On Lurking Dependencies and Naive Bayesian Classifiers

Barbara F.I. Pieters Linda C. van der Gaag

Department of Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands

Abstract

Naive Bayesian classifiers are widely used in a large range of application domains, generally showing good performance despite their strong underlying independency assumptions. For some types of dependency among the feature variables in a data set, researchers have studied the effects on a classifier's performance. The effects of other types of dependency are largely unexplored as yet. As a first step into the investigation of such effects, we demonstrate that a data set can give rise to a naive Bayesian classifier with quite counterintuitive behaviour. To explain the observed behaviour, we exploit qualitative probabilistic concepts and attribute the counterintuitive results to lurking dependencies among the feature variables involved.

1 Introduction

Nowadays a multitude of methods, algorithms and associated software are available for learning stochastic models from a collection of gathered data. Among these are methods for constructing Bayesian network classifiers [1]. These models include a designated variable of interest, called the class variable, and multiple feature variables, each of which is related directly to the class variable. Especially naive Bayesian classifiers have become quite popular for classification purposes. These classifiers build upon the assumption that all feature variables are mutually independent whenever a value for the class variable is known. Naive Bayesian classifiers are quite easy to construct from a collection of data and, despite their strong underlying assumptions of independency, show a tendency to outperform more complex models [2].

In general, a data set from which a naive Bayesian classifier is to be constructed may not exhibit the independency properties assumed by the classifier's learning algorithms, that is, the data set may embed dependencies among the recorded feature variables in view of a particular value of the class variable. Yet, good classification performance is generally observed also for such data sets, even in the presence of quite strong dependencies among the feature variables. In view of this finding, researchers have investigated the effects of the presence of particular types of dependency on a classifier's performance. Several researchers have studied, for example, the effects of dependencies that originate from redundancy among the feature variables. Insights from these studies have resulted in methods for feature selection [3, 4, 5], which aim at removing redundancies from the classifier and thereby further enhancing its classification performance. In general, the dependencies among the feature variables embedded in a data set may not all be attributable to information redundancy. Some dependencies may originate for example from a hidden common cause of two feature variables. The effects that such dependencies can have on the classification performance of a naive Bayesian classifier are largely unexplored as yet.

In this paper, we demonstrate that a data set embedding a particular type of dependency among its feature variables can give rise to a naive Bayesian classifier with quite counterintuitive behaviour. To explain the observed behaviour, we exploit qualitative probabilistic concepts. We observe that a particular feature variable can have a positive influence on the class variable, in the sense that entering a higher value for the feature variable will result in higher values for the class variable becoming more likely; alternatively, a feature variable can have a negative influence indicating that its higher values make the higher values of the class variable less likely. We will show that the presence of a specific type of dependency among the feature variables in a data set can actually give rise to a reversed sign for the influence of a feature variable that is captured in the classifier. Such a sign reversal can result in a user being confronted with counterintuitive behaviour of the constructed classifier upon entering particular combinations of feature values: the classifier may return a decrease in output probability, for example, where the user expects an increase. As argued before by Van der Gaag *et al.* [6], such counterintuitive reasoning behaviour is likely to result in a dip in acceptance of the model in daily practice, even if it shows good performance otherwise. We will use the phrase *lurking dependency* to denote dependencies among the variables in a data set that have the potential of reversing the sign of the influence of a feature variable on the class variable in a naive Bayesian classifier.

The paper is structured as follows. In Section 2, naive Bayesian classifiers are reviewed. In Section 3, we introduce our (fictitious) example and demonstrate the counterintuitive behaviour of the naive Bayesian classifier constructed from the example. In Section 4, we use concepts from the theory of qualitative probabilistic networks to attribute the classifier's behaviour to a lurking dependency. The paper ends in Section 5 with our concluding remarks and suggestions for further research.

