A Hybrid Generative/Discriminative Bayesian Classifier

Changsung Kang and **Jin Tian**
Department of Computer Science
Iowa State University
Ames, IA 50011
{cskang,jtian}@iastate.edu

Abstract

In this paper, we introduce a new restricted Bayesian network classifier that extends naive Bayes by relaxing the conditional independence assumptions, and show that it is partly generative and partly discriminative. Experimental results show that the hybrid classifier performs better than a purely generative classifier (naive Bayes) or a purely discriminative classifier (Logistic Regression) and has performance comparable to some state-of-the-art classifiers.

Introduction

In classification problems, we need to build a classifier that assigns a class label C to instances described by a set of attributes $A_1, \cdots, A_n$. There are two approaches to the classification problems: generative or discriminative. In the generative classification, we model the joint probability $P(A_1, \cdots, A_n, C)$ (or the class-conditional probability $P(A_1, \cdots, A_n|C)$ and the prior $P(C)$), and calculate the posterior $P(C|A_1, \cdots, A_n)$, often by Bayes rule, and then pick the class label for an instance for which $P(C|A_1, \cdots, A_n)$ is maximum. In contrast, discriminative classifiers estimate the posterior $P(C|A_1, \cdots, A_n)$ directly, or directly learn a function from the attributes $A_1, \cdots, A_n$ to the class label C. Examples of generative classifiers are Fisher Discriminant Analysis, Hidden Markov Models, and naive Bayes. Examples of discriminative classifiers include Logistic Regression, Neural Networks, and Generalized Additive Models (Rubinstein & Hastie 1997).

Although discriminative classifiers are often preferred to generative ones, a simple generative classifier, naive Bayes, is shown to perform surprisingly well in many domains. naive Bayes assumes that all the attributes $A_1, \cdots, A_n$ are conditionally independent given the value of the class C. This assumption rarely holds in practice. There have been many work on improving the performance of naive Bayes by relaxing the conditional independence assumptions. For example, (Kononenko 1991) proposes an extension to naive Bayes, called *semi-naive Bayesian classifier*, in which attributes are partitioned into groups and it is assumed that A_i is conditionally independent of A_j if and only if they are in different groups. Similarly, (Ezawa & Schuermann 1995; Pazzani 1996) attempt to improve the classification accuracy by removing some of the conditional independence assumptions. Also, (Kohavi & John 1997; Langley & Sage 1994) attempt to improve naive Bayes by rendering some of the attributes irrelevant. (Friedman, Geiger, & Goldszmidt 1997) introduces the use of Bayesian networks as classifiers. Bayesian networks are powerful tools for representing joint distributions and encoding conditional independence assumptions, and have naive Bayes as a special case. (Friedman, Geiger, & Goldszmidt 1997) show, by experiments, that unrestricted Bayesian networks may not perform better than naive Bayes, and introduced a restricted form of Bayesian networks as a classifier, called *Tree Augmented Naive Bayes* (TAN), which augments naive Bayes by allowing each attribute to depend on at most one other attribute. Their experimental results show that TAN performs better than naive Bayes or unrestricted Bayesian network classifier, while maintaining computational simplicity. Recently there have been many interests in restricted or unrestricted Bayesian network classifiers (Singh & Provan 1996; Keogh & Pazzani 1999; Sahami 1996).

In this paper, we present a new restricted Bayesian network classifier which relaxes some of the conditional independence assumptions in naive Bayes. The set of attributes are partitioned into two sets S_1 and S_2. And we show that the classifier can be viewed as a hybrid generative/discriminative classifier that can be learned by discriminatively learning $P(C|S_1)$ and generatively learning $P(S_2|C)$. In related work, (Rubinstein & Hastie 1997) introduces the ideas of combining generative and discriminative learning. (Ng & Jordan 2002) compares a generative classifier naive Bayes with a discriminative classifier Logistic Regression. (R. Raina & McCallum 2004) introduces a hybrid model that is partly generative and partly discriminative for an application in text classification and shows that the hybrid model outperforms both naive Bayes and Logistic Regression.

In Section 2, we present our hybrid model and give a learning algorithm. In Section 3, we test two versions of our hybrid classifier on data sets and compare them with naive Bayes, Logistic Regression and some state-of-the-art classifiers. Section 4 concludes the paper.

