Finite Mixture Model of Bounded Semi-naive Bayesian Networks Classifier

Kaizhu Huang, Irwin King, and Michael R. Lyu

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, New Territories, Hong Kong
{kzhuang, king, lyu}@cse.cuhk.edu.hk

Abstract. The Semi-Naive Bayesian network (SNB) classifier, a probabilistic model with an assumption of conditional independence among the combined attributes, shows a good performance in classification tasks. However, the traditional SNBs can only combine two attributes into a combined attribute. This inflexibility together with its strong independency assumption may generate inaccurate distributions for some datasets and thus may greatly restrict the classification performance of SNBs. In this paper we develop a Bounded Semi-Naive Bayesian network (B-SNB) model based on direct combinatorial optimization. Our model can join any number of attributes within a given bound and maintains a polynomial time cost at the same time. This improvement expands the expressive ability of the SNB and thus provide potentials to increase accuracy in classification tasks. Further, aiming at relax the strong independency assumption of the SNB, we then propose an algorithm to extend the B-SNB into a finite mixture structure, named Mixture of Bounded Semi-Naive Bayesian network (MBSNB). We give theoretical derivations, outline of the algorithm, analysis of the algorithm and a set of experiments to demonstrate the usefulness of MBSNB in classification tasks. The novel finite MBSNB network shows a better classification performance in comparison with than other types of classifiers in this paper.

1 Introduction

Learning accurate classifiers is one of the basic problems in machine learning. The Naive Bayesian network (NB) [8] shows a good performance in dealing with this problem when compared with the decision tree learner C4.5 [13]. With an independency assumption among the attributes, when given the class label, NB classifies a specific sample into the class with the largest joint probability. This joint probability can be decomposed into a multiplication form based on its independency assumption.

The success of NB is somewhat unexpected since its independency assumption typically does not hold in many cases. Furthermore, the so-called Semi-Naive Bayesian networks are proposed to remedy violations of NB's assumption

O. Kaynak et al. (Eds.): ICANN/ICONIP 2003, LNCS 2714, pp. 115–122, 2003.

by joining attributes into several combined attributes based on a conditional independency assumption among the combined attributes. Some performance improvements have been demonstrated in [6, 11].

However, two major problems exist for the SNB. First, typically,the traditional SNB can only combine two attributes into a combined attribute or it will be computationally intractable [11]. This inflexibility is obviously a problem, since combining more attributes may generate better results. Second, the conditional independency assumption among the joined attributes is still too strong although it is looser than NB's. These two problems restrict the expressive ability of the SNB and therefore may model inaccurate distributions for some datasets. How to solve these two problems effectively and efficiently becomes an important issue. To handle the first problem, in this paper, we develop a Bounded-SNB model based on direct combinatorial techniques. By transforming a learning problem into an integer programming problem, this model can combine any number of attributes within a given bound and maintain a polynomial time cost at the same time.

To solve the second problem, one possible way is to search an independence or dependence relationship among the attributes rather than impose a strong assumption on the attributes. This is the main idea of so-called unrestricted Bayesian Network (BN) [12]. Unfortunately, empirical results have demonstrated that searching an unrestricted BN structure does not show a better result than NB [3, 4]. This is partly because that unrestricted BN structures are prone to incurring overfitting problems [3]. Furthermore, searching an unrestricted BN structure is generally an NP-complete problem [1]. Different from searching unrestricted structures, in this paper, we upgrade the SNB into a mixture structure, where a hidden variable is used to coordinate its components: SNB structures. Mixture approaches have achieved great success in expanding its restricted components expressive power and bringing a better performance.

In summary, in this paper, we use our B-SNB model to deal with the first problem. We then provide an algorithm to perform the mixture structure upgrading on our B-SNB model. On one hand, the B-SNB model enables the mixture a diversity, i.e., it is not necessary to limit the component structure into a SNB with combined attributes consisting of less or equal than two attributes. On the other hand, the mixture model expands the expressive ability for the B-SNB model. This paper is organized as follows. In Section 2, we describe our B-SNB model in detail. Then in Section 3, we discuss the mixture of B-SNB model and give an induction algorithm. Experimental results to show the advantages of our model are demonstrated in Section 4. Finally, we conclude this paper in Section 5.

