The Representational Power of Discrete Bayesian Networks

Charles X. Ling LING@CSD.UWO.CA
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada, N6A 5B7

Huajie Zhang HZHANG@UNB.CA
Faculty of Computer Science
University of New Brunswick
P.O. Box 4400, Fredericton, NB, Canada, E3B 5A3

Editor: Carla Brodley and Andrea Danyluk

Abstract

One of the most important fundamental properties of Bayesian networks is the representational power, reflecting what kind of functions they can or cannot represent. In this paper, we establish an association between the structural complexity of Bayesian networks and their representational power. We use the maximum number of nodes' parents as the measure for the Bayesian network structural complexity, and the maximum XOR contained in a target function as the measure for the function complexity. A representational upper bound is established and proved. Roughly speaking, discrete Bayesian networks with each node having at most k parents cannot represent any function containing $(k+1)$-XORs. Our theoretical results help us to gain a deeper understanding on the capacities and limitations of Bayesian networks.

Keywords: Bayesian Networks, Representational Power, Learning Algorithms

1. Introduction

Bayesian networks (BNs) are probabilistic models that combine probability theory and graph theory (Pearl, 1988). They represent causal and probabilistic relations among random variables that are governed by probability theory. Probabilistic inferences and optimal decisions can be made directly from Bayesian networks. Bayesian networks have been widely used in many applications, because they provide intuitive and causal representations of real-world applications, and they are supported by a rigorous theoretical foundation.

A Bayesian network consists of two parts: a directed acyclic graph and a set of conditional probabilities. The directed acyclic graph represents qualitative dependencies among random variables, and the conditional probabilities quantify these dependencies. The following is a definition of Bayesian networks.

Definition 1 *A Bayesian network, or simply BN, is a directed acyclic graph $G =< N, E >$ and a set P of probability distributions, where $N = \{A_1, \cdots, A_n\}$ is the set of nodes and E is the set of arcs connecting pairs of nodes. P is the set of local conditional distributions, one for each node conditioned on the parents of the node. The local conditional distribution of a node A_i is denoted by $P(A_i|pa_i)$, where pa_i denotes the parents of A_i.*

There are two types of random variables for which nodes represent: discrete variables that take values from a finite set, and numeric or continuous variables that take values from a set of continuous numbers. BNs can thus be classified into three corresponding categories: discrete BNs, continuous BNs, and mixed BNs. In this paper, we restrict our discussion to discrete BNs.

A BN G on $A_1, \cdots, A_n$ defines a joint probability distribution P_G as below:

$$P_G(A_1, \cdots, A_n) = \prod_{i=1}^{n} P(A_i|pa_i) \tag{1}$$

Obviously, the complexity of different BN structures can be different. The simplest case is a set of nodes without arcs, and the most complex one is the maximum graph without circle. It is common to use the maximum number of nodes' parents as a measure for its structural complexity. Thus, we have the following definition.

Definition 2 *Given a BN G, the maximum number of parent of a node on G is called the structural order, denoted by $O_s(G)$.*

It is well known that any node in a BN is conditionally independent of its nondescendants, given its parents (Pearl, 1988). Actually, a node is only affected by the nodes in its Markov blanket (Pearl, 1988), defined below.

Definition 3 *The Markov blanket of a node A in BN G is a set of nodes that make up of A's parents and children, and the parents of A's children.*

For example, in Figure 1 the Markov blanket of A_5 is A_2, A_3, A_4 and A_7.

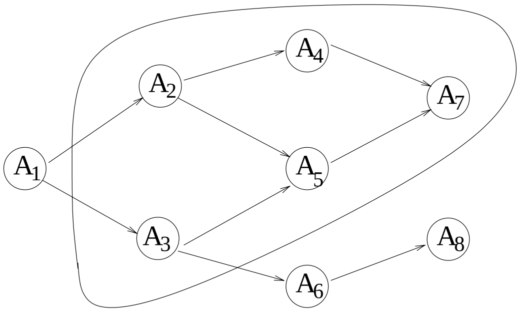


Figure 1: An example of Markov blanket

BNs are often used for classification (Friedman et al., 1997), and a classifier is to be constructed from a given set of training examples with the class label. A classifier is a function that maps from examples to class labels. Assume that A_1, $A_2, \cdots$, A_n are n attributes. An example E is represented by a vector $(a_1, a_2, , \cdots, a_n)$, where a_i is the value of A_i. Again, in this paper, we restrict our discussion to discrete attributes, and in addition, the class label must be binary. We use C to represent the classification variable taking value $+$ (positive class) or $-$ (negative class), and use c to represent the value that C takes.

An especially simple BN structure, often used for classification, is called naive Bayesian classifier, or simply Naive Bayes. In Naive Bayes, the conditional independence assumption is made; that is, all attributes are independent given the value of the class variable. Give an example $E = (a_1, \cdots, a_n)$, the equation below represents formally such conditional independence assumption:

$$p(a_1, a_2, \cdots, a_n | c) = \prod_{i=1}^{n} p(a_i | c).$$

According to Bayes Theorem and the conditional independence assumption, the classification function of Naive Bayes can be represented as:

$$G(E) = \frac{p(C=+)}{p(C=-)} \prod_{i=1}^{n} \frac{p(a_i|C=+)}{p(a_i|C=-)}.$$

Figure 2 (a) is an example of Naive Bayes represented as a BN.

