Same-Decision Probability: Threshold Robustness and Application to Explanation

Silja Renooij S.RENOOIJ@UU.NL
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Abstract

The same-decision probability (SDP) is a confidence measure for threshold-based decisions. In this paper we detail various properties of the SDP that allow for studying its robustness to changes in the threshold value upon which a decision is based. In addition to expressing confidence in a decision, the SDP has proven to be a useful tool in other contexts, such as that of information gathering. We demonstrate that the properties of the SDP as established in this paper allow for its application in the context of explaining Bayesian network classifiers as well.

Keywords: Bayesian network classifiers; threshold-based decisions; Same-decision probability; robustness; explanations.

1. Introduction

The output of any classifier that bases its decision on some function value exceeding a given threshold is sensitive to changes in that threshold. As a result, not only individual decisions, but also quality measures that summarize the performance of such classifiers, such as accuracy, sensitivity, and specificity, depend on the choice of threshold. In fact, ROC-curves are useful tools for displaying the trade-off between sensitivity and specificity of a classifier upon varying the threshold. In this paper we focus on the concept of *same-decision probability* (SDP), a recently introduced confidence measure for threshold-based decisions made by Bayesian network classifiers (Choi et al., 2012). In this context, it is assumed that evidence $\mathbf{e}$ is available for a (possibly empty) subset of attributes and that $\Pr(c \mid \mathbf{e}) \geq T$, that is, the classifier currently outputs decision c based upon its posterior probability surpassing a given threshold T. The SDP now equals the probability of making the same decision upon obtaining additional evidence. The SDP thus quantifies the robustness of threshold-based decisions to changes in evidence. The robustness of such decisions to changes in a network's probability parameters can also be analysed, using sensitivity analyses (van der Gaag and Coupé, 1999). In this paper we focus on the effects of changing the threshold itself.

The SDP has already proven its use in contexts other than measuring decision confidence, such as in selective evidence gathering (Chen et al., 2014), for value of information computations (Chen et al., 2015), and for optimal feature selection (Choi et al., 2017). In addition, so-called prime implicant (PI) explanations were noted to have an SDP of 1 (Shih et al., 2018).

In this paper we further study the SDP and analyse its properties, assuming a given order on evidence instantiations; how to determine this order is also addressed. Our findings allow for establishing threshold robustness for both approximate and exact SDP computations. In addition, we use our findings to exactly specify the relation between the SDP and PI-explanations, opening up the possibility of alternative algorithms for establishing such explanations. This paper therefore provides robustness results for threshold-based measures in general and for the SDP in particular, and moreover illustrates another application context for the SDP.

Algorithm 1: Computing $SDP_{c,\lambda}(\mathbf{X} \mid \mathbf{e})$ in a naive Bayesian network

(Pseudocode, adapted from Chen et al. (2014), is for illustration purposes only)

```
input  : naive Bayesian network with class variable $C$, attributes $\mathbf{X} = \{X_1, \ldots, X_m\}$, evidence $\mathbf{e}$;
         $\lambda$: log-odds threshold
output: Same-Decision Probability $p_{SD}$

    procedure DFS_SDP ($\mathbf{X}^*$, $\mathbf{y}$, $d$)        ($\mathbf{X}^* = \mathbf{X} \setminus \mathbf{Y}$, partial instantiation $\mathbf{y}$, search depth $d$)
        $UB \leftarrow \log O(c \mid \mathbf{e}) + w_{\mathbf{y}} + \sum_{i=d+1}^{m} \max_{x_i} w_{x_i}$        (upperbound for pruning)
        $LB \leftarrow \log O(c \mid \mathbf{e}) + w_{\mathbf{y}} + \sum_{i=d+1}^{m} \min_{x_i} w_{x_i}$        (lowerbound for pruning)
        if $UB < \lambda$ then
            return
        else if $LB \geq \lambda$ then
            $p_{SD} \leftarrow p_{SD} + \Pr(\mathbf{y} \mid \mathbf{e})$; return
            else if $d < m$ then
                for each value $x_{d+1}$ of attribute $X_{d+1}$ do
                    DFS_SDP ($\mathbf{X}^* \setminus \{X_{d+1}\}$, $\mathbf{y}\, x_{d+1}$, $d+1$)

main  :
    global $p_{SD} \leftarrow 0.0$          (initial SDP)
    DFS_SDP ($\mathbf{X}$, $\{\}$, 0)        (initial partial instantiation $\mathbf{y} = \{\}$ and search tree depth $d = 0$)
    return $p_{SD}$
```

This paper is organised as follows. In Section 2 we present the necessary preliminaries. In Section 3, we study the SDP and its relation with threshold T, where in Section 4 we detail its use in PI-explanations. Finally, we end with conclusions and directions for future research in Section 5.

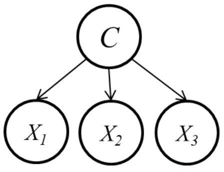
2. Preliminaries

In this paper we consider Bayesian networks $\mathcal{B}$ that define a joint probability distribution $\Pr(\mathbf{V})$ over a finite set of discrete stochastic variables $\mathbf{V}$ (Jensen and Nielsen, 2007). We assume all variables $V \in \mathbf{V}$ to be binary-valued, with v and $\neg v \in V$ indicating the possible value assignments, or instantiations, of V. A capital letter is used to describe a variable or its set of values (the distinction should be clear from the context); bold face is used in case of multiple variables. Lower case letters denote (joint) value assignments. We will write $\mathbf{y} \subseteq \mathbf{x}$ to indicate that $\mathbf{y}$ is a partial instantiation of a subset $\mathbf{Y} \subseteq \mathbf{X}$, consistent with $\mathbf{x}$; we use $\{\}$ to denote an instantiation to an empty set.