2 Bounded Semi-naive Bayesian Network

Our Bounded Semi-Naive Bayesian network model is defined as follows:

Definition 1. *B-SNB Model : Given a set of N independent observations* $D = \{x^1, \dots, x^N\}$ *and a bound K, where* $x^i = (A_1^i, A_2^i, \dots, A_n^i)$ *is an n-*

dimension vector and $A_1, A_2, \ldots, A_n$ are called variables or attributes, B-SNB is a maximum likelihood Bayesian network which satisfies the following conditions:

1. *It is composed of m large attributes $B_1, B_2, \ldots, B_m$, $1 \leq m \leq n$, where each large attribute $B_l = \{A_{l_1}, A_{l_2}, \ldots, A_{l_{k_l}}\}$ is a subset of attribute set: $\{A_1, \ldots, A_n\}$.*
2. *There is no overlap among the large attributes and their union forms the attributes set. That is, (1) $B_i \cap B_j = \phi$, for $i \neq j$, and $1 \leq i, j \leq m$; (2) $B_1 \cup B_2 \cup \ldots \cup B_m = \{A_1, A_2, \ldots, A_n\}$.*
3. *B_i is independent of B_j, for $i \neq j$, namely, $P(B_i, B_j) = P(B_i)P(B_j)$, for $i \neq j$, and $1 \leq i, j \leq m$.*
4. *The cardinality of each large attribute $B_l (1 \leq l \leq m)$ is not greater than K. If each large attribute has the same cardinality K, we call the B-SNB K-regular B-SNB.*

Except for Item 4, the B-SNB model definition is the definition of the traditional SNB. We argue that this constraint on the cardinality is necessary. K cannot be set as a very large value, or the estimated probability for large attributes will be not reliable. When using B-SNB for classification tasks, we first partition the pre-classified dataset into some sub-datasets by the class label and then train different B-SNB structures for different classes. From this viewpoint, Item 3 is actually a conditional independence formulation, when given the class variable, since this independency is assumed in the sub-database with a uniform class label.

2.1 Learning the Optimal B-SNB from Data

In general, the optimal B-SNB estimated from a dataset D can be achieved in two steps. The first step is to learn an optimal B-SNB structure from D; the second step is to learn the optimal parameters for this optimal structure, where B-SNB parameters are those probabilities of each large attribute, i.e., $P(B_j)$. It is easy to show that the sample frequency of a large attribute B_j is the maximum-likelihood estimator for the probability $P(B_j)$, when a specific B-SNB structure is given (See the Appendix for the proof of Lemma 1). Thus the key problem in learning the optimal B-SNB is the structure learning problem, namely how to find the best m large attributes.

However the combination number for m large attributes in an n-dimension dataset will be $\sum_{\{k_1,k_2,\ldots,k_n\} \in G} C_n^{k_1} C_{n-k_1}^{k_2} \ldots C_{n-\sum_{i=1}^{n-1} k_i}^{k_n}$,$G = \{\{k_1, k_2, \ldots, k_n\} : \sum_{i=1}^{n} k_i = n, 0 \leq k_i \leq K\}$. Such a large searching space for an optimal B-SNB will make it nearly impossible to employ greedy methods especially when K is set to some small values. To solve this problem, we firstly develop the following two lemmas.

Lemma 1. *The maximum* log *likelihood of a specific B-SNB S for a dataset D, represented by l_S, can be written into the following form $l_S = -\sum_{i=1}^{m} \hat{H}(B_i)$, where $\hat{H}(B_i)$ is the entropy of large attribute B_i based on the empirical distribution of D.*

Lemma 2. *Let μ and $\mu^{'}$ be two B-SNBs over dataset D. If $\mu^{'}$ is coarser than μ, then $\mu^{'}$ provides a better approximation than μ over D.*

The *coarser* concept is defined in this way: If $\mu^{'}$ can be obtained by combining the large attributes of μ without splitting the large attribute of μ, then $\mu^{'}$ is coarser than μ.

The details of the proof of Lemma 1 and Lemma 2 can be seen in Appendix.

According to Lemma 2, within a reasonable K bound, a higher "order" approximation will be superior to a lower "order" one. For example, it is more accurate using $P(a,b,c)P(d,e,f)$ to approximate $P(a,b,c,d,e,f)$ than using $P(a,b)P(c)P(d,e,f)$ when each subitem probability can be estimated reliably. In a K-B-SNB, K is the possible highest order for any large attributes. Thus we should use as many K-large attributes as possible in constructing B-SNB. Under this consideration, we fix all the large attributes to K large-attributes. On one hand, searching K-regular B-SNBs can reduce the combination number of large attributes to $\frac{n!}{(K!)^{[n/K]}}$. On the other hand, this constraint enables us to transform the optimization into an integer programming (IP) problem easily. Further we can approximate the IP solution via linear programming techniques, which can be solved in a polynomial time cost.