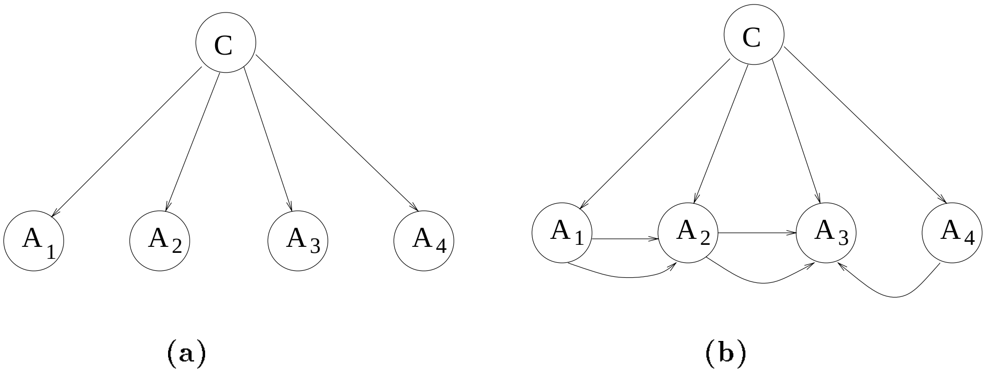


Figure 2: (a) an example of Naive Bayes (b) an example of ANB

Since the conditional independence assumption hardly holds true, the structure of Naive Bayes is often extended by adding arcs, reflecting dependencies among attributes. The resulting BNs are called Augmented Naive Bayes, or simply ANB. In an ANB, the classification node directly points to all attributes, and links among attributes are allowed (except that they do not form any directed cycle). Figure 2 (b) shows an example of ANB represented as a BN.

ANB is a special structure of general BNs, in which the class node is identified and all attributes are within the Markov blanket of the class node. In a general BN, no node is specified as the class node, and each node can be the class node. When we choose a node A_i as the class node, nodes not in A_i's Markov blanket do not affect the classification and can be deleted, assuming the value of the nodes in the Markov blanket is known. Thus, we can view a BN as a set of classifiers with different class nodes.

One of the most fundamental issues of BNs is the representational power. Here the representational power of a set of BNs is defined as the set of target functions whose results can be reproduced by BNs from the set. Essentially, the representational power of BNs reflects their fundamental capacities and limitations. A natural question about BNs is: what are the differences in representational power with different structural complexities? Intuitively, the more complex the structure of a BN, the more complex the target function it can represent. However, to our knowledge, little is known about the representational power of BNs. In our previous work (Zhang and Ling, 2001b), we investigated the representational power of Naive Bayes and ANB. We will review related results on Naive Bayes and ANBs in the next two sections.

2. Related Work

In the binary domain, where all attributes are Boolean, it is easy to show the representational power of both Naive Bayes and ANB. Let us briefly review the relevant results (Duda and Hart, 1973).

Suppose that attributes A_1, A_2, $\cdots$, A_n are binary, taking value 0 or 1. Let p_i and q_i represent the probability $p(A_i = 1|C = +)$ and $p(A_i = 1|C = -)$ respectively, and $E = (a_1, \cdots, a_n)$ be an example. Then the corresponding Naive Bayes $G(E)$ is:

$$G(E) = \frac{p(C=+)}{p(C=-)} \prod_{i=1}^{n} \frac{{p_i}^{a_i}(1-p_i)^{1-a_i}}{{q_i}^{a_i}(1-q_i)^{1-a_i}}. \qquad (2)$$

It is straightforward to obtain a linear classifier by applying logarithm to the above equation. Thus, Naive Bayes is a linear classifier in the binary domain.

For the discrete domain, a general case of the binary domain (since discrete attributes may have more than two values), there was no satisfying result. Assume that A_1, A_2, $\cdots$, A_n are n discrete attributes, each attribute A_i may have m values a_{i1}, a_{i2}, $\cdots$, and a_{im} ($m \geq 2$). Domingos and Pazzani (1997) and Peot (1996) introduced m new Boolean attributes B_{i1}, B_{i2}, $\cdots$, and B_{im} for each attribute A_i, and proved that Naive Bayes is linear over these new binary attributes. However, the linear separability is on $m_1 \times m_2 \cdots \times m_n$ new attributes, not the original attributes.

In fact, Naive Bayes *can* represent nonlinear functions (Zhang and Ling, 2001a). For example, let $A = \{1, 2, 3\}$, $B = \{1, 2, 3\}$, a function f is defined as in Figure 3. Obviously, it is not linearly separable. However, there is a Naive Bayes that represents f. Consider a Naive Bayes G on two specific nominal attributes A and B, where $A = \{1, 2, 3\}$, $B = \{1, 2, 3\}$. Table 1 is the conditional probability table (CPT) for A, and B has the same CPT as A. It is easy to verify that the classification of G is the same as in Figure 3. Thus, f is representable by G. Therefore, Naive Bayes can represent some, but not all (as we will see later), nonlinear functions in the discrete domain. The precise representational power of Naive Bayes in the discrete domain is still unknown.

