Learnability of Augmented Naive Bayes in Nominal Domains

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

Abstract

It is well-known that Naive Bayes can only represent linearly separable functions in binary domains. But the learnability of general Augmented Naive Bayes is open. Little work is done on the learnability of Bayesian networks in nominal domains, a general case of binary domains. This paper explores the learnability of Augmented Naive Bayes in nominal domains. We introduce a complexity measure for nominal functions, and prove upper bounds of the learnability of Augmented Naive Bayes in terms of that measure. Our results deepen our theoretical understanding of the learnability (and limitations) of Naive Bayes, Tree Augmented Naive Bayes, and general Augmented Naive Bayes with different levels of complexity.

1. Introduction

Classification is a fundamental issue in machine learning and pattern recognition. In classification learning problems, a learner attempts to construct a classifier from a given set of training examples with class labels. Assume 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. There are two types of attributes: nominal attributes (taking values from a finite set) and numeric attributes (taking real-number values). In this paper, we restrict our discussion to nominal attributes, a general case of binary attributes. We use C to represent the classification variable, which takes value $+$ (positive class) or $-$ (negative class). We use c to represent the value that C takes.

A classifier is a function that assigns a class label to an example. There are numerous approaches to learning classifiers, such as decision trees, neural networks, rule induction, and so on. From the probability perspective, according to the Bayesian Theorem, the probability of an example $E = (a_1, a_2, \cdots, a_n)$ being class c is

$$p(c|E) = \frac{p(a_1, a_2, \cdots, a_n|c)p(c)}{p(a_1, a_2, \cdots, a_n)}$$

E is classified as the class $C = +$ iff

$$g(E) = \frac{p(C = +|a_1, a_2, \cdots, a_n)}{p(C = -|a_1, a_2, \cdots, a_n)} \geq 1$$

where $g(E)$ is called a Bayesian classifier.

Assume all attributes are independent given class; that is,

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

the resulting $g(E)$ is then:

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

$g(E)$ is called a Naive Bayesian classifier, or simply Naive Bayes. Figure 1 shows an example of Naive Bayes.

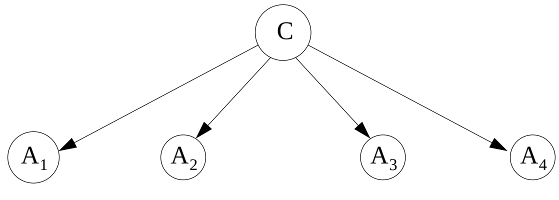

Figure 1. An example of Naive Bayes

Because values of $p(a_i|c)$ can be estimated from the training examples, Naive Bayes is easy to construct. It is also, however, surprisingly effective. Many empirical comparisons between Naive Bayes and decision tree algorithms such as C4.5 (Quinlan, 1993) showed that Naive Bayes predicts just as well as C4.5 (Langley et al., 1992; Kononenko, 1990; Pazzani et al., 1996). However, the independence assumption hardly holds true in most artificial and real-world datasets. For example, Frank et al. (2000) evaluated the performance of Naive Bayes on regression problems and compared it to numerical estimators on many artificial and real-world datasets, and found that it produces worse estimations than other methods.

Domingos and Pazzani (1997) provided an explanation on the good predictive performance of Naive Bayes on classification tasks. They found that even though Naive Bayes' assumption alters the probability distribution of a class, the class with the maximum probability may still be the same. That is, under the MAP (Maximum A Posteriori) Principle, the classification error of Naive Bayes can be very small.

In recent years, efforts have been made to improve Naive Bayes by reducing the negative impact of the independence assumption, mainly in two directions (Cheng & Greiner, 1999). One is to select a subset of attributes which are independent, instead of using all of the attributes (Langley & Sage, 1994; Kohavi & John, 1997). The second approach is to extend the structure of Naive Bayes to account for dependency between attributes. Friedman et al. (1997) presented a tree-like structure, Tree Augmented Naive Bayes (TAN), in which the classification node directly points to all attributes and an attribute can have only one parent from another attribute (in addition to the classification node).[1] Figure 2 shows an example of TAN. TAN is a specific case of general Augmented Naive Bayesian network, or simply Augmented Naive Bayes, in which the classification node also directly points to all attributes, but there is no limitation on links among attributes (except they do not form any directed cycle).