We consider Bayesian networks that are used for classification tasks and therefore identify disjoint subsets of variables with special roles: $C \in \mathbf{V}$ is a *class* variable, and $\mathbf{A} = (\mathbf{E} \cup \mathbf{X}) \subset \mathbf{V}$ a set of observable *attributes*, where $\mathbf{E}$ (possibly empty) is already observed, and observations for $\mathbf{X}$ are yet unknown. The Bayesian network classifier now 'decides' to classify instantiation $\mathbf{e} \in \mathbf{E}$ as belonging to class $c \in C$ iff $\Pr(c \mid \mathbf{e}) \geq T$ for some *threshold* T. To measure the confidence in such a threshold-based decision, Choi et al. (2012) define the *Same-Decision Probability* (SDP) as the probability of making this same decision c upon observing attributes $\mathbf{X}$:

$$SDP_{c,T}(\mathbf{X} \mid \mathbf{e}) = \sum_{\mathbf{x} \in \mathbf{X}} I_{c,T}(\mathbf{x} \mid \mathbf{e}) \cdot \Pr(\mathbf{x} \mid \mathbf{e}), \;\; \text{where} \;\; I_{c,T}(\mathbf{x} \mid \mathbf{e}) = \begin{cases} 1 & \text{if } \Pr(c \mid \mathbf{e}\,\mathbf{x}) \geq T \\ 0 & \text{otherwise} \end{cases}$$

The SDP problem is in general PP$^{\text{PP}}$-complete (Choi et al., 2012) and remains NP-hard in a naive Bayesian network (Chen et al., 2014). For computing the SDP two algorithms are available: (1) an



	$\Pr(C)$	$\Pr(x_1 \mid C)$	$\Pr(x_2 \mid C)$	$\Pr(x_3 \mid C)$
c	$2/3$	0.8	0.9	0.7
$\neg c$	$1/3$	0.1	0.4	0.5

Figure 1: A (naive) Bayesian network representing $\Pr(C\ X_1\ X_2\ X_3) = \Pr(C) \cdot \prod_{i=1}^{3} \Pr(X_i \mid C)$

approximate algorithm that provides a *lowerbound* on $\mathit{SDP}_{c,T}(\mathbf{X} \mid \mathbf{e})$, based upon the one-sided Chebyshev inequality for random variable $Q(\mathbf{X}) = \Pr(c \mid \mathbf{e}\,\mathbf{X})$ with variance $\mathrm{Var}(Q(\mathbf{X}))$:

$$\mathit{SDP\text{-}LB}_{c,T}(\mathbf{X} \mid \mathbf{e}) \overset{\text{def}}{=} 1 - \frac{\mathrm{Var}(Q(\mathbf{X}))}{\mathrm{Var}(Q(\mathbf{X})) + (\Pr(c \mid \mathbf{e}) - T)^2} \tag{1}$$

and (2) an *exact* algorithm that performs a depth-first search in a pruned space of instantiations for $\mathbf{X}$. Pseudocode assuming a naive Bayesian network is given in Algorithm 1 and further explained in Section 3.3.1; the algorithm can be generalised by viewing arbitrary networks as naive Bayesian networks with aggregate attributes (Chen et al., 2014). Algorithm 1 performs all computations for the threshold comparisons in the above indicator function in *log-odds* space: $\Pr(c \mid \mathbf{e}\,\mathbf{x}) \geq T \iff \log O(c \mid \mathbf{e}\,\mathbf{x}) \geq \log \frac{T}{1-T} \overset{\text{def}}{=} \lambda$ where $\log O(c \mid \mathbf{e}\,\mathbf{x}) = w_{\mathbf{x}} + \log O(c \mid \mathbf{e})$ follows from

$$\log \frac{\Pr(c \mid \mathbf{e}\,\mathbf{x})}{\Pr(\neg c \mid \mathbf{e}\,\mathbf{x})} = \log \Big(\frac{\Pr(\mathbf{x} \mid c\,\mathbf{e}) \cdot \Pr(c \mid \mathbf{e})}{\Pr(\mathbf{x} \mid \neg c\,\mathbf{e}) \cdot \Pr(\neg c \mid \mathbf{e})} \Big) = \log \frac{\Pr(\mathbf{x} \mid c\,\mathbf{e})}{\Pr(\mathbf{x} \mid \neg c\,\mathbf{e})} + \log \frac{\Pr(c \mid \mathbf{e})}{\Pr(\neg c \mid \mathbf{e})}$$

In a naive Bayesian network, all m attributes are independent given the decision variable, and $w_{\mathbf{x}}$ can be simplified to $w_{\mathbf{x}} = \sum_{i=1}^{m} w_{x_i}$. We take $\mathit{SDP}_{c,\lambda}$ to indicate the use of log-odds.

We now illustrate the above concepts for an example network that serves as a running example.

Example 1 *Consider the example Bayesian network from Figure 1, where* $\mathbf{E} = \emptyset$*. With a threshold of* $T = 0.5$*, the associated classifier produces the decisions for the various instantiations of* $\mathbf{X}$ *shown in Table 1. Prior to observing evidence, c is the most likely class value. We are interested in the probability of making this same decision upon obtaining evidence for* $\mathbf{X}$*. Summing the relevant values of* $\Pr(\mathbf{x})$ *(in bold), we find* $\mathit{SDP}_{c,0.5}(\mathbf{X}) = 0.7107$*, so the* confidence *in our current decision is* 71.07%*. The variance* $\mathrm{Var}(Q(\mathbf{X}))$ *for* $Q(\mathbf{X}) = \Pr(c \mid \mathbf{X})$ *is computed from*

$$\Big[\sum_{\mathbf{x}} \Pr(c \mid \mathbf{x})^2 \cdot \Pr(\mathbf{x}) \Big] - \Pr(c)^2$$

and equals 0.1255*, so the lowerbound on this SDP as computed by the approximate algorithm is* $1 - \frac{0.1255}{0.1255 + (2/3 - 0.5)^2} = 0.1812$*, which severely underestimates the true SDP.*

3. SDP Properties and Threshold Robustness

In this section we study the relation between threshold T and $\mathit{SDP}_{c,T}(\mathbf{X} \mid \mathbf{e})$. To this end, we first establish several theoretical properties of the SDP, and then use these to study the robustness of both exact and approximate SDP computations to changes in T.

	X_1	X_2	X_3	$\Pr(\mathbf{x})$	$\Pr(c \mid \mathbf{x})$	$\log O(c \mid \mathbf{x})$	decision	PI explanations
$\mathbf{x}_8$:	$\neg x_1$	$\neg x_2$	$\neg x_3$	0.0940	0.0426	-4.4919	$\neg c$	$\neg x_1 \neg x_3$, $\neg x_1 \neg x_2$
$\mathbf{x}_7$:	$\neg x_1$	$\neg x_2$	x_3	0.0993	0.0940	-3.2695	$\neg c$	$\neg x_1 \neg x_2$
$\mathbf{x}_6$:	$\neg x_1$	x_2	$\neg x_3$	0.0960	0.3750	-0.7370	$\neg c$	$\neg x_1 \neg x_3$
$\mathbf{x}_5$:	$\neg x_1$	x_2	x_3	**0.1440**	0.5833	0.4854	c	$x_2 x_3$
$\mathbf{x}_4$:	x_1	$\neg x_2$	$\neg x_3$	**0.0260**	0.6154	0.6781	c	x_1
$\mathbf{x}_3$:	x_1	$\neg x_2$	x_3	**0.0473**	0.7887	1.9005	c	x_1
$\mathbf{x}_2$:	x_1	x_2	$\neg x_3$	**0.1507**	0.9558	4.4330	c	x_1
$\mathbf{x}_1$:	x_1	x_2	x_3	**0.3427**	0.9805	5.6554	c	x_1, $x_2 x_3$

Table 1: Predictions, decisions and explanations for the example classifier ($T = 0.5$)

3.1 Theoretical Properties of the SDP

To establish the exact effect of threshold variation on $SDP_{c,T}(\mathbf{X} \mid \mathbf{e})$ it is convenient to consider the order on instantiations for attributes $\mathbf{X}$ that is embedded in the function $f(\mathbf{X}) = \Pr(c \mid \mathbf{e}\,\mathbf{X})$.