The representational power of arbitrary ANB is also known in the binary domain. Assume that each node A_i can have up to k parents, and let $pa_i = \{A_{i1}, \cdots, A_{ik}\}$ denote the

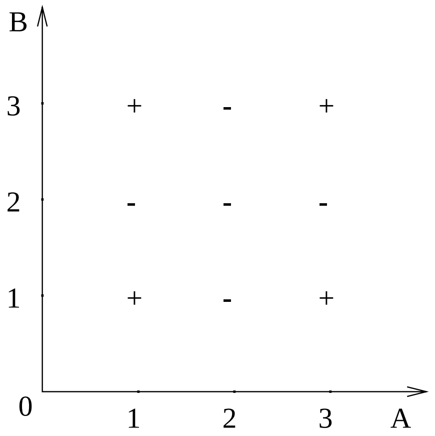


Figure 3: A nonlinear function f

Table 1: The conditional probability table for A.

	$A=1$	$A=2$	$A=3$
$C=-$	0.3	0.4	0.3
$C=+$	0.5	0	0.5

parents of A_i. Then

$$p(A_i|c, pa_i) = \begin{cases} \theta_1 & A_i = 0, A_{i1} = 0, A_{i2} = 0, \cdots, A_{ik} = 0 \\ 1-\theta_1 & A_i = 1, A_{i1} = 0, A_{i2} = 0, \cdots, A_{ik} = 0 \\ \theta_2 & A_i = 0, A_{i1} = 1, A_{i2} = 0, \cdots, A_{ik} = 0 \\ 1-\theta_2 & A_i = 1, A_{i1} = 1, A_{i2} = 0, \cdots, A_{ik} = 0 \\ \cdots & \\ \theta_m & A_i = 0, A_{i1} = 1, A_{i2} = 1, \cdots, A_{ik} = 1 \\ 1-\theta_m & A_i = 1, A_{i1} = 1, A_{i2} = 1, \cdots, A_{ik} = 1 \end{cases}$$

$$= \theta_1^{(1-A_i)(1-A_{i1})\cdots(1-A_{ik})}(1-\theta_1)^{A_i(1-A_{i1})\cdots(1-A_{ik})}\cdots(1-\theta_m)^{A_iA_{i1}\cdots A_{ik}}$$

and

$$G(E) = \frac{p(C=+)}{p(C=-)}\prod_{i=1}^{n}\frac{p(A_i|+, pa_i)}{p(A_i|-, pa_i)} \tag{3}$$

When we apply logarithm to it and convert the product into a sum, we get a set of terms of at most $k+1$ degree as follows:

$$\bar{A}_i\bar{A}_{i1}\cdots\bar{A}_{ik}$$

where $\bar{A}_i$ is either A_i or $1-A_i$. The same is true for $\bar{A}_{ij}$. Thus the representational power of an ANB in which each node has at most k parents equals to a polynomial of degree

$k+1$. Thus, in binary domains, Naive Bayes represents linear functions, TAN[1] represents quadratic functions, and so on.

However, the representational power of ANBs in the discrete domain is much more complex than that in the binary domain, and the derivation above cannot be extended to the discrete domain. Indeed, there is no one-to-one relation between the maximum parent number of ANB and the degree of polynomials. We have shown that Naive Bayes does not correspond to linear functions any more in the discrete domain. To our knowledge, there was little work by other researchers on the linearity of Naive Bayes, and the representational power of ANB with different structures, in the discrete domain.

In our previous work (Zhang and Ling, 2001b), we investigated the representational power of Naive Bayes and ANB. We extend and generalize our previous work in this paper. In the next section, we will briefly review our previous work and introduce a few concepts that we will use in this paper.

3. The Representational Power of Naive Bayes and ANB

Definition 4 *Given n discrete attributes A_1, A_2, $\cdots$, A_n, and a class variable C, a function f from $A_1 \times A_2 \cdots \times A_n$ to C is called an n-dimensional discrete function.*

To discuss the representational power of ANB, we need a measure for the complexity of target functions. The VC-dimension is widely used for hypothesis space complexity (Vapnik and Chevonenkis, 1971), but it might not provide enough granularity. There was some work on complexity measure for Boolean functions by the computational complexity (Paterson, 1992), which uses the size of the minimum combinational network that computes the function.[2] However, this does not seem to be direct enough to measure the complexity with respect to the difficulty in Bayesian learning.

In our previous work (Zhang and Ling, 2001b), we proposed a measure which uses the maximum XOR contained in the function as the complexity of a function in the discrete domain. We adopt this measure in this paper.

The reason to use the maximum XOR (also known as the parity function) as the complexity measure for a function comes from the intuition that the n variables making up an n-XOR depend on each other; i.e., the value of an n-XOR function cannot be determined until the values of all variables are known. Bayesian networks represent target functions by exploiting conditional dependencies among variables. Since there is no such conditional dependencies among variables in an XOR, the maximum XOR contained in a function seems to be an appropriate heuristic for the complexity of a function. As we will see, this measure is indeed appropriate to Bayesian network representation. Let us first briefly review the related concepts.