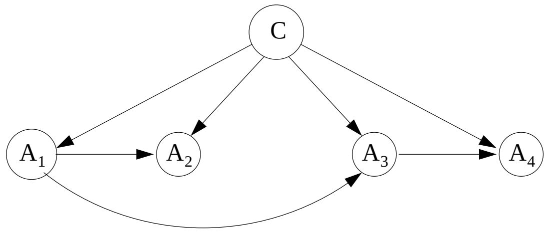

Figure 2. An example of TAN

A fundamental and open question is the learnability of various classes of Augmented Naive Bayes. It is well-known that Naive Bayes is limited in expressing linear functions in binary domains (Duda & Hart, 1973). What about TANs, or general Augmented Naive Bayes with more complex structure? To our knowledge, there is no general result on the learnability of various types of Augmented Naive Bayes. This paper establishes a relation between the complexity of the target functions and the learnability of Augmented Naive Bayes. Roughly speaking, if the target function contains an XOR function with a maximum of k variables, then no Augmented Naive Bayes with nodes having at most $k-2$ parents can represent it.

The results in this paper are more general in the sense that they are based on nominal domains, a general case of binary domains. Assume $A_1, A_2, \cdots, A_n$ are n nominal attributes, each attribute A_i may have m values $a_{i1}, a_{i2}, \cdots, a_{im}$, $m \geq 2$. There is little previous work on the learnability of Bayesian network in nominal domains. An exception is Domingos and Pazzani (1997)'s work, which introduced m new Boolean attributes $B_{i1}, B_{i2}, \cdots, B_{im}$, for each attribute A_i, and then proved that Naive Bayes is a linear classifier over those new attributes. However, linear separability on the original attributes was transformed to new attributes with a higher dimension.

The remainder of this paper is organized as follows. We first introduce a complexity measure for nominal functions (Section 2.1), and present an upper bound of the learnability of Naive Bayes in terms of that measure (Section 2.2). We then extend our results to Tree Augmented Naive Bayes (TAN) (Section 2.3), and last to general Augmented Naive Bayes (Section 2.4). In the conclusions (Section 3), we summarize our work and discuss a few issues for future research.

2. Learnability of Augmented Naive Bayes

This section attempts to answer two questions: Are there any functions which are inherently hard to learn by Augmented Naive Bayes? If there are, how to measure their complexity?

2.1 Complexity Measure

We first introduce the definition of nominal functions, which are the basis of our paper.

Definition 1 *Given n nominal attributes $A_1, A_2, \cdots, A_n$, and two classification labels $\{+,-\}$, a function f from $A_1 \times A_2 \cdots \times A_n$ to $\{+,-\}$ is called n-dimensional nominal function.*

How to measure the complexity of a nominal function for the purpose of learning Augmented Naive Bayes? VC-dimension is a widely used measure for hypothesis space complexity (Vapnik & Chevonenkis, 1971), but it might be too general. There is some work on complexity measure of Boolean functions from the view-

[1] In this paper, the number of parents of a node does not include the classification node.

point of computational complexity (Paterson, 1992), which uses the size of the minimum combinational network which computes the function.[2] This does not seems to be simple enough to measure the complexity with respect to the difficulty in Bayesian learning.

We notice that for Bayesian learning of nominal functions, the discriminating granularity of attributes to determine the classification seems to reflect the complexity of the functions. That is, the number of attributes on which a function depends in order to determine a class (assigning a unique class label to an example) is important. Intuitively, the more attributes that are needed, the more complex is the function. One very simple case is that one attribute can determine a class. An extremely complex case is that a class cannot be determined unless all of the attributes have been known. In binary domains, XOR is such a function that requires values of all of its variables to be known. In this paper, if f_b is an n-variable XOR function on n binary attributes, we call f_b an n-XOR, and we call n the order of the n-XOR. By this notation, 2-XOR is a regular (2-variable) XOR. Often, n-XOR is also called parity function with n variables, which returns 1 if and only if an even number of variables are 1.

We propose to use the highest order of XOR "contained" in a nominal function as its complexity measure; that is, the maximum subset of nominal attributes consisting of a XOR pattern.

