\mathbf{V}^S = \mathbf{I}^S \cup \{O^S\}$*, where* $\mathbf{I}^S$ *is a non-empty set of* n *input variables and* $\{O^S\}$ *represents a single output variable.*

We assume that the two models partly share the same variables (with the same values) or that there is a straightforward mapping between them. More specifically, in this paper we assume that $\mathbf{V}^N = \mathbf{I}^S \cup \{O^S\} \cup \mathbf{A}$, which means that the normative model includes the AI system's input and output variables, as well as a non-empty set of additional variables $\mathbf{A}$. The variables in $\mathbf{A}$ are used for representing context-specific norms; through a value-assignment $\mathbf{a}'$ to $\mathbf{A}' \subseteq \mathbf{A}$ the normative model can be adapted to a specific context $\mathbf{a}'$.

In this paper we assume that the normative model is a BN; as a result we have complete information about the distribution $\Pr^N(\mathbf{V}^N)$ it represents. For the AI system we have available input-output pairs $(\mathbf{i}, o)$, but we lack exact knowledge of $\Pr^S$.

Example To provide insight into how the abstract idea of normative monitoring can be used in practice, we reconsider the example in medical decision-making. The AI system designed to assist is a black-box system trained using patient data from e.g. many different, inconsistent sources; it is able to fulfill its general task at a high level of accuracy. When the system is used to support treatment decisions for an individual patient in a specific hospital, this is the specific context in which the AI system operates and in which we want to monitor it. The monitoring system compares a patient-specific input-output pair from the AI system to a normative model that captures the context-specific information. We adopt a strongly simplified interpretation of this example to demonstrate our findings. Reconsider the small diagnostic BN whose graph is shown in Figure 2. In the normative monitoring setting the BN captures the norms, it represents the distribution $\Pr^N(\mathbf{V})$, with $\mathbf{V} = \{I_1, I_2, A, O\}$ of which we have complete knowledge. We have input variables $\mathbf{I} = \{I_1, I_2\}$, the additional variable $\mathbf{A} = \{A\}$ and the output variable O. It is used to monitor the AI system, representing the distribution $\Pr^S(I_1, I_2, O)$, the details of which are unknown to us. In order for the monitoring system to determine which $(i_1 \wedge i_2, o)$ to flag, we need to be able to detect whether or not the pair is acceptable in context $\mathbf{a} = a$.

Detecting Unacceptable Input-Output Pairs

The normative model can be used in different ways to detect unacceptable input-output pairs, just like models of normal behaviour are used in different ways to detect anomalies in standard AD. When using a BN as a model of normal behaviour, detecting anomalies can be done by using probability-based measures (Johansson and Falkman 2007; Mascaro, Nicholson, and Korb 2014; Nielsen and Jensen 2007; Kirk, Legg, and El-Mahassni 2014). In this section we will analyse the suitability of using Jensen's conflict measure in the setting of monitoring AI systems with a BN-based normative model.

Conflict as a Measure for Detection

Jensen's measure to detect conflict within an instance $\mathbf{e} = e_1 \wedge \ldots \wedge e_t$ that combines $t \geq 2$ pieces of evidence, is defined as

$$\text{confl}(e_1, \ldots, e_t) = \log \frac{\Pr(e_1) \cdot \ldots \cdot \Pr(e_t)}{\Pr(\mathbf{e})} \qquad (1)$$

Note that in case all pieces of evidence are mutually independent, the numerator is equal to the denominator and the measure becomes $\log(1) = 0$. When the joint probability

is larger than the product of the marginal probabilities, it means that the observations in the instance are more likely to occur together than separately, while the reverse indicates conflict. Therefore, a positive value for the conflict measure indicates conflict and a negative value indicates no conflict.

In our setting, we are interested in calculating the conflict using the normative model, so Pr is $\Pr^N$. From the perspective of the normative model, the input-output pairs from the AI system form the observable instances $\mathbf{e} = o \wedge \mathbf{i}$ over which we calculate the conflict measure. That is, we compute $\text{confl}(o, i_1, \ldots, i_n)$, where $i_1 \wedge \ldots \wedge i_n = \mathbf{i}$. The normative model includes additional variables $\mathbf{A}$ that may also have observations. Thus, we want to determine whether or not there is an input-output conflict in a context $\mathbf{a}'$ for $\mathbf{A}' \subseteq \mathbf{A}$. This means that $\Pr^N(\cdot)$ is in fact a conditional distribution $\Pr^N(\cdot \mid \mathbf{a}')$, which we denote by $\Pr^N_{\mathbf{a}'}(\cdot)$.

Adjusting the conflict measure The conflict measure as defined above is not directly suitable for normative monitoring of AI systems. By indirectly modelling the target system through the AI system (see Figure 1b), there is additional uncertainty in the overall setting, both in how the AI system models the target system, as well as in the predictions of the AI system itself. With the increase in complexity in the normative setting, we have to carefully consider what is exactly being measured by the conflict measure.