Definition 5 *An n-XOR function with n Boolean variables is defined as to return 1 if and only if an even number of variables are 1. n is called the order of the XOR.*

[1] TAN stands for Tree Augmented Naive Bayes, a special case of ANB in which each attribute can have at most one parent other than the class node, thus forming a tree structure among attributes.

[2] A combinational network consists of NOT, AND and OR gates, and its size is the number of such gates.

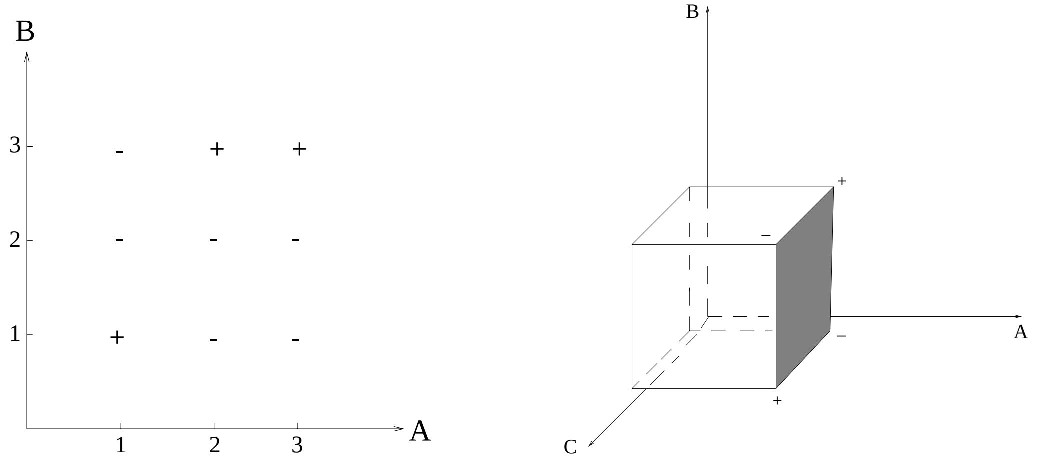
By this notation, 2-XOR is a regular (2-variable) XOR (parity function). We propose to use the highest order of XOR "contained" in a discrete function as its complexity measure (Zhang and Ling, 2001b); that is, the maximum subfunction that forms an XOR pattern.

Definition 6 *Assume that f is an n-dimensional discrete function on A_1, A_2, $\cdots$, A_n, C. An $(n-1)$-dimensional partial function f_p on A_1, $\cdots$, A_{i-1}, A_{i+1}, $\cdots$, A_n and C, and $A_i = a_{ij}$, is called an $(n-1)$-dimensional subfunction of f at $A_i = a_{ij}$, denoted by $f(a_{ij})$, where $1 \leq i \leq n$.*

Similarly, we can get an arbitrary k-dimensional subfunction of f, by fixing $(n-k)$ attributes, where $2 \leq k \leq n-1$. An important feature of n-XOR is that its any k-dimensional subfunction is also a k-XOR.

Definition 7 *An n-dimensional discrete function f is said to contain a k-XOR, if there is a k-dimensional subfunction f_p on attributes A_{k1}, A_{k2}, $\cdots$, A_{kk}, and for each attribute A_{ki}, there are two different values, a_{ki_1}, a_{ki_2}, denoted by a_{ki} and $\bar{a}_{ki}$, such that a partial function $f_{p'}$ of f_p from $\{a_{k1}, \bar{a}_{k1}\} \times \cdots \times \{a_{kk}, \bar{a}_{kk}\}$ to $\{+, -\}$ is a k-XOR function.*

Figure 4 (a) shows a discrete function in two dimensions containing a 2-XOR (on A = 1 and 3, and B = 1 and 3), and (b) shows a binary function in three dimensions containing a 2-XOR (on B-C with A = 1).



(a) (b)

Figure 4: (a) a function containing 2-XOR in two dimensions (b) a function containing 2-XOR in three dimensions

Definition 8 *An n-dimensional discrete function f is said to have an order of m, if the maximum XOR it contains is an m-XOR, denoted by $O_f(f)$.*

This complexity measure for a discrete function is different from the one that uses the size of the minimum combinational network (Paterson, 1992) in two ways. First, it is applicable to any discrete functions, rather than just the binary functions. Second, it is simpler, since we only consider the part of a function that consists of the highest order of XOR, instead of the whole function.

In the discrete domain, it has been shown that Naive Bayes can produce nonlinear boundaries, but it still cannot represent any nonlinear function containing a 2-XOR (Zhang and Ling, 2001a). In our previous work, we proved upper bounds on the representational power of Naive Bayes and TAN, and presented a conjecture on the upper bound of general ANB (Zhang and Ling, 2001b).

As we discussed earlier, however, ANB is a special form of BNs. What is the representational power of general BNs? More precisely, what is the relation between the structural complexity of a BN and its representational power? This paper will answer this question by extending the previous results on ANB to general BNs.