Definition 2 *Assume f is an n-dimensional nominal function on A_1, A_2, $\cdots$, A_n. An (n-1)-dimensional partial function f_p on A_1, $\cdots$, A_{i-1}, A_{i+1}, $\cdots$, A_n, 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 $1 \leq i \leq n$.*

Figure 3 shows the intuitively geometric meaning of a 2-dimensional subfunction in which the shaded plane corresponds to a 2-dimensional subfunction $f(a_2)$ in three dimensions. Similarly, we can get an arbitrary k-dimensional subfunction of f, by fixing $n-k$ attributes, where $2 \leq k \leq n-1$.

We are now ready to give a formal definition for a nominal function to "contain" a k-XOR function.

Definition 3 *An n-dimensional nominal 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}, such that a partial function $f_{p'}$ of f_p on $\{a_{k1_1}, a_{k1_2}\} \times \cdots \times \{a_{kk_1}, a_{kk_2}\}$ is a k-XOR function.*

Definition 4 *An n-dimensional nominal function f is said to have an order of m, if the maximum XOR it contains is an m-XOR.*

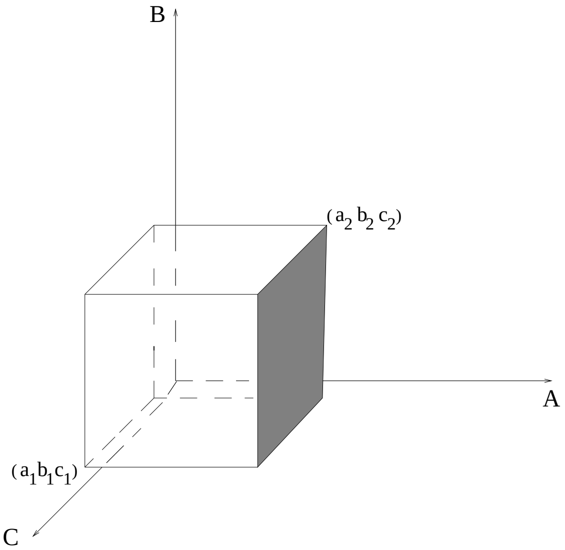

Figure 3. Geometric meaning of 2-dimensional subfunction

The above complexity measure for a nominal function is different from the one that uses the size of the minimum combinational network computing the function (Paterson, 1992) in two aspects. First, it is applicable for arbitrary nominal functions, a general case of Boolean functions. Second, it is simpler, since we only consider the part of a function that consists of the highest order XOR, instead of the whole function. We find that this measure is appropriate (as we will see below) to Bayesian learning.

2.2 Naive Bayes' Learnability

Using the order of a nominal function as the measure of its complexity, we now establish an upper bound on the learnability of Naive Bayes.

Theorem 1 *If an n-dimensional nominal function f has an order of 2 (contains a 2-XOR), then no Naive Bayes can represent f.*

Proof: Suppose A_1, A_2,$\cdots$, A_n are the nominal attributes of f. By Definition 3, if f contains a 2-XOR, then there is a 2-dimensional subfunction which contains a 2-XOR. Then there are two attributes A_i and A_j, and each of them has two values, a_{i1}, a_{i2}, and a_{j1}, a_{j2}, respectively, such that:

$$f(a_1, \cdots, a_{i1}, \cdots, a_{j1}, \cdots, a_n) = + \quad (1)$$

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

$$f(a_1, \cdots, a_{i1}, \cdots, a_{j2}, \cdots, a_n) = - \quad (2)$$
$$f(a_1, \cdots, a_{i2}, \cdots, a_{j1}, \cdots, a_n) = - \quad (3)$$
$$f(a_1, \cdots, a_{i2}, \cdots, a_{j2}, \cdots, a_n) = + \quad (4)$$

where a_k is a value of A_k, $k \neq i$ and j.