Our aim is to monitor the AI system's behaviour, regardless of the target system's stochasticity that feeds into the monitoring system via the input $\mathbf{i}$. In monitoring the AI system we only want to consider the dependency between the input and output of that system, rather than considering all conflict, including that between the inputs $i_1, \ldots, i_n$. We do not want to monitor the process that generated the input data, as would be the case in regular anomaly detection. The conflict within the input is noise in determining whether there is conflict in what the AI system is outputting according to the normative model. Intuitively we can therefore remove the conflict that is in the input from the conflict of input and output together. We therefore define the IOconfl measure for an input-output instance $o \wedge \mathbf{i}$ with $\mathbf{i} = i_1 \wedge \ldots \wedge i_n$ as:

$$\begin{aligned} \text{IOconfl}(o, \mathbf{i}) &= \text{confl}(o, i_1, \ldots, i_n) - \text{confl}(i_1, \ldots, i_n) \\ &= \log \frac{\Pr(o) \cdot \Pr(\mathbf{i})}{\Pr(o \wedge \mathbf{i})} \end{aligned} \tag{2}$$

From the above we have that IOconfl() is in essence a special case of confl() with exactly 2 arguments. As such, it inherits the properties of the original measure: it is easy to calculate, independent of the order of the arguments, and has a natural interpretation in terms of capturing a degree of (in)coherence among its arguments (Jensen et al. 1990).

Flagging The threshold for the original measure is an intrinsic threshold of 0, capturing the state of the model in which the t individual pieces of evidence under consideration are independent. We consider what it means to use this same threshold for IOconfl(). An IOconfl()-value of 0 indicates that $\mathbf{i}$ and o are independent according to the normative model and the given context ($\Pr = \Pr^N_{\mathbf{a}'}$). If it exceeds this default threshold, then $(\Pr(o) \cdot \Pr(\mathbf{i}))/(\Pr(o \wedge \mathbf{i})) = \Pr(o)/\Pr(o \mid \mathbf{i}) > 0$ indicates nothing more than that the probability of output o has decreased as a result of input $\mathbf{i}$ in the given context. This might be an intuitive interpretation for a conflict measure, but whether or not this is sufficient reason to flag may depend on the domain of application.

Conclusion

In this paper we introduced the novel setting of normative monitoring of black-box AI systems using prescriptive models of context-specific behaviour. By building on transparent normative models, the setting provides for a first step towards an implementation of a glass-box concept.

Bayesian networks are interpretable probabilistic models that can be constructed from both data and expert knowledge. As such they are applied in various domains and for different tasks, including that of standard anomaly detection. For the latter purpose, an easy to compute measure of data conflict exists. Inspired by the use of BNs in combination with conflict measures in the standard anomaly detection context, we studied how to transfer these techniques to our novel setting. More specifically, we proposed the use of BNs for representing prescriptive normative models and adjusted a conflict measure to allow for measuring the conflict, according to the normative model, within an input-output pair produced by the AI system.

Further analysis into the behaviour of the measure under various circumstances is needed to determine whether the threshold of the original measure satisfies.

To further demonstrate the strengths of the proposed measure and suitability of the threshold, a proper evaluation in practice is necessary. This, however, requires the availability of a researched and evaluated normative model, which is far beyond the scope of this paper to accurately achieve. For illustration purposes, we used the problem of monitoring a medical decision-support system that should adhere to local hospital protocols, captured in the normative model. Important properties such as safety and reliability of an AI system can in some regard be considered as emergent (Leveson 2012). As a result, only when monitoring an AI system in the context where it is deployed can we monitor for these properties. This leads us to conclude that by using transparent normative models, such as those based on BNs, we can effectively create a glass-box by utilising existing research on knowledge-driven techniques and uncertainty to enhance data-driven techniques, leading us to overall more reliable, safe, responsible and usable AI systems.

Acknowledgements

This research was funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl.

References

Aler Tubella, A.; Theodorou, A.; Dignum, F.; and Dignum, V. 2019. Governance by glass-box: Implementing transparent moral bounds for AI behaviour. In *Proceedings of*

the Twenty-Eighth International Joint Conference on Artificial Intelligence, 5787–5793. International Joint Conferences on Artificial Intelligence.

Brennan, G.; Eriksson, L.; Goodin, R. E.; and Southwood, N. 2013. *Explaining Norms*. Oxford University Press.

Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. *ACM Computing Surveys* 41(3):1–58.

Dastani, M.; Torroni, P.; and Yorke-Smith, N. 2018. Monitoring norms: A multi-disciplinary perspective. *The Knowledge Engineering Review* 33:e25.

Jensen, F. V.; Chamberlain, B.; Nordahl, T.; and Jensen, F. 1990. Analysis in HUGIN of data conflict. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 546–554.

Johansson, F., and Falkman, G. 2007. Detection of vessel anomalies – A Bayesian network approach. In *Proceedings of the Third International Conference on Intelligent Sensors, Sensor Networks and Information*, 395–400. IEEE.

Kirk, A.; Legg, J.; and El-Mahassni, E. 2014. Anomaly detection and attribution using Bayesian networks. Technical report, Defence Science and Technology Organisation Canberra.

Kjaerulff, U. B., and Madsen, A. L. 2013. *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*. Springer, 2nd edition.

Leveson, N. G. 2012. *Engineering a Safer World: Systems Thinking Applied to Safety*. The MIT Press.

Mascaro, S.; Nicholson, A. E.; and Korb, K. B. 2014. Anomaly detection in vessel tracks using Bayesian networks. *International Journal of Approximate Reasoning* 55(1):84–98.

Nielsen, T. D., and Jensen, F. V. 2007. On-line alert systems for production plants: A conflict based approach. *International Journal of Approximate Reasoning* 45:255–270.

Nielsen, T. D., and Jensen, F. V. 2009. *Bayesian Networks and Decision Graphs*. Springer Science & Business Media.

Zheng, H.-T.; Kang, B.-Y.; and Kim, H.-G. 2006. An ontology-based Bayesian network approach for representing uncertainty in clinical practice guidelines. In *Uncertainty Reasoning for the Semantic Web I*. Springer. 161–173.