2.2 Transforming into Integer Programming Problem

We first describe our B-SNB optimization problem under Maximum Likelihood Estimation criterion when the cardinality of each large attribute is constrained to be exactly bound K.

B-SNB Optimization Problem: From the attributes set, find $m = [n/K]$ K-cardinality subsets, which satisfy the B-SNB conditions, to maximize the log likelihood $l_S = -\sum_{i=1}^{m} \hat{H}(B_i)$.

We write this B-SNB optimization problem into the following IP problem:

$$\text{Min} \sum_{V_1,V_2,\dots,V_K} x_{V_1,V_2,\dots,V_K} \hat{H}(V_1,V_2,\dots,V_K), \quad \textit{where},$$

$$(\forall V_K) \sum_{V_1,V_2,\dots,V_{K-1}} x_{V_1,V_2,\dots,V_K} = 1, \quad x_{V_1,V_2,\dots,V_K} \in \{0,1\} \qquad (1)$$

Here $V_1, V_2, \dots, V_k$ represent any K attributes. Equation (1) describes that for any attribute, it can just belong to one large attribute, i.e., when it occurs in one large attribute, it must not be in another large attribute, since there is no overlapping among the large attributes.

We approximate the solution of IP via Linear Programming (LP) method, which can be solved in a polynomial time. By relaxing $x_{V_1,V_2,\dots,V_K} \in \{0,1\}$ into $0 \le x_{V_1,V_2,\dots,V_K} \le 1$, the IP problem is transformed into an LP problem. Then a rounding procedure to get the integer solution is conducted on the solution of LP. It should be addressed that direct solving for IP problem is infeasible. It is reported that IP problems with as few as 40 variables can be beyond the abilities of even the most sophisticated computers.

Approximating IP solution by LP may reduce the accuracy of the SNB while it can decrease the computational cost to a polynomial one. Furthermore, shown in our experiments, this approximation achieves a satisfactory prediction accuracy.

3 The Mixture of Bounded Semi-naive Bayesian Network

In this section, we first define the Mixture of Bounded Semi-Naive Bayesian network (MBSNB) model, then we give the optimization problem of the MBSNB model. Finally we conduct theoretical induction to provide the optimization algorithm for this problem under the EM [7] framework.

Definition 2. *Mixture of Bounded Semi-Naive Bayesian network model is defined as a distribution of the form: $Q(x) = \sum_{k=1}^{r} \lambda_k S^k(x)$, where $\lambda_k \geq 0$, $k = 1, \ldots, r$,$\sum_{k=1}^{r} \lambda_k = 1$, r is the number of components in the mixture structure. S^k represents the distribution of the kth component K Bounded Semi-Naive network. λ_k can be called component coefficient.*

Optimization Problem of MBSNB: *Given a set of N independent observations $D = \{x^1, x^2, \ldots, x^N\}$ and a bound K, find the mixture of K-Bounded-SNB model Q, which satisfies $Q = \arg\max_{Q'} \sum_{i=1}^{N} \log Q'(x^i)$.*

We use a modified derivation process as [9] to find the solution of the above optimization problem. According to the EM algorithm, finding the optimal model Q of the above is equal to maximizing the following complete log-likelihood function:

$$\begin{aligned} l_c(x^{1,\cdots,N}, z^{1,\cdots,N}|Q) &= \sum_{i=1}^{N} \log \prod_{k=1}^{r} \left(\lambda_k S^k(x^i)\right)^{\delta_{k,z^i}} \\ &= \sum_{i=1}^{N} \sum_{k=1}^{r} \delta_{k,z^i} (\log \lambda_k + \log S^k(x^i)) \end{aligned} \tag{2}$$

where z is the choice variable which can be seen as the hidden variable to determine the choice of the component Semi-Naive structure; δ_{k,z^i} is equal to 1 when z^i is equal to the kth value of choice variable and 0 otherwise. We utilize the EM algorithm to find the solution of above log-likelihood formulation. First taking the expectation with respect to z, we will obtain