4. The Representational Power of General BNs

4.1 Augmented Naive Bayes vs General Bayesian Networks

Naive Bayes and ANB represent a classification function with the root as the class variable. What does a general BN represent? One way to view a BN from a classification viewpoint is that each node could be a class variable. Thus, a BN represents a set of classification functions. Formally, we have the following definition.

Definition 9 *Given a BN G on $A_1, \cdots, A_n$ and an example $E = (a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n)$, the classification function f_i corresponding to node A_i is defined as:*

$$f_i(a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n) = \max_{a_i} P_G(a_i | a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n).$$

The set of classification functions represented by G is $\{f_1, \cdots, f_n\}$, denoted by $R(G)$.

Definition 10 *A BN G's representational order $O_r(G)$ is defined as:*

$$O_r(G) = \max\{O_f(f_i), f_i \in R(G)\}$$

4.2 The Representational Power of 1-order BNs

For a BN of order 1, each node has at most one parent. In fact, the structure of such a BN is a forest. Naive Bayes belongs to this class. We have shown that Naive Bayes can represent linear functions and some nonlinear functions, but cannot represent any function containing 2-XOR (Zhang and Ling, 2001b). The following lemma extends the result on Naive Bayes to the BNs with order of 1, thus establishing an upper bound of BNs of order 1.

Lemma 1 *For any BN G, if $O_s(G) = 1$, then $O_r(G) \leq 1$.*

Proof: suppose that G is a BN on A_1, $\cdots$, A_n and $O_s(G) = 1$. if $O_r(G)$ were 2 or above, then there would be a function f of order 2, i.e., f contains a 2-XOR, that G can represent it.

Suppose that f were from A_1, $\cdots$, A_{k-1}, A_{k+1}, $\cdots$, A_n to A_k. Since f contains a 2-XOR, according to Definition 7, there are two attributes A_i and A_j, i and $j \neq k$, and each of them has two different values, a_i, $\bar{a}_i$, and a_j, $\bar{a}_j$, respectively, and two different values a_k and $\bar{a}_k$ of A_k such that:

$$f(a_1, \cdots, a_i, \cdots, a_j, \cdots, a_n) = a_k \tag{4}$$

$$f(a_1, \cdots, a_i, \cdots, \bar{a}_j, \cdots, a_n) = \bar{a}_k \tag{5}$$

$$f(a_1, \cdots, \bar{a}_i, \cdots, a_j, \cdots, a_n) = \bar{a}_k \tag{6}$$

$$f(a_1, \cdots, \bar{a}_i, \cdots, \bar{a}_j, \cdots, a_n) = a_k \tag{7}$$

where a_l is a value of A_l, $l \neq i$, j and k.

If one of A_i and A_j is out of the Markov blanket of A_k, it does not affect the value of A_k when other attributes are assigned values. Therefore, it is not possible that Equation (4), (5), (6) and (7) hold true simultaneously.

Assume that both A_i and A_j are in the Markov blanket of A_k. Since each node has at most one parent, A_i and A_j are connected directly with A_k. There are two cases for the connection patterns.

(1) A_k points to both A_i and A_j. In this case, it is a structure of Naive Bayes with A_k as the class variable. It is known that Naive Bayes cannot represent 2-XOR (Zhang and Ling, 2001b).

(2) A_i points to A_k and A_k points to A_j. In this case, the joint distribution P_G can be represented as below.

$$P_G(A_i, A_j, A_k) = P(A_k)P(A_i|A_k)P(Aj|A_i, A_k) = P(A_k)P(A_i|A_k)P(Aj|A_k)$$

This means that P_G can be represented by a Naive Bayes with A_k as the class variable. Therefore, if G could represent 2-XOR, the correspondent Naive Bayes could too.

Therefore, G cannot represent any function of order 2.

□

4.3 The Representational Power of General BNs

Let us consider the representational power of general BNs. We will prove that a BN with nodes having at most k parents cannot represent any function of $k+1$ order. First we prove a lemma which is a special case of the main theorem. Its proof helps to illustrate ideas in the proof of the main theorem.

Lemma 2 *For any BN G with $(m+1)$ nodes each of which has at most $(m-1)$ parents, $O_r(G) \leq O_s(G) = m - 1$.*

Proof:

Suppose that G is a BN on discrete attributes A_1, $\cdots$, A_{m+1} and $O_s(G) = m-1$, i.e., each node has at most $m-1$ parents. Let f be a function represented in G, we will prove by contradiction that f cannot contain an m-XOR. Assume that f contains an m-XOR from A_1, $\cdots$, A_m to A_{m+1}. According to Definition 7, there would be two values a_i, $\bar{a}_i$ for each attribute A_i, $1 \le i \le m+1$, such that the partial function f_G of G's classification function from $\{a_1, \bar{a}_1\} \times \cdots \times \{a_m, \bar{a}_m\}$ to $\{a_{m+1}, \bar{a}_{m+1}\}$ is an m-XOR. To simplify our notation, we denote a_{m+1}, $\bar{a}_{m+1}$ by $+$ and $-$ respectively. Then we have 2^m inequalities below:

$$f_G(A_1, A_2, \cdots, A_m) \begin{cases} \geq 1 & \text{if the number of } A_i \text{ taking value } a_i \text{ is even} \\ < 1 & \text{if the number of } A_i \text{ taking value } a_i \text{ is odd} \end{cases} \tag{8}$$

where $i = 1, \cdots, m$ and $f_G(A_1, A_2, \cdots, A_m)$ is specified below.