Suppose that there were a Naive Bayes to represent f, then the following inequalities would be true.

$$\frac{p(+)p(a_{i_1}|+)p(a_{j1}|+)\prod_{k\neq i,j} p(a_k|+)}{p(-)p(a_{i_1}|-)p(a_{j1}|-)\prod_{k\neq i,j} p(a_k|-)} \geq 1 \quad (5)$$
$$\frac{p(+)p(a_{i_1}|+)p(a_{j2}|+)\prod_{k\neq i,j} p(a_k|+)}{p(-)p(a_{i_1}|-)p(a_{j2}|-)\prod_{k\neq i,j} p(a_k|-)} < 1 \quad (6)$$
$$\frac{p(+)p(a_{i_2}|+)p(a_{j1}|+)\prod_{k\neq i,j} p(a_k|+)}{p(-)p(a_{i_2}|-)p(a_{j1}|-)\prod_{k\neq i,j} p(a_k|-)} < 1 \quad (7)$$
$$\frac{p(+)p(a_{i_2}|+)p(a_{j2}|+)\prod_{k\neq i,j} p(a_k|+)}{p(-)p(a_{i_2}|-)p(a_{j2}|-)\prod_{k\neq i,j} p(a_k|-)} \geq 1 \quad (8)$$

It is easy to verify that none of the terms in the above inequalities can be zero, since otherwise (1), (2), (3) and (4) would not hold.

Divide (5) by (6), we have:

$$\frac{p(a_{j1}|+)}{p(a_{j2}|+)} > \frac{p(a_{j1}|-)}{p(a_{j2}|-)} \quad (9)$$

Divide (8) by (7), we have:

$$\frac{p(a_{j2}|+)}{p(a_{j1}|+)} > \frac{p(a_{j2}|-)}{p(a_{j1}|-)} \quad (10)$$

It is impossible to satisfy both inequalities (9) and (10). Therefore, we conclude that no Naive Bayes can represent f.
□

Actually, we can extend Theorem 1 further.

Definition 5 *Assume f is an n-dimensional nominal function on A_1, A_2, $\cdots$, A_n. f is said to contain a diagonal 2-XOR, if there are two attributes A_i and A_j, and each of them has two values, a_{i1}, a_{i2}, and a_{j1}, a_{j2}, respectively, such that:*

$$f(a_{11}, \cdots, a_{i1}, \cdots, a_{j1}, \cdots, a_{n1}) = + \quad (11)$$
$$f(a_{11}, \cdots, a_{i1}, \cdots, a_{j2}, \cdots, a_{n1}) = - \quad (12)$$
$$f(a_{12}, \cdots, a_{i2}, \cdots, a_{j1}, \cdots, a_{n2}) = - \quad (13)$$
$$f(a_{12}, \cdots, a_{i2}, \cdots, a_{j2}, \cdots, a_{n2}) = + \quad (14)$$

where a_{lk}, $k = 1, 2$, are two different values of A_l, $l \neq i$ and j.

Figure 4 shows intuitively geometric meaning of a diagonal 2-XOR (the shaded plane) in 3-dimensional space. Notice that for a diagonal 2-XOR, the four points consisting of the 2-XOR are not on the same vertical or horizontal plane, different from the situation in Definition 2.

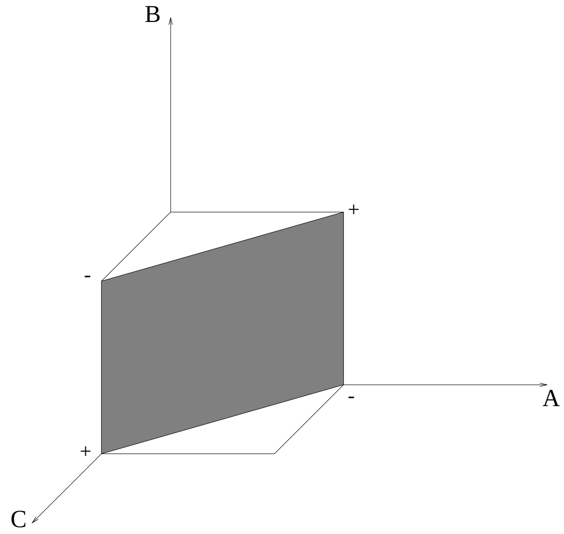

Figure 4. Geometric meaning of diagonal 2-XOR

Corollary 1 If an n-dimensional nominal function f contains a diagonal 2-XOR, then no Naive Bayes can represent f.