$$E[l_c(x^{1,\cdots,N}, z^{1,\cdots,N}|Q)] = \sum_{i=1}^{N} \sum_{k=1}^{r} E(\delta_{k,z^i}|D)(\log \lambda_k + \log S^k(x^i)), \tag{3}$$

where $E(\delta_{k,z^i}|D)$ is actually the posterior probability given the ith observation, which can be calculated as: $E(\delta_{k,z^i}|D) = P(z^i|V = x^i) = \frac{\lambda_k S^k(x^i)}{\sum_{k'} \lambda_{k'} S^{k'}(x^i)}$. We define $\gamma_k(i) = E(\delta_{k,z^i}|D)$, $\mathit{\Gamma}_k = \sum_{i=1}^{N} \gamma_k(i)$, $P^k(x^i) = \frac{\gamma_k(i)}{\mathit{\Gamma}_k}$. Thus we obtain the expectation:

$$E[l_c(x^{1,\cdots,N}, z^{1,\cdots,N}|Q)] = \sum_{k=1}^{r} \Gamma_k \log \lambda_k + \sum_{k=1}^{r} \Gamma_k \sum_{i=1}^{N} P^k(x^i) \log S^k(x^i). \quad (4)$$

Then we perform the Maximization step in Equation (4) with respect to the parameters. It is easy to maximize the first part of Equation (4) by Lagrange method with the constraint $\sum_{k=1}^{r} \lambda_k = 1$. We can obtain: $\lambda_k = \frac{\Gamma_k}{N}$, $k = 1, \ldots, r$.

Table 1. Description of data sets used in the experiments

Dataset	♯Variables	♯Class	♯Train	♯Test
Xor	6	2	2000	CV-5
Vote	15	2	435	CV-5
Tic-tac-toe	9	2	958	CV-5
Segment	19	7	2310	30%

If we consider $P^k(x^i)$ as the probability for each observation over the kth component B-SNB, the latter part of Equation (4) is in fact a B-SNB network optimization problem, which can be solved by our earlier proposed algorithm in Section 2.

4 Experiments

To evaluate the performance of our B-SNB and MBSNB models, we conduct a series of experiments on four databases, among which three come from the UCI Machine learning Repository [10] and the other one dataset called Xor is generated synthetically. In Xor, the class variable C is the result of xor operation between the first two binary attributes and other four binary attributes are created randomly. Table 1 is the detailed information for these four datasets. We use the 5-fold Cross Validation (CV) method [5] to perform testing on these datasets. We train a MBSNB model Q_{C_i} for each class C_i of every dataset. And we use the Bayes formula: $c(x) = \arg\max_{C_i} P(C_i)Q_{C_i}(x)$ to classify a new instance x. We compare B-SNB, MNSNB models with NB, Chow-Liu tree (CLT) algorithm and C4.5 (CLT is a kind of competitive Bayesian classifier [2]). We set the bound K for B-SNB and MBSNB as 2 and 3 to examine their performances. Table 2 summarizes the prediction results of the main approaches in this paper. 2(3)-B-SNB and 2(3)-MBSNB means K is set as 2(3). It is observed that B-SNB can improve the NB's performance. Moreover B-SNB performance can be further improved with the mixture upgrading. Since,B-SNB can be considered as the special case of MBSNB, with mixture number equal to 1. We take the highest accuracy rate as the one of MBSNB from the 2(3)-B-SNB and 2(3)-MBSNB. This result is shown as MBSNB* in the last column of Table 2. We can observe that this column almost demonstrates the best overall performance in comparison with NB, CLT and C4.5.

Table 2. Prediction Accuracy of the Primary Approaches in this paper(%)

Dataset	NB	CLT	C4.5	2-B-SNB	3-B-SNB	2-MBSNB	3-MBSNB	MBSNB*
Xor	54.50	**100**	**100**	100	99.50	99.50	99.50	**100**
Tic-tac-toe	70.77	73.17	84.84	72.65	78.39	88.33	79.38	**88.33**
Vote	90.11	91.26	**94.18**	92.40	92.64	93.10	94.00	94.00
Segment	88.29	91.33	90.61	91.90	89.16	91.47	90.90	**91.90**

5 Conclusion

In this paper, we propose a Bounded Semi-Naive Bayesian network based on direct combinatorial optimization. Different with the traditional SNBs, this model can combine any number of attributes within a given bound and maintain a polynomial time cost at the same time. Furthermore, we upgrade it into a finite mixture model. We designed a serious of experiments to demonstrate our model's advantages. The results show that this mixture model brings in an increase in prediction accuracy.