$$f_G(A_1, A_2, \cdots, A_m) = \frac{p_G(A_1, A_2, \cdots, A_m, +)}{p_G(A_1, A_2, \cdots, A_m, -)}, \tag{9}$$

where $p_G(A_1, A_2, \cdots, A_m, +)$ and $p_G(A_1, A_2, \cdots, A_m, -)$ are the joint distributions of G in the class of $A_{m+1} = a_{m+1}$ and the class of $A_{m+1} = \bar{a}_{m+1}$ respectively. Obviously, all of $A_1, A_2, \cdots, A_m$ should be in A_{m+1}'s Markov blanket.

Since G is a BN, we have:

$$p_G(A_1, A_2, \cdots, A_m, +) = p(+|pa_{m+1}) \prod_{i=1}^{m} p(A_i|pa_i), \tag{10}$$

$$p_G(A_1, A_2, \cdots, A_m, -) = p(-|pa_{m+1}) \prod_{i=1}^{m} p(A_i|pa_i), \tag{11}$$

where pa_i is an assignment of all of A_i's parents, $1 \le i \le m+1$.

Note that if a term in Equation (10) and (11) does not contain $+$ or $-$, then it is cancelled out in f_G and does not affect classification. Thus, all terms in f_G should have a form either $p(A_{m+1}|pa_{m+1})$ or $p(A_i|pa_i)$, where $i \neq m+1$ and $A_{m+1} \in pa_i$.

Let $A_1 = a_1$. We have 2^{m-2} inequalities below:

$$\frac{p_G(a_1, A_2, \cdots, A_m, +)}{p_G(a_1, A_2, \cdots, A_m, -)} \geq 1, \tag{12}$$

where the number of A_i taking value a_i is odd ($i \neq 1$).

Multiply all these 2^{m-2} inequalities together, we have:

$$\prod^{odd} \frac{p_G(a_1, A_2, \cdots, A_m, +)}{p_G(a_1, A_2, \cdots, A_m, -)} \geq 1. \tag{13}$$

Similar to (12), we have 2^{m-2} inequalities below:

$$\frac{p_G(a_1, A_2, \cdots, A_m, +)}{p_G(a_1, A_2, \cdots, A_m, -)} < 1, \tag{14}$$

where the number of A_i taking value a_i is even ($i \neq 1$).

Multiply all these 2^{m-2} inequalities together, we have:

$$\prod^{even} \frac{p_G(a_1, A_2, \cdots, A_m, +)}{p_G(a_1, A_2, \cdots, A_m, -)} < 1. \tag{15}$$

Divide (13) by (15), we have:

$$\prod^{odd} \frac{p_G(a_1, A_2, \cdots, A_m, +)}{p_G(a_1, A_2, \cdots, A_m, -)} \prod^{even} \frac{p_G(a_1, A_2, \cdots, A_m, -)}{p_G(a_1, A_2, \cdots, A_m, +)} > 1 \tag{16}$$

Denote the left side of the above inequality by L_1.

Let $A_1 = \bar{a}_1$, we have the similar inequality below.

$$\prod^{even} \frac{p_G(\bar{a}_1, A_2, \cdots, A_m, +)}{p_G(\bar{a}_1, A_2, \cdots, A_m, -)} \prod^{odd} \frac{p_G(\bar{a}_1, A_2, \cdots, A_m, -)}{p_G(\bar{a}_1, A_2, \cdots, A_m, +)} > 1 \tag{17}$$

Let us denote the left side by L_2. Notice that L_1 and $\frac{1}{L_2}$ are almost the same except a_1 and $\bar{a}_1$. Next we try to prove that $L_1 = \frac{1}{L_2}$ by showing that all items containing a_1 or $\bar{a}_1$ will be cancelled out in both L_1 and L_2.

Let A_{m+1} be + or −. Note that $p_G(A_1, \cdots, A_m, A_{m+1})$ can be decomposed in terms of (10) and (11), so all items in the inequalities are in the form of $p(A_i|pa_i)$, and none of them is zero (otherwise, it is impossible for A_1, $\cdots$, A_m, A_{m+1} to form an m-XOR). For the item $p(A_i|pa_i)$ in which both a_1 and $\bar{a}_1$ do not occur, it is obvious that if that item occurs in the numerator of L_1, it should occur in the denominator of L_2. Thus, we only need to consider the items containing a_1 or $\bar{a}_1$. There are two cases.

(1) The items in the form of $p(A_1|pa_1)$ occur, where A_1 is a_1 or $\bar{a}_1$. Since A_1 has at most $m-1$ parents and A_{m+1} should be in pa_1 (otherwise this term will be cancelled out from f_G), so there is at least one attribute other than A_{m+1} which is not in pa_1. Let $\bar{pa}_1$ denote all the attributes that are not in pa_1, and t be the number of such attributes ($t \geq 1$). Then there are 2^{t-1} assignments of $\bar{pa}_1$ to make the number of attribute A_i taking a_i ($i \neq 1$) even, and 2^{t-1} assignments odd. Thus, each $p(A_1|pa_1)$ occurs 2^{t-1} times in both the numerator and denominator of L_1, and is therefore cancelled out. The same situation happens for L_2.