Definition 1 *Let $f : \mathbf{X} \to [0,1]$ be a function that maps variable instantiations to probabilities. Then the* probability-induced order *$\preceq_f$ on $\mathbf{X}$ is defined as:*

$$\mathbf{x}_1 \preceq_f \mathbf{x}_2 \iff f(\mathbf{x}_1) \leq f(\mathbf{x}_2)$$

Note that two different instantiations can have the same f-value and thus be equivalent with respect to $\preceq_f$. As a result, the totally ordered set $(\mathbf{X}, \preceq_f)$ can actually have multiple instantiations as maximal element, denoted $\mathbf{x}^\top$, or as minimal element, denoted $\mathbf{x}_\perp$. We can employ these concepts in bounding probabilities.

Lemma 2 *Let $\Pr, c, \mathbf{e}, \mathbf{X}$ and T be as before, and let $\mathbf{x}^\top$ and $\mathbf{x}_\perp$ be maximal and minimal elements, respectively, of $\mathbf{X}$ according to the order $\preceq_f$ induced by $f(\mathbf{X}) = \Pr(c \mid \mathbf{e}\,\mathbf{X})$. Then*

$$\Pr(c \mid \mathbf{e}, \mathbf{x}^\top) \geq \Pr(c \mid \mathbf{e}) \geq \Pr(c \mid \mathbf{e}\,\mathbf{x}_\perp)$$

Proof $\Pr(c \mid \mathbf{e}) = \sum_{\mathbf{x} \in \mathbf{X}} \Pr(c \mid \mathbf{e}\,\mathbf{x}) \cdot \Pr(\mathbf{x} \mid \mathbf{e})$, so $\max_{\mathbf{x} \in \mathbf{X}}\{\Pr(c \mid \mathbf{e}\,\mathbf{x})\} \geq \Pr(c \mid \mathbf{e}) \geq \min_{\mathbf{x} \in \mathbf{X}}\{\Pr(c \mid \mathbf{e}\,\mathbf{x})\}$. ■

We will now derive some properties of the SDP. We first show that the SDP is always non-zero.

Proposition 3 *Let $\Pr, c, \mathbf{e}, \mathbf{X}$ and T be as before, then $SDP_{c,T}(\mathbf{X} \mid \mathbf{e}) > 0$.*

Proof There always exists an $\mathbf{x}_i$ with $\Pr(\mathbf{x}_i \mid \mathbf{e}) > 0$ such that $\Pr(c \mid \mathbf{e}\,\mathbf{x}_i) \geq \Pr(c \mid \mathbf{e}) \geq T$. Hence $SDP_{c,T}(\mathbf{X} \mid \mathbf{e}) > 0$. ■

We now demonstrate that the SDP can always become one upon changing the threshold.

Proposition 4 *Let $\Pr, c, \mathbf{e}$, and $\mathbf{X}$ be as before, then $\exists\, T : \; SDP_{c,T}(\mathbf{X} \mid \mathbf{e}) = 1$.*

Proof Let $\mathbf{x}_\perp$ be a minimal element of $(\mathbf{X}, \preceq_f)$ with $\preceq_f$ induced by $f(\mathbf{X}) = \Pr(c \mid \mathbf{e}\,\mathbf{X})$, then by Lemma 2: $\Pr(c \mid \mathbf{e}) \geq \Pr(c \mid \mathbf{e}\,\mathbf{x}_\perp)$. Therefore, for any $T \leq \Pr(c \mid \mathbf{e}\,\mathbf{x}_\perp)$ we have $SDP_{c,T}(\mathbf{X} \mid \mathbf{e}) = \sum_{\mathbf{x} \in \mathbf{X}} \Pr(\mathbf{x} \mid \mathbf{e}) = 1$. ■

We observe from the above proof that, assuming $\Pr(\mathbf{x}_\perp \mid \mathbf{e}) > 0$, $T \leq \Pr(c \mid \mathbf{e}\,\mathbf{x}_\perp)$ is not only a sufficient, but also a necessary condition for obtaining an SDP of one.

We can now completely specify the relation between SDP and threshold value.

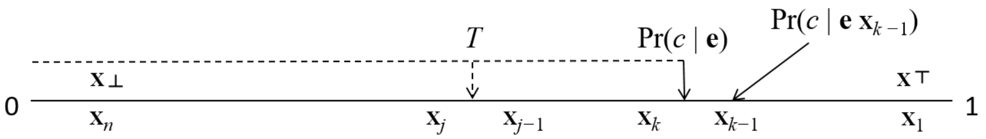


Figure 2: Probability line with each $\mathbf{x}_i$ indicating the location of $\Pr(c \mid \mathbf{e}\,\mathbf{x}_i)$; in addition the range of values for T is shown that would give the current decision c for evidence $\mathbf{e}$

Proposition 5 *Let* $\Pr, c, \mathbf{e}, \mathbf{X}$ *and* T *be as before. Let* $f(\mathbf{X}) = \Pr(c \mid \mathbf{e}\,\mathbf{X})$ *induce the following order on* $\mathbf{X}$*:* $\mathbf{x}_n \preceq_f \ldots \preceq_f \mathbf{x}_1$. *If* $\Pr(c \mid \mathbf{e}) \in [\Pr(c \mid \mathbf{e}\,\mathbf{x}_k), \Pr(c \mid \mathbf{e}\,\mathbf{x}_{k-1})]$, $k \in [2, n]$ *then*

$$SDP_{c,T}(\mathbf{X} \mid \mathbf{e}) = \begin{cases} \sum_{i=1}^{j-1} \Pr(\mathbf{x}_i \mid \mathbf{e}) & \text{if } \Pr(c \mid \mathbf{e}\,\mathbf{x}_j) < T \leq \Pr(c \mid \mathbf{e}\,\mathbf{x}_{j-1}),\ k \leq j \leq n \\ 1 & \text{if } T \leq \Pr(c \mid \mathbf{e}\,\mathbf{x}_n) \end{cases}$$

Proof Consider Figure 2 which visualises the situation under consideration. The SDP of 1 follows from Proposition 4 and $\mathbf{x}_n$ being a minimal element of $(\mathbf{X}, \preceq_f)$. The remainder of the result follows from $\mathbf{x}_{j-1}$ being the smallest element (according to $\preceq_f$) giving rise to decision c. ■

The above proofs build on the assumption that we have at least some knowledge of the order on instantiations of $\mathbf{X}$ embedded in $\Pr(c \mid \mathbf{e}\,\mathbf{X})$. In the current context these probabilities are computed, rather than specified, and establishing the order by computing $\Pr(c \mid \mathbf{e}\,\mathbf{x})$ for all $\mathbf{x} \in \mathbf{X}$ is not a very feasible approach. Sometimes, however, as shown in Section 3.4, we can efficiently exploit properties of the network's distribution to establish at least a sufficing partial order on instantiations.