Acknowledgement. The work described in this paper was fully supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4351/02E and Project No. CUHK4360/02E).

References

1. D. M. Chickering. Learning bayesian networks is NP-complete. In D. Fisher and H.-J. Lenz, editors, *Learning from Data*. Springer-Verlag, 1995.
2. C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Information Theory*, 14:462–467, 1968.
3. Pedro Domingos and Pazzani Michael. On the optimality of the simple baysian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.
4. N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–161, 1997.
5. R. Kohavi. A study of cross validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th IJCAI*, pages 338–345. San Francisco, CA:Morgan Kaufmann, 1995.
6. I. Kononenko. Semi-naive bayesian classifier. In *Proceedings of sixth European Working Session on Learning*, pages 206–219. Springer-Verlag, 1991.
7. N. M. Laird, A. P. Dempster, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Society*, B39:1–38, 1977.
8. P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. *In Proceedings of AAAI-92*, pages 223–228, 1992.
9. M. Meila and M. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, 2000.
10. Patrick M. Murphy. UCI repository of machine learning databases. In *School of Information and Computer Science, University of California, Irvine*, 2003.

11. M. J. Pazzani. Searching dependency in bayesian classifiers. In D. Fisher and H.-J. Lenz, editors, *Learning from data: Artificial intelligence and statistics V*, pages 239–248. New York, NY:Springer-Verlag, 1996.
12. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: networks of plausible inference.* Morgan Kaufmann, CA, 1988.
13. J. R. Quinlan. *C4.5 : programs for machine learning.* San Mateo, California: Morgan Kaufmann Publishers, 1993.

6 Appendix

Proof for Lemma 1:
Let S is a specific B-SNB with n variables or attributes which are represented respectively by A_i, $1 \leq i \leq n$. And this B-SNB's large attributes are represented by B_i, $1 \leq i \leq m$. We use $(B_1, \ldots, B_m)$ as the short form of $(B_1, B_2, \ldots, B_m)$. The log likelihood over a data set can be written into the following:

$$l_S(x^1, x^2, \ldots, x^s) = \sum_{j=1}^{s} \log P(x^j)$$

$$= \sum_{j=1}^{s} \log(\prod_{i=1}^{m} P(B_i)) = \sum_{i=1}^{m} \sum_{j=1}^{s} \log P(B_i) = \sum_{i=1}^{m} \sum_{B_i} \hat{P}(B_i) \log P(B_i)$$

The above term will be maximized when $P(B_i)$ is estimated by $\hat{P}(B_i)$, the empirical probability for large attribute B_i. This can be easily obtained by maximizing l_S with respect to $P(B_i)$. Thus,

$$l_{S\,max} = \sum_{i=1}^{m} \sum_{B_i} \hat{P}(B_i) \log \hat{P}(B_i) = -\sum_{i=1}^{m} \hat{H}(B_i)$$

Proof for Lemma 2:
We just consider a simple case, the proof for the general case is much similar. Consider one partition as $\mu = (B_1, B_2, \ldots, B_m)$ and another partition as

$$\mu_1 = (B_1, B_2, \ldots, B_{m-1}, B_{m1}, B_{m2}), \quad \text{where}$$
$$B_{m1} \cap B_{m2} = \phi \quad \text{and} \quad B_{m1} \cup B_{m2} = B_m$$

According to the proof of Lemma 1 above, we have:

$$l_{S_\mu\,max} = \sum_{i=1}^{m} \hat{H}(B_i) = -\sum_{i=1}^{m-1} \hat{H}(B_i) - \hat{H}(B_m) \tag{5}$$

According to Entropy theory, $\hat{H}(XY) \leq \hat{H}(X) + \hat{H}(Y)$. We can write Eq. (5) into:

$$l_{S_\mu\,max} = -\sum_{i=1}^{m-1} \hat{H}(B_i) - \hat{H}(B_m) \geq -\sum_{i=1}^{m-1} \hat{H}(B_i) - \hat{H}(B_{m1}) - \hat{H}(B_{m2})$$
$$= l_{S_{\mu_1}\,max} \tag{6}$$