2 Bayesian Networks and Classifiers

A Bayesian network is a model of a joint probability distribution $\Pr$ over a set of random variables $\mathbf{V}$ [7]. For ease of exposition, we assume all variables to be binary, that is, each variable $V_i \in \mathbf{V}$ adopts one of the values *true*, which will be denoted as v_i, and *false*, written as $\bar{v}_i$; joint value assignments to a subset of variables $\mathbf{U} \subseteq \mathbf{V}$ will be denoted by bold-faced letters $\mathbf{u}$. To model the probability distribution $\Pr$, the Bayesian network includes a directed acyclic graph in which each node captures a random variable and where the set of arcs captures the probabilistic (in)dependencies between the variables. We say that a chain[1] between two variables is *blocked* by the observed[2] variables if the chain contains either an observed variable with at least one emanating arc, or a variable with two incoming arcs such that neither the variable itself nor any of its descendants in the graph have been observed. The concept of blocking in the network's graph is related to the concept of independency in the represented probability distribution in the following way: if all chains between two variables are blocked, then the two variables are considered mutually independent given the entered observations. The strengths of the relationships between the variables are expressed by means of probability distributions. For each variable V_i, the (conditional) probability distributions $p(V_i \mid \pi(V_i))$ are specified for all possible value assignments to the parents $\pi(V_i)$ of V_i; these probability distributions with each other constitute the (conditional) probability table of the variable V_i. The network now represents the unique joint probability distribution $\Pr$ over the variables $\mathbf{V}$, with

$$\Pr(\mathbf{V}) = \prod_{V_i \in \mathbf{V}} \Pr(V_i \mid \pi(V_i))$$

In essence, a Bayesian network allows the computation of any probability of interest over its variables $\mathbf{V}$. Efficient algorithms are available, more specifically, for computing distributions over single variables [7].

Naive Bayesian classifiers are Bayesian networks of restricted topological structure that are tailored to solving problems in which instances described by a number of features have to be classified in one of several distinct classes [1]. For this purpose, a naive Bayesian classifier partitions its set of random variables into a designated class variable C and a set $\mathbf{F}$ of feature variables F_i, $i = 1, \ldots, n$, $n \geq 1$. A joint value assignment to all feature variables will be termed a case; a joint value assignment to all variables involved is called an instance. The probabilistic relationships between the classifier's variables are modelled in a directed tree in which the class variable C is the unique root and each feature variable F_i has C for its only parent. This graphical structure captures dependency of the class variable on each feature variable separately and models mutual independency of any two feature variables given this class variable. To supplement the graphical structure, the classifier specifies a prior probability distribution $\Pr(C)$ over its class variable and the conditional probability distributions $\Pr(F_i \mid C)$ over each feature variable separately. Through its graphical structure and associated probability distributions, a naive Bayesian classifier represents the joint probability distribution $\Pr(\mathbf{F}, C)$ over its variables, factorised according to:

$$\Pr(\mathbf{F}, C) = \Pr(C) \cdot \prod_{i=1}^{n} \Pr(F_i \mid C)$$

Although any probability over its variables can be computed from a naive Bayesian classifier, it is commonly used for establishing the posterior probability distribution $\Pr(C \mid \mathbf{f})$ over the class variable for a newly

[1] A chain in a directed graph is a path which disregards arc directions, that is, it is a path in the underlying undirected graph.

[2] A variable is considered observed when a value for that variable has been entered; the variable then assumes a 0/1 distribution.

presented case $\mathbf{f}$. Associated with the classifier is a decision rule which serves to assign the presented case to a specific class based upon the computed posterior distribution [1, 2]. Since the exact rule used is not relevant for the present paper, we refrain here from further discussion.

Learning a naive Bayesian classifier from a data set of instances amounts to first modelling the random variables involved in a graphical structure; note that since this structure is of fixed topology, it is constructed from a specification of the variables only without any reference to their values in the data set. To supplement the graphical structure, all required probabilities are extracted from the available data as proportions over (sub-)sets of instances; these proportions constitute maximum-likelihood estimates for the separate probabilities. Because of their simplicity, naive Bayesian classifiers are being developed for a wide range of application domains and, despite their simplicity, often very good performance is reported [1, 2].

3 A Striking Example

As argued in the introduction, several researchers have investigated the effects of redundancy of information among the feature variables on the performance of a naive Bayesian classifier. The effects of other types of dependency among the feature variables have received far less attention. In this section, we demonstrate the undesirable effects that the presence of a particular type of dependency among the feature variables can have on a classifier's performance. For this purpose, we present a small fictitious example; in Section 4, we attribute the observed effects to the presence of a lurking dependency. We would like to note that although the example itself is small, the demonstrated effect is not unlikely to arise in real-life problem domains.