(2) The items in the form $p(A_i|pa_i)$ occur, where $i \neq 1$ and A_1 is in pa_i. Similar to (1), since there is at least one attribute other than A_{m+1} not in pa_i, each $p(A_i|pa_i)$ occurs the same times in the numerator and denominator of L_1 or L_2, and thus is cancelled out.

Therefore, $L_1 = \frac{1}{L_2}$. It is impossible to satisfy both inequalities (16) and (17). Therefore, we conclude that no such ANB can represent f.

□

Now we are ready to prove the main theorem about the representational upper bound of BNs.

Theorem 3 *For any BN G, $O_r(G) \leq O_s(G)$.*

Proof: suppose that G is a BN on discrete attributes A_1, $\cdots$, A_n and $O_s(G) = m-1$, i.e., each node has at most $m-1$ parents. Let f be a function represented in G, we will prove by contradiction that f cannot contain an m-XOR. Assume that f contains an m-XOR from A_1, $\cdots$, A_m to A_{m+1}. According to Definition 7, there would be two values

a_i and $\bar{a}_i$ for each attribute A_i, $1 \leq i \leq m+1$, such that the partial function f_G of G's classification function from $\{a_1, \bar{a}_1\} \times \cdots \times \{a_m, \bar{a}_m\}$ to $\{a_{m+1}, \bar{a}_{m+1}\}$ is an m-XOR, while $A_k = a_k$ $(m+1 < k \leq n)$. To simplify our notation, we denote a_{m+1}, $\bar{a}_{m+1}$ by $+$ and $-$ respectively. Here we have two types of attributes: unfixed attributes A_1, $\cdots$, A_m and A_{m+1} that compose an m-XOR, and fixed attributes A_{m+2}, $\cdots$, A_n that have values a_{m+2}, $\cdots$, a_n respectively.

Obviously all of A_1, $\cdots$, A_m should be in A_{m+1}'s Markov blanket, and we only need to consider the node in A_{m+1}'s Markov blanket.

Consider $A_1 = a_1$ and $A_1 = \bar{a}_1$, we have the inequalities below similar to (16) and (17):

$$\prod^{odd} \frac{p_G(a_1, A_2, \cdots, A_m, a_{m+2} \cdots, a_n, +)}{p_G(a_1, A_2, \cdots, A_m, a_{m+2}, \cdots, a_n, -)} \prod^{even} \frac{p_G(a_1, A_2, \cdots, A_m, a_{m+2}, \cdots, a_n, -)}{p_G(a_1, A_2, \cdots, A_m, a_{m+2}, \cdots, a_n, +)} > 1 \qquad (18)$$

$$\prod^{even} \frac{p_G(\bar{a}_1, A_2, \cdots, A_m, a_{m+2}, \cdots, a_n, +)}{p_G(\bar{a}_1, A_2, \cdots, A_m, a_{m+2}, \cdots, a_n, -)} \prod^{odd} \frac{p_G(\bar{a}_1, A_2, \cdots, A_m, a_{m+2}, \cdots, a_n, -)}{p_G(\bar{a}_1, A_2, \cdots, A_m, a_{m+2}, \cdots, a_n, +)} > 1. \qquad (19)$$

where *odd*/*even* specifies that the number of A_i taking a_i is odd or even, $2 \leq i \leq m$. Let us denote the left sides of the above two inequalities by L_3 and L_4 respectively. Similarly, we try to prove that $L_3 = \frac{1}{L_4}$, and we also only need consider the items containing a_1 or $\bar{a_1}$. (18) and (19) are different from (16) and (17) only in that there exist fixed attributes.

Let $A_1 = a_1$. There are three cases that a fixed attribute occurs in an item together with a_1 in L_3 or L_4.

(1) A fixed attribute has A_1 and some other attributes (fixed or unfixed) as its parents. That is, the items in the form of $p(a_k|pa_k^u, pa_k^f)$ occur, where a_k is an the value of a fixed attribute A_k $(k > m+1)$, and pa_k^u are the parents of A_k that are unfixed attributes and $A_1 \in pa_k^u$, and pa_k^f are the parents of A_k that are fixed attributes. Since A_k has at most $m-1$ parents and A_{m+1} should be one of them, there are at least two attribute in $\{A_2, \cdots, A_m\}$ not in pa_k.[3] Based on the same reason in proving Lemma 2, these items occur the same times in the numerator and denominator of L_3, and thus are cancelled out. The similar situation happens for L_4.

(2) An unfixed attribute has A_1 and some other attributes (fixed or unfixed) as its parents. That is, the items in the form of $p(A_k|pa_k^u, pa_k^f)$ occur, where A_k is an unfixed attribute $(k < m+1)$, and pa_k^u are the parents of A_k that are unfixed attributes and $A_1 \in pa_k^u$, and pa_k^f are the parents of A_k that are fixed attributes. Since A_k has at most $m-1$ parents and A_{m+1} should be one of them, there is at least one attribute in $\{A_2, \cdots, A_m\} - \{A_k\}$ not in pa_k.[4] Based on the same reason in proving Lemma 2, these items occur the same times in the numerator and denominator of L_3, and thus are cancelled out. The similar situation happens for L_4.