3.2 Threshold Robustness and Improved Lowerbound for Approximate SDP

We consider the SDP-lowerbound $SDP\text{-}LB_{c,T}(\mathbf{X} \mid \mathbf{e})$ given in Equation 1, which immediately provides for studying the relation between the lowerbound and the threshold T. Figure 3(a) shows the lowerbound as a function of T for five different prior probabilities $\Pr(c)$ of the example Bayesian network from Figure 1. Note that all functions should be undefined for $T > \Pr(c \mid \mathbf{e})$ since otherwise we would be interested in the SDP for decision $\neg c$. Also note that the functions have a rather steep drop off and become zero for $T = \Pr(c \mid \mathbf{e})$, where the current decision is exactly on the decision boundary defined by T.

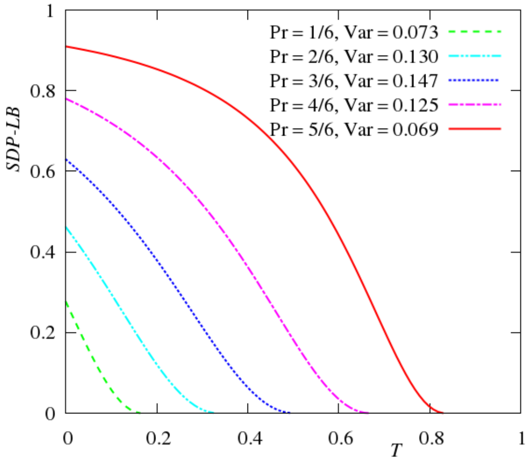
To get an impression of the quality of the lowerbound, we further provide in Figure 3(b) the SDP in our example classifier for different choices of T and prior $\Pr(c)$. As already noted by (Choi et al., 2012), the lowerbound becomes rather weak as it approaches zero. This is perhaps not surprising given Proposition 3; in fact, its proof provides a means for improving the lowerbound.

Corollary 6 *Let* $\Pr, c, \mathbf{e}, \mathbf{X}$ *and* T *be as before. Let* $\mathbf{x}^\top$ *be a maximal element of* $(\mathbf{X}, \preceq_f)$ *with* $\preceq_f$ *induced by* $f(\mathbf{X}) = \Pr(c \mid \mathbf{e}\,\mathbf{X})$*, where* $\mathbf{X}$ *is restricted to* $\mathbf{x}$ *with* $\Pr(\mathbf{x} \mid \mathbf{e}) > 0$. *Then*

$$SDP_{c,T}(\mathbf{X} \mid \mathbf{e}) \geq \max\{\Pr(\mathbf{x}^\top \mid \mathbf{e}),\ SDP\text{-}LB_{c,T}(\mathbf{X} \mid \mathbf{e})\}$$

The probability $\Pr(\mathbf{x}^\top \mid \mathbf{e})$ is a tighter lowerbound for the SDP than $SDP\text{-}LB$ as long as

$$T \in \Big(\ \Pr(c \mid \mathbf{e}) - \sqrt{\frac{\Pr(\mathbf{x}^\top \mid \mathbf{e})}{1 - \Pr(\mathbf{x}^\top \mid \mathbf{e})} \cdot \mathrm{Var}(Q(\mathbf{X}))},\ \ \Pr(c \mid \mathbf{e})\Big]$$



(a)

		$\Pr(c)$			
$SDP_{c,T}(\mathbf{X})$		2/6	3/6	4/6	5/6
T	0.25	0.4953	0.6030	0.8067	0.8933
	0.50	×	0.4230	0.7107	0.8933
	0.75	×	×	×	0.8183
$\Pr(\mathbf{x}^{\top})$		0.1813	0.2620	0.3427	0.4233
$\nearrow$ for $T >$		0.1637	0.2713	0.4109	0.6088

(b)

Figure 3: For different values of $\Pr(c)$ in our example network: (a) *SDP-LB* as a function of threshold T, (b) the true SDP for different choices of T, and the range of T values for which bound $\Pr(\mathbf{x}^{\top})$ improves on ($\nearrow$) *SDP-LB*. Values are only specified if $\Pr(c) \geq T$.

Example 2 *For our example network, for different values of* $\Pr(c)$*, the probability* $\Pr(\mathbf{x}^{\top}) = \Pr(x_1\ x_2\ x_3)$ *is shown in Figure 3(b). We note that this new bound is still weak compared to the true SDP values (also shown for various values of* T*), yet improves the bound from Equation 1 for a certain range of* T*-values. As an example (not shown in the table), consider* $\Pr(c) = 1/6$*. We find* $\Pr(\mathbf{x}^{\top}) = 0.1007$ *which is better than SDP-LB for* $0.0762 < T \leq 1/6$.

3.3 Establishing Threshold Robustness for Exact SDP

Although Proposition 5 specifies the exact relation between threshold values and SDP, it does not directly provide us with a feasible way of establishing this relation. We therefore narrow the scope of the problem to studying the following question: given an SDP and threshold T, what is the range of values within which we can change T without changing the SDP? We will argue that some additional bookkeeping within the exact algorithm for computing the SDP by Chen et al. (2014) serves to establish the interval of T-values for which the computed SDP holds. In order to understand the necessary changes, we first highlight some aspects of the original algorithm from the viewpoint of probability-induced orders.