Our example pertains to the weather in the Netherlands and how it is perceived by the Dutch. The Netherlands have a moderate maritime climate with cool summers and mild winters. The weather often is quite windy, rainy and cold, with heavily clouded skies. The Dutch like to complain about their weather, especially about the often high humidity, be it so in reality or in perception. Our fictitious example network now includes some weather aspects about which the Dutch tend to complain. The network is depicted in Figure 1 and serves for predicting the Dutch' sense of humidity based on temperature and cloudiness. The network includes the three variables C, F_1 and F_2. The variable C captures whether or not there will be a sense of high humidity; the value $true$ denotes the weather being perceived as "sticky". The variable F_1 models the temperature; the value $true$ denotes a temperature of 22°C or more, which is quite high by Dutch standards. The variable F_2 captures whether or not the sky is heavily clouded; the value $true$ indicates an overcast sky. The graphical structure of the network portrays the probabilistic (in)dependencies between the three variables; it shows, in fact, that all three variables are dependent of one another. The strengths of the relationships between the variables are expressed by the (fictitious) conditional probability tables shown in the figure. These distributions express, for example, that the Dutch have a warm day with a probability of 0.15. On such days, the sky is not very likely to be heavily clouded: just one in every ten warm days will show an overcast sky. The conditional probability table for the variable C further shows that the Dutch are likely to perceive the weather as humid, irrespective of temperature or cloudiness, although an overcast sky appears to play a bigger role in their sense of humidity than the temperature. We note, moreover, that when it is warm and heavily clouded, the Dutch complain even more about a high humidity than otherwise.

From the example network in essence any probability over the variables F_1, F_2 and C can be computed, but we suppose that we are interested specifically in predicting the sense of humidity, given the temperature and cloudiness. In other words, we are interested in the posterior probability distributions over the variable C given values for the variables F_1 and F_2. From the network, the prior probability of a sense of high humidity is computed to be $\Pr(c) = 0.77$. Given evidence that the temperature is high and the sky is overcast, the posterior probability of the humidity being perceived as high is found to be equal to $\Pr(c \mid f_1, f_2) = 0.82$.

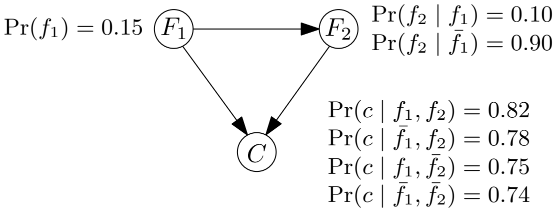
$\Pr(f_1) = 0.15$
F_1
F_2
$\Pr(f_2 \mid f_1) = 0.10$
$\Pr(f_2 \mid \bar{f}_1) = 0.90$
$\Pr(c \mid f_1, f_2) = 0.82$
$\Pr(c \mid \bar{f}_1, f_2) = 0.78$
$\Pr(c \mid f_1, \bar{f}_2) = 0.75$
$\Pr(c \mid \bar{f}_1, \bar{f}_2) = 0.74$
C

Figure 1: The example Bayesian network and its (conditional) probability tables

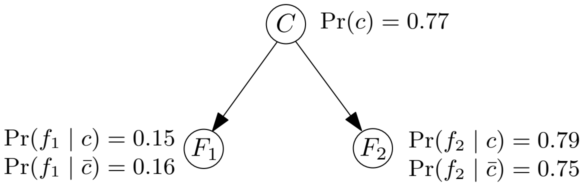

Figure 2: The constructed classifier and its (conditional) probability tables

So, given this particular weather condition, the probability of a sense of high humidity has increased.

We now construct a naive Bayesian classifier for the same prediction problem. For this purpose, we decided to compute the probability tables required for the classification model directly from the probability distributions of the original model, and not from artificially generated data. Note that by doing so we are guaranteed that any observed effect cannot be attributed to chance properties of the necessarily finite data set. We thus computed from the original model the prior probability distribution over the class variable C and the conditional probability distributions for the feature variables F_1 and F_2, respectively, given C. The thus constructed naive Bayesian classifier is shown in Figure 2, along with its (conditional) probability tables. From the classifier, we compute the same probabilities as from the original model. The prior probability of a sense of high humidity again is found to be $\Pr(c) = 0.77$. Given evidence of a warm day and an overcast sky, we now find the posterior probability of a sense of high humidity to be $\Pr(c \mid f_1, f_2) = 0.76$. So, while according to the true distribution we should find an increased probability of high humidity, from the constructed classifier actually a lower posterior probability is found! Note that, dependent upon the decision rule used with the classifier, the weather condition can in fact even be assigned to a different class.