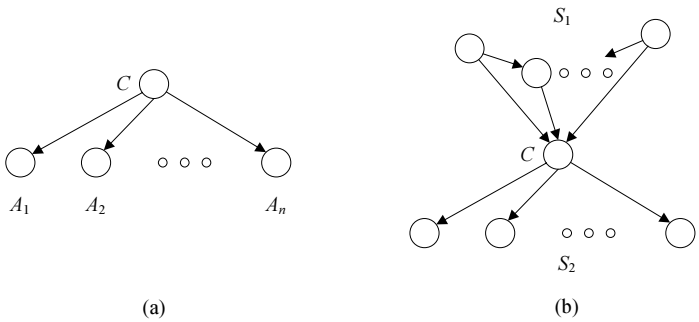

Figure 1: (a) The structure of naive Bayes; (b) Our proposed Bayesian network classifier.

A Hybrid Generative/Discriminative Bayesian Classifier

The structure of naive Bayes as a Bayesian network is shown in Figure 1(a). The assumption in naive Bayes that all the attributes $A_1, \cdots, A_n$ are conditionally independent given the value of the class C induces the following class posterior.

$$P(C|A_1, \cdots, A_n) = \alpha P(C) \prod_i P(A_i|C) \qquad (1)$$

where α is a normalization constant.

Relaxing the strong conditional independence assumption of naive Bayes, we propose a Bayesian network classifier with a structure in the form shown in Figure 1(b), in which the set of attributes $\{A_1, \cdots, A_n\}$ are partitioned into two sets S_1 and S_2; each attribute in S_2 has the class C as the only parent; each attribute in S_1 is a parent of C and there can be unspecified edges among attributes in S_1. The Bayesian network structure in Figure 1(b) assumes the following conditional independence relations:

(i) All the attributes in S_2 are conditionally independent given the value of the class C,

(ii) The set of attributes S_1 are conditionally independent of the set of attributes S_2 given the value of the class C.

All these conditional independence assumptions are assumed in the naive Bayes classifier. And we have eliminated the assumption that the attributes in S_1 are conditionally independent given the value of the class C.

Assuming the Bayesian network structure in Figure 1(b), the posterior of the class label C can be computed as, using conditional Bayes rule,

$$\begin{aligned} P(C|A_1, \cdots, A_n) &= \alpha P(C|S_1) P(S_2|C) \\ &= \alpha P(C|S_1) \prod_{A_i \in S_2} P(A_i|C) \end{aligned} \qquad (2)$$

where $\alpha = 1/P(S_2|S_1)$. To use this model as a Bayesian classifier, from (2), the parameters we need to learn are $P(C|S_1)$ and $P(A_i|C)$ for $A_i \in S_2$, and one key observation is that the probability $P(S_1)$ is not needed for the classification purpose. But first, we need to decide how to partition the set of attributes into S_1 and S_2. One approach is to learn a Bayesian network restricted to the form of Figure 1(b). The problem with this approach is that it is computationally costly and that we have to learn a structure over S_1 which is not needed for the classification purpose and the best structure learned may not be the best one for classification as argued in (Friedman, Geiger, & Goldszmidt 1997). In this paper, we partition the attributes directly based on the training-data classification accuracy. We start with the naive Bayes structure, then add variables to S_1 in a greedy way, that is, add a single attribute to S_1 each time until the classification accuracy is not improved.

In standard Bayesian networks, conditional probability distributions (CPDs) are normally represented in a tabular form. And the parameters $P(C|S_1)$ and $P(A_i|C)$ can be estimated using the maximum likelihood estimation. The problem is that when the number of attributes in S_1 is large, the estimation for $P(C|S_1)$ is not reliable. One way to tackle this problem is to use some *local structures* to encode the CPDs that reduce the number of parameters. For example, CPDs can be represented by Logistic Regression, noisy-or, decision trees, and neural networks (Friedman & Goldszmidt 1996). In fact, we can think this as a discriminative learning problem that learns the class posterior $P(C|S_1)$ with S_1 as the set of attributes. In this sense, our proposed model is a hybrid generative/discriminative classifier that partitions the attributes into two sets S_1 and S_2, then learns $P(C|S_1)$ discriminatively and $P(S_2|C)$ generatively, and combines them naturally by Bayes rule as shown in Eq. (2). We can use any discriminative learning method that can output the posterior $P(C|S_1)$ (for example, Logistic Regression), we can use any generative learning model to estimate $P(S_2|C)$ (for example, TAN), and we can decide the partition of attributes based on the classification accuracy. In this paper, we use Logistic Regression to learn the CPD $P(C|S_1)$, and use naive Bayes and TAN to estimate $P(S_2|C)$. Note that when using TAN to estimate $P(S_2|C)$, we assume a Bayesian network structure in Figure 1(b) with some additional edges among S_2 and the posterior of the class label C is computed as

$$P(C|A_1, \cdots, A_n) = \alpha P(C|S_1) \prod_{A_i \in S_2} P(A_i|PA_i) \qquad (3)$$

where $\alpha = 1/P(S_2|S_1)$ and PA_i are the parents of A_i.