3.3.1 SDP ALGORITHM FROM AN ORDERING PERSPECTIVE

From the pseudocode in Algorithm 1, we have that the SDP-algorithm consists of a depth-first search in a search tree that enumerates all possible instantiations of the set $\mathbf{X}$ under consideration. To this end, each attribute has its own layer of nodes in the search tree, and each edge corresponds to a value of the associated attribute. An example search tree is shown in Figure 4(a). For the purpose of more efficient SDP-computation, the search tree is pruned using upperbound *UB* and lowerbound *LB* in lines 1 and 2 of the `DFS-SDP` subprocedure. These bounds do not bound the SDP, but rather

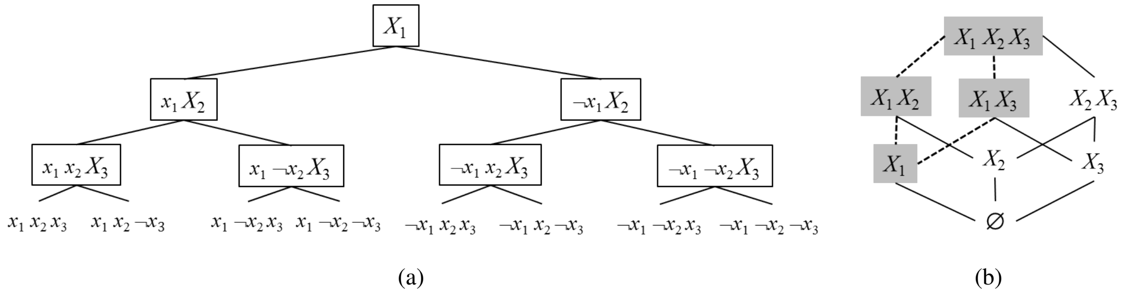


Figure 4: (a) Search tree for three binary attributes. Nodes show the associated attribute as well as the partial instantiation on the path; (b) Subset-lattice for the attributes (grey: sublattice)

the log posterior odds for partial instantiations of $\mathbf{X}$. More specifically, at depth $d = 0$, UB equals $\max_{\mathbf{x} \in \mathbf{X}}\{\log O(c \mid \mathbf{e}\,\mathbf{x})\}$ and $LB = \min_{\mathbf{x} \in \mathbf{X}}\{\log O(c \mid \mathbf{e}\,\mathbf{x})\}$. Or, in our terminology:

$$\text{at } d = 0 : \;\; UB = \log O(c \mid \mathbf{e}\,\mathbf{x}^{\top}) \geq \log O(c \mid \mathbf{e}) \geq \log O(c \mid \mathbf{e}\,\mathbf{x}_{\perp}) = LB$$

where $\mathbf{x}^{\top}$ and $\mathbf{x}_{\perp}$ are maximal and minimal elements, respectively, of $(\mathbf{X}, \preceq_f)$ with $\preceq_f$ induced by $f(\mathbf{X}) = \Pr(c \mid \mathbf{e}\,\mathbf{X})$. At depth $j > 0$, the partial instantiation $\mathbf{y}$ covers j variables from the original set of variables $\mathbf{X}$ and the set $\mathbf{X}^*$ consists of the remaining $m - j$ attributes. Therefore:

$$\text{at } d = j : \;\; UB = \log O(c \mid \mathbf{e}\,\mathbf{y}\,\mathbf{x}^{*\top}) \geq \log O(c \mid \mathbf{e}\,\mathbf{y}) \geq \log O(c \mid \mathbf{e}\,\mathbf{y}\,\mathbf{x}^{*}{}_{\perp}) = LB$$

where $\mathbf{x}^{*\top}$ and $\mathbf{x}^{*}{}_{\perp}$ are maximal and minimal elements, respectively, of $(\mathbf{X}^*, \preceq_h)$ with order $\preceq_h$ induced by $h(\mathbf{X}^*) = \Pr(c \mid \mathbf{e}\,\mathbf{y}\,\mathbf{X}^*)$. Now, since $\mathbf{y}$ and $\mathbf{x}^*$ together form a complete instantiation of $\mathbf{X}$, the order induced on $\mathbf{X}^*$ by $h(\mathbf{X}^*)$ is in fact the *same* as the order induced on $\mathbf{y}\mathbf{X}^*$ by $f(\mathbf{X})$.

The upper- and lowerbounds are used to prevent unnecessary traversal of subtrees. Indeed, if the log-odds threshold λ exceeds UB, then none of the instantiations represented by the current subtree will contribute to the SDP, so further traversal of the subtree is unnecessary; similarly, if $LB \geq \lambda$ then *all* instantiations from the subtree will definitely contribute to the SDP, so further exploration of the subtree is forestalled and $\sum_{\mathbf{x}^* \in \mathbf{X}^*} \Pr(\mathbf{x}^*\,\mathbf{y} \mid \mathbf{e}) = \Pr(\mathbf{y} \mid \mathbf{e})$ is added to the SDP.

3.3.2 EXPLOITING THE SDP ALGORITHM FOR THRESHOLD ROBUSTNESS

From Proposition 5 we have that threshold T can be varied without changing $SDP_{c,T}(\mathbf{X} \mid \mathbf{e})$, if it remains within the interval $(\ \Pr(c \mid \mathbf{e}\,\mathbf{x}_j), \min\{\Pr(c \mid \mathbf{e}),\ \Pr(c \mid \mathbf{e}\,\mathbf{x}_{j-1})\}\ ]$ of posterior probabilities for the two subsequent instantiations $\mathbf{x}_j$ and $\mathbf{x}_{j-1}$ of $\mathbf{X}$. Instantiation $\mathbf{x}_{j-1}$ now is the 'lowest' instantiation resulting in a posterior probability of T or higher. Similarly, instantiation $\mathbf{x}_j$ is the 'highest' instantiation that results in a posterior lower than T. The upper- and lowerbounds currently computed by the SDP algorithm suffice for establishing this robustness interval for the threshold. Since Algorithm 1 works in log-odds space, we will establish bounds on λ rather than T.

Proposition 7 *Let* $\Pr, c, \mathbf{e}, \mathbf{X}$, *and* λ *be as before, and let* $\preceq_f$ *be the order induced by* $f(\mathbf{X}) = \Pr(c \mid \mathbf{e}\,\mathbf{X})$. *Suppose* $\log O(c \mid \mathbf{e}\,\mathbf{x}_j) < \lambda \leq \log O(c \mid \mathbf{e}\,\mathbf{x}_{j-1})$ *for two successive instantiations* $\mathbf{x}_j \preceq_f \mathbf{x}_{j-1}$. *Let* $UB(i)$ *and* $LB(i)$ *be the values for* UB *and* LB, *respectively, computed by*

Algorithm 1 for every visited *node i in the search tree. Then,*

$$\log O(c \mid \mathbf{e}\,\mathbf{x}_j) = \max_i\{\,UB(i) \mid UB(i) < \lambda\} \ \textit{and}\ \log O(c \mid \mathbf{e}\,\mathbf{x}_{j-1}) = \min_i\{LB(i) \mid LB(i) \geq \lambda\}$$