4 An Explanation for the Example Effect

The example from the previous section served to show that constructing a naive Bayesian classifier from a data set of instances may result in a classification model with quite unexpected behaviour. In this section, we study the dependencies between the variables from the example and thereby arrive at an explanation for the counterintuitive behaviour exhibited by the classifier. For this purpose, we will first abstract from the concrete numerical probabilities involved and use qualitative probabilistic concepts for our analysis. We review these qualitative concepts of probability in Section 4.1, and employ them for studying our example in Section 4.2; in Section 4.3, we address the role of the numerical probabilities involved.

4.1 Qualitative Concepts of Probability

Qualitative probabilistic networks were introduced in the 1990s for reasoning about probabilities in a qualitative way [8]. A qualitative probabilistic network again is a model of a joint probability distribution over a set of random variables. Like a numerical Bayesian network, it comprises a directed acyclic graph that encodes the random variables involved as nodes and the probabilistic influences between them as arcs. An arc $A \rightarrow B$ between two variables A and B now expresses that observing a value for A occasions a shift in the probability distribution for B. The direction of this shift can be positive, negative or ambiguous, and is indicated by a qualitative sign. More formally, a qualitative probabilistic influence between two variables expresses how observing a value for one variable affects the probability distribution for the other variable. A positive qualitative influence of a variable A on a variable B along an arc $A \rightarrow B$, for example, expresses that observing the value *true* for A makes the value *true* for B more likely, regardless of any other direct influences on B, that is, $\Pr(b \mid a, \mathbf{x}) - \Pr(b \mid \bar{a}, \mathbf{x}) \geq 0$ for any combination of values $\mathbf{x}$ for the set $\mathbf{X} = \pi(B) \setminus \{A\}$ of parents of B other than A; the influence is denoted $S^+(A, B)$, where the '+' is termed the sign of the influence. A negative qualitative influence, denoted by S^-, and a zero qualitative influence, denoted by S^0, are defined analogously, replacing $\geq$ in the above formula by $\leq$ and $=$, respectively. For a positive, negative or zero qualitative influence of A on B, the difference $\Pr(b \mid a, \mathbf{x}) - \Pr(b \mid \bar{a}, \mathbf{x})$ has the same sign for *all* combinations of values $\mathbf{x}$ for the set $\mathbf{X}$. These influences thus describe a monotonic effect of a shift in A's probability distribution on the distribution for B. If the influence of A on B is positive given one particular combination of values and negative given another combination, however, the influence

$\otimes$	$+$	$-$	0	$?$
$+$	$+$	$-$	0	$?$
$-$	$-$	$+$	0	$?$
0	0	0	0	0
$?$	$?$	$?$	0	$?$

$\oplus$	$+$	$-$	0	$?$
$+$	$+$	$?$	$+$	$?$
$-$	$?$	$-$	$-$	$?$
0	$+$	$-$	0	$?$
$?$	$?$	$?$	$?$	$?$

Table 1: The $\otimes$- and $\oplus$-operators for combining signs

is non-monotonic. Non-monotonic influences are associated with the sign '?', indicating that their effect is unknown. In addition to influences, a qualitative probabilistic network also includes additive and product synergies to capture joint effects among the variables involved; since we will not exploit these concepts for our analysis in the sequel, we refrain from discussing them here.

The set of all influences of a qualitative probabilistic network exhibits various important properties that can be used for establishing the sign of a net influence between any two variables [8]. The property of symmetry, for example, states that if a network includes the influence $S^{\delta}(A, B)$, then it also includes the influence $S^{\delta}(B, A)$, $\delta \in \{+, -, 0, ?\}$. The transitivity property asserts that the qualitative influences along a chain that specifies at most one incoming arc for each variable, combine into a net influence whose sign is defined by the $\otimes$-operator from Table 1. The property of composition asserts that multiple influences between two variables along parallel chains combine into a net influence whose sign is defined by the $\oplus$-operator. From the definition of the $\oplus$-operator in Table 1, we observe that the composition of two influences with opposite signs along parallel chains will give rise to an unknown result, captured by the sign '?'.