Proof: We can verify that no Naive Bayes can satisfy Equation 11, 12, 13, 14, in the same way as in proving Theorem 1.
□

Theorem 1 and Corollary 1 establish an upper bound on the learnability of Naive Bayes. Intuitively, the condition that an n-dimensional nominal function contains a 2-XOR is pretty weak. We can see that by the quantitative analysis below. Assume f is a randomly generated function on n binary attributes. Then the number of f's 2-dimensional subfunctions is:

$$m = \binom{n}{2} \times 2^{n-2}$$

The probability p that a 2-dimensional subfunction is a 2-XOR is 0.125. So the probability that f contains a 2-XOR is:

$$p_{xor} = \sum_{k=1}^{m} (-1)^{k-1} \binom{m}{k} p^k$$

It is obvious that when n increases, p_{xor} will rapidly approach to 1. Actually, when $n = 4$, $p_{xor} = 0.9594$. It might be easy for a high dimensional function to

Table 1. The Conditional Probability Table for $p(B|A, c)$.

	a_1, b_1	a_1, b_2	a_2, b_1	a_2, b_2
$c = -$	0.4	0.6	0.6	0.4
$c = +$	0.6	0.4	0.4	0.6

contain a 2-XOR. Therefore, the learnability of Naive Bayes is quite limited.

On the other hand, Naive Bayes cannot learn some functions even not containing a 2-XOR. An example is m-of-n concepts. An m-of-n concept is a Boolean concept that is true if m or more out of n Boolean variables are true. Clearly, it does not contain 2-XOR. Domingos and Pazzani (1997) showed that for the concept 8-of-25, when the input Boolean variables have just six or seven 1's, Naive Bayes gives an incorrect answer of 1 (instead of 0). This fact makes it difficult to set a lower bound for Naive Bayes.

2.3 Upper Bound for Tree Augmented Naive Bayes

Tree Augmented Naive Bayes (TAN) is a tree-like structure with more complex structure than Naive Bayes (Friedman et al., 1997). However, TAN's structure is limited to express dependency since an attribute can have only one parent (in addition to the classification node). Intuitively, TAN is more powerful in representation than Naive Bayes. However, it is still open as to what TAN can represent but Naive Bayes cannot, and what the limitation of TAN is.

Assume G is a TAN on attributes $A_1, A_2, \cdots, A_n$. The corresponding classifier $G(E)$ is then:

$$G(E) = \frac{p(C=+)}{p(C=-)} \prod_{i=1}^{n} \frac{p(a_i|C=+, pa(A_i))}{p(a_i|C=-, pa(A_i))} \qquad (15)$$

where $pa(A_i)$ is the value of A_i's parent. If A_i does not have a parent, $pa(A_i)$ is empty. It is easy to verify that Theorem 1 is not true for TAN. That is, 2-order nominal functions are not the upper bound of the learnability of TAN. The following is an example of TAN representing 2-XOR.

Consider a TAN G on two binary attributes A and B, where $A = \{a_1, a_2\}$, $B = \{b_1, b_2\}$, and A is the parent of B. Let $p(+) = p(-) = 0.5$, $p(a_1|+) = p(a_2|+) = 0.5$ and $p(a_1|-) = p(a_2|-) = 0.5$. Table 1 shows the Conditional Probability Table (CPT) for $p(B|A, c)$. It can be easily verified that G represents a 2-XOR.

However, 3-order nominal functions are the upper bound of TAN, as proved in the following theorem.

Theorem 2 *If an n-dimensional nominal function f has an order of 3 (contains a 3-XOR), then no TAN can represent f.*