Proof In a visited leaf node i representing some $\mathbf{x}_i \in \mathbf{X}$, we have that $LB(i) = \log O(c \mid \mathbf{e}\,\mathbf{x}_i) = UB(i)$. Therefore, if all leaves of the search tree are visited during search, the proposition is trivially true. We now prove that the proposition remains true upon pruning. If $UB(i) < \lambda$ then the search tree is pruned because no instantiation in the subtree will contribute to the SDP; in this case $UB(i)$ is associated with an instantiation $\mathbf{x}_i \preceq_f \mathbf{x}_j$. If $\log O(c \mid \mathbf{e}\,\mathbf{x}_i) = \log O(c \mid \mathbf{e}\,\mathbf{x}_j)$ then we have found our threshold *lower*bound; otherwise $\mathbf{x}_j$ must be in a different subtree. With a similar argument we can prove that we will find our threshold *upper*bound even upon pruning. ■

From the above proposition we have that the following adaptation of Algorithm 1 suffices for establishing the interval $(\,\lambda LB, \lambda UB\,]$ within which threshold λ can be varied without changing the SDP:

```
add to **main**:   $\lambda LB \leftarrow -\infty;\ \lambda UB \leftarrow \infty$          (initial $\lambda$ lower- and upperbounds)
                   **return** $(\,\lambda LB,\ \min\{\log O(c \mid \mathbf{e}), \lambda UB\}\,]$        (robustness interval)
add to line 4:     $\lambda LB \leftarrow \max\{\,UB, \lambda LB\}$
add to line 6:     $\lambda UB \leftarrow \min\{LB, \lambda UB\}$
```

Example 3 *In our example classifier with threshold $\lambda = 0$ ($T = 0.5$), we have that $\log O(c) = 1$. The threshold robustness interval found for $SDP_{c,\lambda}(\mathbf{X} \mid \mathbf{e})$ now is $(\,-0.7370,\ \min\{1, 0.4854\}\,]$. For thresholds within this interval the SDP is guaranteed to remain the same. Lowering the threshold to $\lambda' \leq -0.7370$ will simply increase the SDP. Since $\log O(c) > \lambda UB$, the SDP will decrease for $0.4854 < \lambda' \leq 1$. Increasing λ' beyond 1 changes the SDP of interest to $SDP_{\neg c,\lambda'}$.*

3.4 Probability-Induced Orders and Monotonicity

From the above we can conclude that Algorithm 1 applies to Bayesian networks in general if we rephrase the upper- and lowerbounds in terms of maximal and minimal elements of $(\mathbf{X}^*, \preceq_f)$ for subsets $\mathbf{X}^*$ of attributes. In addition, we have seen that the probability of maximal element $\mathbf{x}^\top$ sometimes improves the $SDP\text{-}LB$ approximation. Of course, in order to exploit this, we need to be able to easily establish maximal and minimal elements. In this section we show that in certain cases, we can easily establish a partial order consistent with $\preceq_f$ that suffices for establishing the maximal and minimal elements relevant for our upper- and lowerbound computations. This problem is related to that of establishing monotonicity properties.

In Bayesian networks, the concept of monotonicity typically pertains to the posterior distributions $\Pr(C \mid \mathbf{A})$ (van der Gaag et al., 2004). More specifically, it pertains to cumulative distributions over C, thereby implicitly assuming an order $<$ on the values of C. Since we are specifically interested in orderings, we will make any such assumptions explicit. A network $\mathcal{B}$ is now said to be (positively) *monotone* in $\mathbf{A}$ with respect to total order $<$ on C and partial order $\preceq$ on $\mathbf{A}$, if for all values c_k of C and instantiations $\mathbf{a}, \mathbf{a}'$ of $\mathbf{A}$,

$$\mathbf{a} \preceq \mathbf{a}' \Rightarrow \Pr(C \leq c_k \mid \mathbf{a}') \leq \Pr(C \leq c_k \mid \mathbf{a})$$

We first prove that any monotonicity-inducing order is consistent with $\preceq_f$ for a certain value of C.

Proposition 8 *Let $\mathcal{B}$, C, and $\mathbf{A}$ be as before. Suppose $\mathcal{B}$ is monotone in $\mathbf{A}$ with respect to partial order $\preceq_m$ on $\mathbf{A}$ and total order $<$ on C. Let c be the maximal element of $(C, <)$. Moreover, let $\preceq_f$ be the order induced by $\Pr(c \mid \mathbf{A})$. Then for any two instantiations $\mathbf{a}$ and $\mathbf{a}'$ of $\mathbf{A}$, $\mathbf{a} \preceq_m \mathbf{a}' \Rightarrow \mathbf{a} \preceq_f \mathbf{a}'$.*

Proof Assume $\mathbf{a} \preceq_m \mathbf{a}'$. Then $\Pr(C \leq c_k \mid \mathbf{a}') \leq \Pr(C \leq c_k \mid \mathbf{a})$ for all values c_k of C. This implies for $C = c$ that $\Pr(c \mid \mathbf{a}') \geq \Pr(c \mid \mathbf{a})$ and therefore $\mathbf{a} \preceq_f \mathbf{a}'$. ■

The partial order $\preceq_m$ now suffices for establishing the relevant maximal and minimal elements.

Proposition 9 *Let $\mathcal{B}$, C and $\mathbf{A} = \mathbf{E} \cup \mathbf{Y} \cup \mathbf{X}^*$ be as before, and let $\mathbf{ey}$ be any instantiation of $\mathbf{E} \cup \mathbf{Y}$. Let $\mathcal{B}$ be monotone in $\mathbf{A}$ with respect to partial order $\preceq_m$ and total order $<$ on C. Let c be the maximal element of $(C, <)$. Then $(\mathbf{A}, \preceq_m)$ provides for establishing maximal and minimal elements of $(\mathbf{X}^*, \preceq_h)$ with order induced by $h(\mathbf{X}^*) = \Pr(c \mid \mathbf{e}\,\mathbf{y}\,\mathbf{X}^*)$.*

Proof $(\mathbf{A}, \preceq_m)$ can be represented as a lattice, called an *assignment lattice* (van der Gaag et al., 2006). The sub-lattice induced by a partial instantiation $\mathbf{ey}$ then has $\mathbf{eyx}^{*\top}$ and $\mathbf{eyx}^{*}_{\perp}$ as maximal and minimal elements, respectively. ■

Deciding if a network is monotone is in general a coNP$^{\mathrm{PP}}$-complete problem (van der Gaag et al., 2004). It can, however, be approximated using the QPN[1] concept of qualitative influence (Wellman, 1990). For an arc $A \rightarrow C$ in the network graph, its *qualitative influence* is said to be *positive*, written $S^+(A, C)$, if for all values c_k of C and all values $a < a'$ of A, we have that $\Pr(C \geq c_k \mid a'\,\mathbf{z}) \geq \Pr(C \geq c_k \mid a\,\mathbf{z})$ for any instantiation $\mathbf{z}$ for $\mathbf{Z} = \mathrm{Par}(C) \setminus \{A\}$, the set of parents of C other than A. Note that this definition assumes a total order on the values of each variable; these total orderings together induce a partial order on all joint instantiations of the variables. Qualitative influences exhibit a number of properties that allow for deriving influences between any pair of variables from those specified for arcs. Positive qualitative influences suffice for concluding monotonicity.