Probabilistic inference with a qualitative network is based on the idea of combining and propagating signs over the network's graphical structure [9, 10]. The algorithm traces the effects of observing a node's value on the other nodes in the network by message-passing between neighbours. Informally speaking, for each observed node, the appropriate sign is entered. Each node receiving a message updates its sign with the $\oplus$-operator, and then sends a message to each neighbour from which it is not blocked given the entered observations. The sign of this message is the $\otimes$-product of the node's (new) sign and the sign of the influence it traverses. This process is repeated throughout the network, building on the properties of symmetry, transitivity, and composition of influences, until no further sign changes are induced.

4.2 The Example Revisited

We consider again our example Bayesian network and the naive Bayesian classifier that was constructed from the represented joint probability distribution over the three variables involved. To find an explanation for the counterintuitive behaviour of the classifier, we derive the signs of the influences between the variables and use the sign-propagation algorithm to establish the net effect of the observations on the class variable.

From the probability distributions specified for the class variable C in the original Bayesian network, we find that the direct influence of the variable F_1 on C along the arc $F_1 \rightarrow C$ is positive:

$$\begin{aligned}
&\Pr(c \mid f_1, f_2) - \Pr(c \mid \bar{f}_1, f_2) = 0.82 - 0.78 = 0.04 \geq 0, \quad \text{and} \\
&\Pr(c \mid f_1, \bar{f}_2) - \Pr(c \mid \bar{f}_1, \bar{f}_2) = 0.75 - 0.74 = 0.01 \geq 0
\end{aligned}$$

The direct qualitative influence of F_1 on C thus captures the information that a higher temperature increases the probability of a sense of high humidity. Similarly, the variable F_2 is found to have a positive direct influence on C: a heavily clouded sky thus is associated with an increased probability of the humidity being perceived as high. The original Bayesian network in addition includes a direct qualitative influence of F_1 on F_2, which is found to be negative:

$$\Pr(f_2 \mid f_1) - \Pr(f_2 \mid \bar{f}_1) = 0.10 - 0.90 = -0.80 \leq 0$$

indicating that a heavily clouded sky is less likely with a higher temperature. Figure 3 again depicts, on the left, the original network from our example, now supplemented with the signs of its qualitative influences.

From the constructed qualitative probabilistic network, we would like to establish the sign of the joint influence of a high temperature and a clouded sky on the probability of a sense of high humidity. For this purpose, the value $true$ is entered into the network as evidence for the two variables F_1 and F_2. The sign-propagation algorithm thereupon finds that the variable F_1 exerts a direct positive influence on the class variable and that its indirect influence on C over the variable F_2 is blocked by the entered evidence.

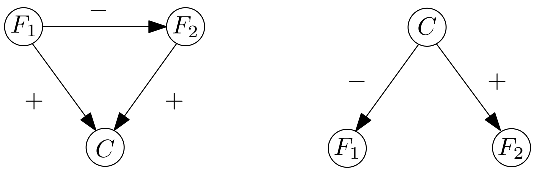

Figure 3: The qualitative abstractions of the original Bayesian network and of the constructed classifier

The algorithm thus establishes the net influence of F_1 on C to be positive. By a similar argument, the net influence of F_2 on C is also computed to be positive. With the $\oplus$-operator, the signs of the two net influences are combined into a '$+$' sign for the compound influence of the two observations:

$$+ \oplus + = +$$

We note that this positive compound influence is conform to the increase in probability for the value $true$ of the variable C which we found from comparing the prior probability $\Pr(c)$ with the posterior probability $\Pr(c \mid f_1, f_2)$ computed from the original numerical network.

We now consider the probabilistic relationships between the three variables in the naive Bayesian classifier computed from the original (numerical) Bayesian network and derive the signs of its qualitative influences. From the probability distributions computed for the variable F_2, we find a positive direct influence of C on F_2:

$$\Pr(f_2 \mid c) - \Pr(f_2 \mid \bar{c}) = 0.79 - 0.75 = 0.04 \geq 0$$

From the probability distributions established for the variable F_1, we find that the direct influence of the class variable C on F_1 is negative:

$$\Pr(f_1 \mid c) - \Pr(f_1 \mid \bar{c}) = 0.15 - 0.16 = -0.01 \leq 0$$

Figure 3 depicts, on the right, the originally constructed naive Bayesian classifier, now supplemented with the signs of its qualitative influences.