(3) The fixed attributes are the parents of A_1. That is, the items in the form of $p(a_1|pa_1^u, pa_1^f)$ occur. Since A_{m+1} should be in pa_1^u, there is at least one attribute in $\{A_2, \cdots, A_m\}$ not in pa_1.[5] Similarly, those items will be cancelled out in both L_3 and L_4.

[3] For Corollary 1, there is at least one attribute in $\{A_2, \cdots, A_m\}$ not in pa_k.

[4] For Corollary 1, there is also at least one attribute in $\{A_2, \cdots, A_m\} - \{A_k\}$ not in pa_k.

[5] For Corollary 1, since A_1 has at most $m-1$ parents from $\{A_2, \cdots, A_m, A_{m+1}\}$ in which one is A_{m+1} and others are from $\{A_2, \cdots, A_m\}$, there is also at least one of $\{A_2, \cdots, A_m\}$ not in pa_1.

For $A_1 = \bar{a}_1$, we have the similar result. Thus, $L_3 = \frac{1}{L_4}$. It is impossible to satisfy both inequalities (18) and (19). Therefore, we conclude that no such BN can represent f.
□

Theorem 3 presents a representational upper bound for general BNs, thus establishing an explicit association between the structural complexity of a BN and its representational capacity.

Theorem 3 can be further extended to the following corollary.

Corollary 1 *If an n-dimensional discrete function f has an order of m (contains an m-XOR), with attributes A_1, $\cdots$, A_m and A_{m+1} forming the m-XOR, $m \geq 2$, then no BN of order m, with attributes A_1, $\cdots$, A_m and A_{m+1} having at most $m-1$ parents from $\{A_1, \cdots, A_m, A_{m+1}\}$, can represent f.*

Corollary 1 shows that, a BN of order m cannot represent a function of order m, if each of the nodes forming the m-XOR has at most $m-1$ parents from the nodes forming the m-XOR. The proof is similar to the proof of Theorem 3, and the differences are indicated in footnotes in the proof. Of course, such a function of order m might still be represented by a BN of order m with a different structure. In fact, it is our conjecture on the lower bound of BNs: any function of order m can be represented by a BN of order m.

5. Conclusions

In this paper, we discuss the representational power of discrete BNs. We use the maximum number of parents of a node in a BN as the measure for its structural complexity, and the order of the maximum XOR contained in a target function as the measure for complexity of the target function. Then we establish a relation between the structural complexity and the representational power of Bayesian networks, by proposing and proving a representational upper bound of BNs. Roughly speaking, any BNs of order m cannot represent a target function of order $m+1$. Moreover, a Bayesian network of order m cannot represent a target function of order m, if each of the nodes that forms the m-XOR has at most $m-1$ parents from the nodes forming the m-XOR. Our results show the ultimate limitation of BNs in representing discrete classification functions.

Our theoretical results establish a clear association between the topology of Bayesian networks and the complexity of functions that they can represent. They help us to understand the limitation of Bayesian networks. In addition, our results can be useful in real-world applications. Before we learn a Bayesian network from data, we often need to decide the structure of the network. Our results suggest to detect the number of n-XOR ($n = 2, 3, \cdots$) contained in the data. If there exists 2-XOR, then no Naive Bayes can learn it perfectly. However, if the number of 2-XOR is small, then Naive Bayes might still be proper to learn it (Zhang and Ling, 2001a); otherwise, more complex structures, such as TAN, should be chosen.

We give only a representational upper bound of BNs in this paper. A natural question is: what is the representational lower bound of BNs? An interesting and intuitive conjecture is that any function of order m can be represented by some BN of order m. This conjecture is correct for Naive Bayes and TAN, but the general case has not been proved, and it is one of our future research interests.

Additional interesting future work is to determine the representational power of Bayesian networks with hidden nodes. Intuitively, BNs with hidden nodes have a higher representational power.

References

P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. *Machine Learning*, 29:103–130, 1997.

Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* A Wiley-Interscience Publication, 1973.

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

M. S. Paterson. *Boolean Function Complexity.* Cambridge University Press, 1992.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kauhmann, 1988.

M. A. Peot. Geometric implications of the Naive Bayes assumption. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pages 414–419. Morgan Kaufmann, 1996.

V. N. Vapnik and A. Chevonenkis. On the uniform convergence of relative frequencie of events to their probabilities. *Theory of Probability and Its Applications*, 16:264–280, 1971.

H. Zhang and C. X. Ling. Geometric properties of Naive Bayes in nominal domains. In L. De Raedt and P. Flach, editors, *Proceedings of 12th European Conference on Machine Learning*, pages 588–599. Springer, 2001a.

H. Zhang and C. X. Ling. Learnability of augmented Naive Bayes in nominal domains. In C. E. Brodley and A. P. Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 617–623. Morgan Kaufmann, 2001b.