Proposition 10 (van der Gaag et al. (2004)) *Let $\mathcal{B}$, C, and $\mathbf{A}$ be as before. Let $\preceq_q$ be the partial order on $\mathbf{A}$ induced by the total orders on the values of the individual variables. If $S^+(A_i, C)$ for each $A_i \in \mathbf{A}$ then network $\mathcal{B}$ is monotone in $\mathbf{A}$ with respect to $\preceq_q$.*

Combining all the above results we conclude that if we can define total orders on the values of each variable such that $S^+(A_i, C)$ for all $A_i \in \mathbf{A}$, then the partial order $\preceq_q$ on $\mathbf{A}$ induced by these total orders suffices for establishing any maximal or minimal elements we need for our upper- and lowerbound computations. For a naive Bayesian network classifier with binary-valued variables we can *always* find such orders.

Proposition 11 *Let $\Pr, C$ and $\mathbf{A}$ be as before. For binary-valued variables in a naive Bayesian classifier, we can always define orderings $<$ on their values such that $S^+(A_i, C)$ for all $A_i \in \mathbf{A}$.*

Proof Without loss of generality, suppose we require $\neg c < c$ for the values of C. For each attribute $A_i \in \mathbf{A}$ we have that either $\Pr(a_i \mid c) \geq \Pr(a_i \mid \neg c)$ or $\Pr(\neg a_i \mid c) \geq \Pr(\neg a_i \mid \neg c)$. So $S^+(C, A_i)$ holds in the former case if we take $\neg a_i < a_i$, and in the latter for $a_i < \neg a_i$. Since each A_i is only directly connected to C, its value ordering can be chosen independent from the choice for

[1] Qualitative Probabilistic Network: a qualitative abstraction of a Bayesian network.

the other attributes. The result now follows from the property of symmetry of qualitative influences for binary-valued variables (Renooij, 2001): $S^{+}(C, A_i) \rightarrow S^{+}(A_i, C)$. ■

Example 4 *From the conditional probability tables of our example network in Figure 1 we have that* $\Pr(x_i \mid c) \geq \Pr(x_i \mid \neg c)$ *for all* X_i*,* $i = 1, 2, 3$*. With* $\neg c < c$ *and* $\neg x_i < x_i$ *for all* X_i *we therefore have* $S^{+}(C, X_i)$ *and* $S^{+}(X_i, C)$*. The network is thus monotone in* $\mathbf{X}$ *with respect to the partial order* $\preceq_q$ *induced by the orders* $<$*.* $(\mathbf{X}, \preceq_q)$ *has maximal element* $\mathbf{x}^{\top} = x_1 x_2 x_3$ *and minimal element* $\mathbf{x}_{\perp} = \neg x_1 \neg x_2 \neg x_3$*. These are also maximal and minimal elements of* $(\mathbf{X}, \preceq_f)$ *with order* $\preceq_f$ *induced by* $f(\mathbf{X}) = \Pr(c \mid \mathbf{X})$*.*

4. SDP and Prime Implicant Explanations

As mentioned before, the SDP has proven its use in various contexts. Here we will argue that our theoretical analysis of the SDP provides for another application: that of computing explanations.

Shih et al. (2018) propose to take the concept of *prime implicant* from Boolean logic to provide a symbolic means for explaining Bayesian network classifiers. A prime-implicant (PI) explanation for an instantiation of attributes that results in a given decision then equals a minimal partial instantiation that renders the values of the remaining attributes irrelevant to the decision. A PI-explanation therefore has an SDP of 1 (Shih et al., 2018). However, not every partial instantiation with an SDP of 1 is a PI-explanation. We can define a PI-explanation in terms of the SDP as follows:

Definition 12 *Let* $\Pr, c, \mathbf{A}$*, and* T *be as before. A PI-explanation for decision* c *is an instance* $\mathbf{a}'$ *for* $\mathbf{A}' \subseteq \mathbf{A}$ *such that* $SDP_{c,T}(\mathbf{A} \setminus \mathbf{A}' \mid \mathbf{a}') = 1$ *and there exists no* $\mathbf{a}'' \subset \mathbf{a}'$ *with this property.*

Numerous algorithms exist for computing prime implicants of logic formulas and the problem is already NP-hard for just a single prime implicant (Palopoli et al., 1999). For the purpose of explaining Bayesian network classifiers, Shih et al. (2018) propose two algorithms: one that computes a prime implicant *cover* for all instantiations, and one that computes the prime implicants for a single instantiation. In both cases, the Bayesian network classifier is first compiled into an Ordered Binary Decision Diagram (OBDD), which is a tractable representation of the decision function that maps instantiations of attributes into yes/no decisions. The clear dependency between threshold and decision is lost in this representation. In fact, a different value for T can result in a different OBDD.

The exact relation between PI-explanations and the SDP, together with our concept of probability-induced ordering on instantiations provide for an alternative way of computing PI-explanations. We provide a sketch of an algorithm to this end, which is illustrated in Example 5. As search structure we use the subset lattice for poset $(\mathcal{P}(\mathbf{A}), \subseteq)$, where $\mathcal{P}(\mathbf{A})$ denotes the powerset of $\mathbf{A}$; an example lattice for $\mathbf{A} = \{X_1, X_2, X_3\}$ is shown in Figure 4(b). To compute a PI-explanation for instance $\mathbf{a}$, we associate with each lattice element $\mathbf{Y} \subseteq \mathbf{A}$ the probability $\Pr(c \mid \mathbf{y}\ \mathbf{a^*}_{\perp})$, where $\mathbf{y} \subseteq \mathbf{a}$, $\mathbf{A}^* = \mathbf{A} \setminus \mathbf{Y}$, and $\mathbf{a^*}_{\perp}$ is a minimal element of $(\mathbf{y}\mathbf{A}^*, \preceq_f)$ with order $\preceq_f$ induced by $\Pr(c \mid \mathbf{A})$. We traverse the lattice in a breadth-first, bottom-up manner, similar to algorithms typically employed for item set mining (see e.g. Agrawal and Srikant (1994)). We check if $\Pr(c \mid \mathbf{y}\ \mathbf{a^*}_{\perp}) \geq T$; if so, then $SDP_{c,T}(\mathbf{A}^* \mid \mathbf{y}) = 1$ and therefore $\mathbf{y}$ is a PI-explanation for any instance $\mathbf{a} \supseteq \mathbf{y}$. Since the SDP for any such non-minimal $\mathbf{a}$ will equal 1 as well, we can and should prune all supersets of $\mathbf{Y}$ from the lattice. If $\Pr(c \mid \mathbf{a}_{\perp})$, associated with the empty set, already exceeds the threshold than any instance will result in the current decision.