From the qualitative classifier, we would again like to establish the joint influence of a high temperature and a clouded sky on the probability of a sense of high humidity. For this purpose, again the value $true$ is entered into the classifier as evidence for the two feature variables F_1 and F_2. With the sign-propagation algorithm, we find that the variable F_2 exerts a positive direct influence on C and that F_1 now exerts a negative direct influence on the class variable. The sign of the compound influence of the two observations is once again established using the $\oplus$-operator: we find that

$$+ \oplus - = ?$$

which expresses that the sign of the compound influence is unknown. This result indicates that the sign of the joint influence of the two observations on the class variable cannot be established by qualitative arguments only and in practice is dependent on the actual parameter probabilities in the original numerical network. Note that our example in Section 3 showed that these parameter probabilities cause the compound influence in the constructed classifier to be negative.

The above analysis reveals that, while the direct influence of the variable F_1 on the variable C is positive in the original numerical network, it is established to be negative upon constructing the classifier. This seeming contradiction originates from the observation that the *direct* influence of F_1 on C to be modelled in the classifier is actually computed from the original network as the *compound* influence of F_1 on C. The influence of F_1 on C which is modelled in the classifier thus combines the positive direct influence of F_1 on C from the original network with the negative indirect influence through F_2. Note that the sign of this indirect influence is established by using the $\otimes$-operator to combine the signs of the two separate influences along the arcs $F_1 \rightarrow F_2$ and $F_2 \rightarrow C$. The direct and indirect influences of F_1 on C then combine into a compound influence whose sign depends upon the actual parameter probabilities in the original numerical network as argued above. Since the negative indirect influence of F_1 on C is stronger than the positive direct influence, the overall result is a negative influence of F_1 on C in the classifier.

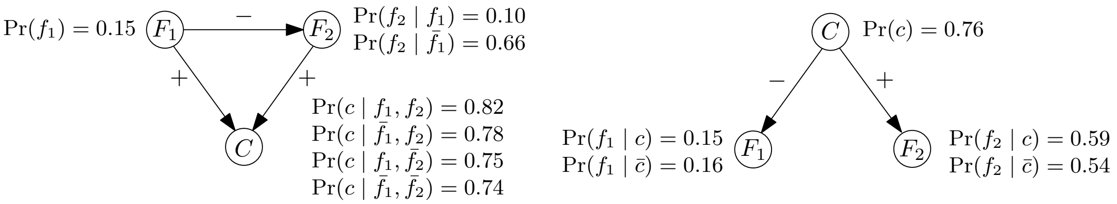
$\Pr(f_1) = 0.15$
F_1
$-$
F_2
$\Pr(f_2 \mid f_1) = 0.10$
$\Pr(f_2 \mid \bar{f}_1) = 0.66$
$+$
$+$
C
$\Pr(c \mid f_1, f_2) = 0.82$
$\Pr(c \mid \bar{f}_1, f_2) = 0.78$
$\Pr(c \mid f_1, \bar{f}_2) = 0.75$
$\Pr(c \mid \bar{f}_1, \bar{f}_2) = 0.74$
C
$\Pr(c) = 0.76$
$-$
$+$
$\Pr(f_1 \mid c) = 0.15$
$\Pr(f_1 \mid \bar{c}) = 0.16$
F_1
F_2
$\Pr(f_2 \mid c) = 0.59$
$\Pr(f_2 \mid \bar{c}) = 0.54$

Figure 4: The modified numerical Bayesian network and the constructed naive Bayesian classifier, along with their (conditional) probability distributions and associated signs

4.3 Lurking Dependencies and Counterintuitive Results

In the previous section we demonstrated that the dependency between the two feature variables F_1 and F_2 in the original example joint probability distribution resulted in a sign reversal for the influence of F_1 on the class variable C for the constructed classifier; we use the phrase *lurking dependency* for any dependency that has the potential of inducing such a sign reversal. Whether a sign reversal will actually result in counterintuitive behaviour of a constructed classifier is dependent, however, on the strengths of the influences concerned and, hence, of the parameter probabilities involved.