Proof: Suppose A_1, A_2,$\cdots$, A_n are the nominal attributes of f. By Definition 3, if f contains a 3-XOR, then there is a 3-dimensional subfunction which contains a 3-XOR. That is, there are three attributes A_i, A_j and A_k, and each of them has two values, a_{i1}, a_{i2}; a_{j1}, a_{j2}; a_{k1}, a_{k2}, respectively, such that:

$$\begin{aligned}
f(a_1, \cdots, a_{i1}, \cdots, a_{j1}, \cdots, a_{k1}, \cdots, a_n) &= + \\
f(a_1, \cdots, a_{i1}, \cdots, a_{j1}, \cdots, a_{k2}, \cdots, a_n) &= - \\
f(a_1, \cdots, a_{i1}, \cdots, a_{j2}, \cdots, a_{k1}, \cdots, a_n) &= - \\
f(a_1, \cdots, a_{i1}, \cdots, a_{j2}, \cdots, a_{k2}, \cdots, a_n) &= + \\
f(a_1, \cdots, a_{i2}, \cdots, a_{j1}, \cdots, a_{k1}, \cdots, a_n) &= - \\
f(a_1, \cdots, a_{i2}, \cdots, a_{j1}, \cdots, a_{k2}, \cdots, a_n) &= + \\
f(a_1, \cdots, a_{i2}, \cdots, a_{j2}, \cdots, a_{k1}, \cdots, a_n) &= + \\
f(a_1, \cdots, a_{i2}, \cdots, a_{j2}, \cdots, a_{k2}, \cdots, a_n) &= -
\end{aligned}$$

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

Suppose there were a TAN G which can represent f, then G should satisfy the above equations. Since the values of all the attributes are fixed, except A_i, A_j and A_k, we can easily construct another TAN only on A_i, A_j and A_k from G, which is equal to the original TAN. So we only need to consider the arcs between A_i, A_j and A_k. Since there are at most two arcs between A_i, A_j and A_k, for each case, we can verify that it is impossible to satisfy the above equations simultaneously in the same way as in proving Theorem 1.

Therefore, it is impossible for a TAN to represent a nominal function with an order of 3.
□

2.4 General Augmented Naive Bayes

General Augmented Naive Bayes extends Naive Bayes in that it has no limitation on links among nominal attributes (except they do not form a directed cycle). Although its structure is a special case of arbitrary Bayesian networks, it does not have limitations in its representation. The following discussion shows that an arbitrary Bayesian network can be represented by an Augmented Naive Bayes (with different structures).

A Bayesian classifier classifies an example E based on $p(c|E)$:

$$p(c|E) = \frac{p(a_1, a_2, \cdots, a_n, c)}{p(a_1, a_2, \cdots, a_n)}$$

So the classification is determined only by the joint probability distribution $p(a_1, a_2, \cdots, a_n, c)$. According to product rule, we have :

$$p(a_1, a_2, \cdots, a_n, c) = p(c)p(a_1|c)p(a_2|a_1, c)\cdots p(a_n|a_{(n-1)}, \cdots, a_1, c)$$

Obviously, it can be represented by an Augmented Naive Bayes G, in which c points to all attributes A_1, A_2, $\cdots$, A_n, and each A_i has parents A_1, $\cdots$, A_{i-1}.

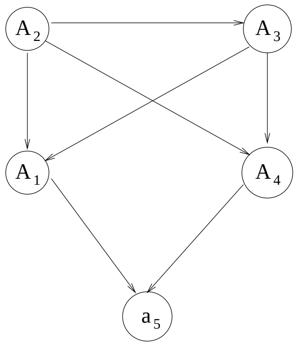

Figure 5. An example of 2-order Augmented Bayes

We expect to extend Theorems 1 and 2 to general Augmented Naive Bayes. The pattern of Theorems 1 and 2 is quite clear: if a nominal function contains an m-XOR, then no Augmented Naive Bayes with nodes having at most $m-2$ parents can represent it. However, we have not been able to prove it. We present it here as a conjecture and open question.

Conjecture 1 *If an n-dimensional nominal function f has an order of m (contains an m-XOR), then no Augmented Naive Bayes with nodes having at most $m-2$ parents can represent f.*

Our reviewer gives an interesting example to test the correctness of the conjecture. We change it slightly (make it more complex) and show it in Figure 5. The Augmented Naive Bayes G in the figure is on $\{A_1, A_2, A_3, A_4, A_5\}$ and 4-XOR is defined on $\{A_1, A_2, A_3, A_4\}$ while A_5 is fixed to a_5. Obviously, fixing the value of A_5 creates dependence between A_1 and A_4. It seems that one of A_1 and A_4 would have three parents, so G might be able to represent a 4-XOR. However, it can be verified that G cannot represent a 4-XOR. Actually, we have obtained a result similar to Theorem 2 for 4-XOR. We omit it in this paper, since it is not significant after proving Theorem 2.