Like before, we can keep track of the lowerbounds computed in the process, in order to again provide an upperbound for the threshold robustness interval within which the current PI-explanations are guaranteed. The described process can be repeated for decision $\neg c$ and corresponding threshold $1 - T$ (note that the classifier makes decision $\neg c$ only if $\Pr(\neg c \mid \mathbf{a})$ is *strictly* larger than $1 - T$, so all lowerbound threshold comparisons should now use strict inequality).

Example 5 *Consider our example classifier and instance* $\mathbf{x} = x_1\ x_2\ x_3$. *With threshold* $T = 0.5$*, the decision corresponding with this instance is c. Associated with lattice element* $\emptyset$ *is the lowerbound* $\Pr(c \mid \neg x_1\ \neg x_2\ \neg x_3) = 0.0426$ *(see Table 1); since this lowerbound is below T, we continue upwards in the lattice. With lattice element* X_1*, we associate* $\Pr(c \mid x_1\ \neg x_2\ \neg x_3) = 0.6154 \geq T$. *We conclude that* $SDP_{c,T}(X_2\ X_3 \mid x_1) = 1$ *and therefore* x_1 *is a PI-explanation for all instances* $x_1 X_2 X_3$. *The remainder of the lattice for supersets of* X_1 *is then pruned (grey in Figure 4(b)). For lattice elements* X_2 *and* X_3 *we find that their respective lowerbounds are below* T*; we therefore proceed with the only remaining element* $X_2 X_3$. *Since* $\Pr(c \mid x_2\ x_3\ \neg x_1) = 0.5833 \geq T$*, we conclude that* $x_2 x_3$ *is a PI-explanation for any instance* $X_1 x_2 x_3$. *Moreover,* 0.5833 *is the upperbound for* T*, above which PI-explanations may change.*

We repeat the process for instance $\mathbf{x} = \neg x_1 \neg x_2 \neg x_3$ *with current decision* $\neg c$. *Now we have to proceed to subsets of size two before finding lowerbounds that exceed* $1 - T = 0.5$. *For example, for lattice element* $X_1 X_2$ *we have that* $\Pr(\neg c \mid \neg x_1\ \neg x_2\ x_3) = 1 - 0.0940 > 1 - T$. *Therefore,* $SDP_{\neg c, 1-T}(X_3 \mid \neg x_1\ \neg x_2) = 1$ *and* $\neg x_1 \neg x_2$ *is a PI-explanation for all instances* $\neg x_1 \neg x_2 X_3$. *The lowest lowerbound encountered equals* $1 - 0.3750 = 0.6250$ *and is a strict upperbound on* $1 - T$. *The threshold robustness interval for* T *that guarantees the validity of all PI-explanations found therefore equals* $(\ 1 - 0.6250,\ 0.5833\ ]$. *Table 1 lists all PI-explanations for our example network.*

Note that in our example the computation of PI-explanations for just two instances actually provides for establishing the PI-explanations for *all* instances.

5. Conclusions and Further Research

In this paper we studied the SDP and existing algorithms for computing it. We provided properties of the SDP by assuming knowledge of a probability-induced order $\preceq_f$ on the instantiations under consideration. This served to improve the lowerbound approximation of the SDP on the one hand, and also provides another option for turning Algorithm 1 into a version applicable beyond the scope of naive Bayesian classifiers. Moreover, it allowed us to detail the relation between SDP and PI-explanations, thereby opening up the possibility of alternative ways of computing PI-explanations. Most importantly, our analyses provide a means for efficiently establishing a threshold robustness interval that captures the range of threshold values for which SDP and PI-explanations are guaranteed to remain unchanged. Since the interval exactly captures the probability range in which the classifier shifts from one decision to another, it also applies to other measures that are threshold-dependent, such as accuracy.

The presentation of our results relies mostly on a theoretical ordering that is not generally available in practice. We have argued, however, that sufficient information about this ordering can be established if the network is monotone in its attributes, which can always be easily achieved for binary-valued naive Bayesian networks. Although the paper assumes all variables to be binary-valued, the latter result is the only one that actually builds on assumptions that are otherwise not guaranteed (Woudenberg, 2016).

In this paper we have studied the robustness of the SDP and demonstrated its use as a tool in yet another context of application. In future research we want to explore possible other applications of the SDP and investigate the broader applicability of our probability-induced orderings and their relation with monotonicity.

References

R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th Conference on Very Large Data Bases (VLDB)*, pages 487–499, 1994.

S. Chen, A. Choi, and A. Darwiche. Algorithms and applications for the Same-Decision Probability. *Journal of Artificial Intelligence Research*, 49:601–633, 2014.

S. Chen, A. Choi, and A. Darwiche. Value of information based on decision robustness. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, pages 3503–3510, 2015.

A. Choi, Y. Xue, and A. Darwiche. Same-Decision Probability: A confidence measure for threshold-based decisions. *International Journal of Approximate Reasoning*, 53:1415–1428, 2012.

Y. Choi, A. Darwiche, and G. Van den Broeck. Optimal feature selection for decision robustness in Bayesian networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1554–1560, 2017.

F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer Verlag, 2007.

L. Palopoli, F. Pirri, and C. Pizzuti. Algorithms for selective enumeration of prime implicants. *Artificial Intelligence*, 111:41–72, 1999.

S. Renooij. *Qualitative Approaches to Quantifying Probabilistic Networks*. PhD thesis, Utrecht University, The Netherlands, 2001.

A. Shih, A. Choi, and A. Darwiche. A symbolic approach to explaining Bayesian network classifiers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5103–5111, 2018.

L. C. van der Gaag and V. M. H. Coupé. Sensitivity analysis for threshold decision making with Bayesian belief networks. In *AI*IA 99: Advances in Artificial Intelligence*, Lecture Notes in Artificial Intelligence, pages 37–48, Berlin, 1999. Springer-Verlag.

L. C. van der Gaag, H. L. Bodlaender, and A. J. Feelders. Monotonicity in Bayesian networks. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 569–576. AUAI Press, 2004.

L. C. van der Gaag, S. Renooij, and P. L. Geenen. Lattices for studying monotonicity of Bayesian networks. In *Proceedings of the 3rd Workshop on Probabilistic Graphical Models (PGM)*, pages 99–106, 2006.

M. P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44:257 – 303, 1990.

S. D. Woudenberg. Personal communication, 2016.