Informally speaking, only reversal of the sign of a relatively strong influence is likely to enforce counterintuitive results. In our example Bayesian network, the lurking dependency between the feature variables F_1 and F_2 was strong enough to lead to an unexpected result. To demonstrate that a weaker dependency need not result in the same counterintuitive behaviour, we slightly modified our original network; the modified numerical network and accompanying classifier are shown in Figure 4. Note that we only changed the probability $\Pr(f_2 \mid f_1)$, from 0.90 to 0.66. The strength of the influence of F_1 on F_2, expressed by the difference $|\Pr(f_2 \mid f_1) - \Pr(f_2 \mid \bar{f}_1)\,|$, has thereby weakened, from 0.80 to 0.56; as a consequence, the strength of the influence of the class variable C on F_2 established for the classifier, has slightly gained in strength, from 0.04 to 0.05. Further note that although we changed a parameter probability and thereby changed the strengths of some influences, the signs of all influences in the resulting classifier remained the same. From the modified Bayesian network and associated classifier, the prior probability of the value $true$ of the class variable C is found to be $\Pr(c) = 0.76$. The posterior probability $\Pr(c \mid f_1, f_2)$ computed from the modified numerical network equals 0.82, while this same probability is computed from the associated classifier to be $\Pr(c \mid f_1, f_2) = 0.77$. Although the two posterior probabilities are different, they both are found to be larger than the prior probability $\Pr(c)$. The naive Bayesian classifier therefore does not return a probability distribution that is counterintuitive to the user.

5 Conclusions and Further Research

Naive Bayesian classifiers are simple stochastic classification models that are known to perform quite well, even compared to more complex models. Naive Bayesian classifiers build, however, on the rather strong assumption of mutual independence of their feature variables given the class variable. Over the years, researchers have investigated the effects of several types of dependency between the feature variables on the performance of a naive Bayesian classifier. In this paper, we contributed to this line of research by studying the undesirable effects of lurking dependencies among the feature variables, where dependencies are called lurking whenever they have the potential of reversing the sign of the influence of a particular feature variable on the class variable in a classifier. In our study, we decided not to learn our classifiers from data, so as to guarantee that no other effects could arise than those inflicted by the lurking dependencies. We would like to note, however, that upon learning a classifier from a finite data set, the effects of lurking dependencies may be magnified by the differences between the underlying true and estimated probability distributions.

Our analyses suggest that lurking dependencies among feature variables may result in unexpected model behaviour when the topological structure of the true joint probability distribution and that of the classifier are different. When the topological structure of a classifier is equivalent to that of the true distribution, the problems studied in this paper need not arise. However, upon constructing a TAN classifier, [1], from the joint probability distribution of our example network, for instance, the two models will result in the same behaviour. In most practical applications, the constructed Bayesian network classifier will have a simpler topology than the true distribution. The undesirable effects from lurking dependencies studied in this paper are then likely to arise. Further investigation of these observations is required, however, before any definite conclusions can be drawn.

To conclude, our examples served to demonstrate that a sign reversal inflicted by a lurking dependency may cause results to arise that are counterintuitive to the user. Unfortunately, the acceptance of classification models relies not just on their overall performance, but also on the nature of any erroneous results. To reduce model rejection due to unexpected behaviour, it is important to thoroughly understand the effects of lurking dependencies and to be able to eliminate these effects, to at least some extent, upon constructing Bayesian network classifiers. We plan to continue our research in this direction and hope to present useful results and techniques in the near future.

Acknowledgement. This research was funded by the Netherlands Organisation for Scientific Research (NWO).

References

[1] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[2] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[3] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[4] P. Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall symposium on relevance*, pages 140–144. AAAI Press, 1994.

[5] P.L. Geenen, L.C. van der Gaag, W.L.A. Loeffen, and A.R.W. Elbers. On the robustness of feature selection with absent and non-observed features. In J.M. Barreiro, F. Martin-Sanchez, V. Maojo, and F. Sanz, editors, *Proceedings of the Fifth International Symposium on Biological and Medical Data Analysis*, pages 148–159, Springer-Verlag, Heidelberg, 2004.

[6] L.C. van der Gaag, H.J.M. Tabachneck-Schijf, and P.L. Geenen. Verifying monotonicity of Bayesian networks with domain experts. *International Journal of Approximate Reasoning*, 50:429–436, 2009.

[7] F.V. Jensen and Th.D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer, New York, 2007.

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

[9] M.J. Druzdzel and M. Henrion. Efficient reasoning in qualitative probabilistic networks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 548–553, AAAI Press, Menlo Park, 1993.

[10] S. Renooij, L.C. van der Gaag, and S. Parsons. Propagation of multiple observations in QPNs revisited. In F. van Harmelen, editor, *Proceedings of the Fifteenth European Conference on Artificial Intelligence*, pages 665–669, IOS Press, Amsterdam, 2002.