On the other hand, we are able to prove a weaker version of Conjecture 1, which limits the total number of nominal attributes to m.

Theorem 3 *If an m-dimensional nominal function f has an order of m (contains an m-XOR), then no Augmented Naive Bayes with m nodes each of which has at most $m-2$ parents can represent f.*

Proof: Suppose that A_1, A_2,$\cdots$, A_m are the nominal attributes of f. We apply induction on m.

When $m = 2$, there is no parent for each node. The statement is true according to Theorem 1.

Suppose when $m = k-1$, the statement is true. Consider $m = k$. Let G be an Augmented Naive Bayes with k nodes and each node of G has at most $k-2$ parents. Suppose G could represent f with an order of k. Obviously, G has at most two nodes with $k-2$ parents. In this case, we assume these two nodes are A_i and A_j. Then all other attributes have arcs pointing to both A_i and A_j, and there exists a node A_r, where $r \neq i$, j, having no parent (because G is acyclic).

Let us fix A_r's value to be a_r. We can construct another Augmented Naive Bayes G' on $\{A_1, \cdots, A_k\}$-$\{A_r\}$ from G, and G' represents the $(k-1)$-dimensional subfunction $f(a_r)$ and each node of G' has at most $k-3$ parents. Since f has an order of k, $f(a_r)$ has an order of $(k-1)$. Therefore G' can represent a nominal function with an order of $(k-1)$. This is contradictory to the assumption.

Similarly, we can prove the case where only one node has $k-2$ parents. Therefore, the statement is true for all m.
□

The above conjecture and theorem provide us the upper bound of the learnability of general Augmented Naive Bayes in nominal domains. Intuitively, nominal functions with a higher order XOR present more difficulty to Augmented Naive Bayes, regardless of learning algorithms. Therefore, the size of the XOR contained in a nominal function is an adequate measure for the learnability of Augmented Naive Bayes.

3. Conclusions

We investigated the learnability of Augmented Naive Bayes by means of introducing a complexity measure on the basis of the XOR functions. We proved that the nominal functions containing XOR are inherently hard to learn. More specifically, nominal functions with k-variable XOR cannot be represented by Augmented Naive Bayes with nodes having at most $k-2$ parents. This sets an upper bound on the learnability of Augmented Naive Bayes in nominal domains.

A few questions still remain unanswered. The conjecture remains unproved. We only gave the upper bound of the learnability of Augmented Naive Bayes. What is the reachable upper bound? What is the lower bound? We also need to consider situations with noise. In our future work, we will investigate these questions.

Acknowledgements

We thank our reviewers for their valuable suggestions to improve this paper.

References

Cheng, J., & Greiner, R. (1999). Comparing bayesian network classifiers. In K. B. Laskey and H. Prade (Eds.), *Proceedings of the fifteenth conference on uncertainty in artificial intelligence*, 101–108. Morgan Kaufmann.

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

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis.* A Wiley-Interscience Publication.

Frank, E., Trigg, L., Holmes, G., & Witten, I. H. (2000). Naive bayes for regression. *Machine Learning, 41(1)*, 5–15.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning, 29*, 131–163.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence Journal, 97*, 273–324.

Kononenko, I. (1990). Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. In B. Wielinga (Ed.), *Current trends in knowledge acquisition.* IOS Press.

Langley, P., Iba, W., & Thomas, K. (1992). An analysis of Bayesian classifiers. In *Proceedings of the tenth national conference of artificial intelligence*, 223–228. AAAI Press.

Langley, P., & Sage, S. (1994). Induction of selective bayesian classifiers. In *Proceedings of uncertainty in artificial intelligence 1994.* Morgan Kaufmann.

Paterson, M. S. (1992). *Boolean function complexity.* Cambridge University Press.

Pazzani, M., Muramatsu, J., & Billsus, B. (1996). Syskill & webert: Identifying interesting web sites. In *Proceedings of the thirteen national conference of artificial intelligence*, 54–62. AAAI Press.

Quinlan, J. (1993). *C4.5: Programs for machine learning.* Morgan Kaufmann: San Mateo, CA.

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