Next we give the details of our algorithm.

Our Algorithm

We now introduce HBayes, a hybrid generative/discriminative Bayesian classifier. We consider two versions of HBayes. Both of them learn the CPD $P(C|S_1)$ using Logistic Regression with a ridge estimator (le Cessie & van Houwelingen 1992) which is implemented in Weka (Witten & Frank 1999). The first, HBayes-NB, uses the naive Bayes model to estimate $P(S_2|C)$. HBayes-NB chooses S_1 based on the classification accuracy over the training data. Since searching the space of all possible subsets is computationally hard, we use a greedy method to select S_1 starting with S_1 being empty. We start by examining every pair of attributes since Eq. (2) will reduce to the naive Bayes formula (1) when S_1 has only one attribute. Given training data, we add the pair of attributes to S_1 that improve the accuracy of the classifier the most, or we stop and output S_1 as empty if none of the pair improves the accuracy. Subsequently, we test every single attribute to find an attribute that maximally improves the classification accuracy when it is added to S_1. We keep adding attributes to S_1 until the classification accuracy cannot increase. HBayes-NB is presented in Figure 2.

The parameters $P(A_i|C)$ for $A_i \in S_2$ are easily estimated, based on the frequencies over the training data. We use the Laplace correction (Niblett 1987) to prevent the harmful effects of zero probabilities. The Laplace corrected estimate of $P(A_i = a|C = c)$ is

$$P(A_i = a|C = c) = \frac{n_{ca} + 1/N}{n_c + n_i/N} \quad (4)$$

where n_{ca} is the number of times class c and value a of attribute A_i occur together, n_c is the number of times class c occurs, n_i is the number of values of attribute A_i, and N is the number of examples in the training set.

The second version, HBayes-TAN, uses TAN to estimate $P(S_2|C)$. In this network structure, Eq. (3) will not be identical for every S_1 with single attribute. This enables us to do the greedy search much faster than HBayes-NB. HBayes-TAN starts by considering every single attribute to add to the initially empty set S_1, not every pair of attributes as in HBayes-NB.

Another key aspect of the proposed classifier is that it learns the network structure by maximizing the classification accuracy while setting some parameters by maximizing conditional likelihood and others by maximum likelihood. Note that the parameters $P(C|S_1)$ are set to maximum conditional likelihood by Logistic Regression and the parameters $P(A_i|C)$ for $A_i \in S_2$ to maximum likelihood. Our approach contrasts with the one proposed by (Grossman & Domingos 2004) which chooses structures by maximizing conditional likelihood while setting parameters by maximum likelihood. If we learned an unrestricted Bayesian network, our method would be computationally too costly. However, the restriction on the structure given in Figure 1(b) greatly reduces the computational effort.

procedure HBayes-NB(D)
INPUT: training instances D
OUTPUT: Bayesian network B
  Let A_i, A_j be the pair of attributes that maximize **Accuracy**($D, \{A_i, A_j\}$);
  $S_1 = \{A_i, A_j\}$;
  $maxAccuracy =$ **Accuracy**($D, \{A_i, A_j\}$);
  if $maxAccuracy <$ **Accuracy**($D, \emptyset$) **then**
    $S_1 = \emptyset$;
  else
    repeat
      Let A_k be the attribute that maximizes **Accuracy**($D, S_1 \cup \{A_k\}$);
      if $maxAccuracy <$ **Accuracy**($D, S_1 \cup \{A_k\}$) **then**
        $S_1 = S_1 \cup A_k$;
        $maxAccuracy =$ **Accuracy**($D, S_1 \cup \{A_k\}$);
      else
        exit loop;
  Let B be a Bayesian network in which the class C has parents S_1 and C is the only parent of all the other attributes S_2. Estimate the parameters of B on D. Use Logistic Regression and naive Bayes to estimate $P(C|S_1)$ and $P(S_2|C)$, respectively.;
  return B

procedure Accuracy(D, S_1)
INPUT: training instances D, parent set S_1
OUTPUT: accuracy
  Let B be a Bayesian network in which the class C has parents S_1 and C is the only parent of all the other attributes S_2. Estimate the parameters of B on D. Use Logistic Regression and naive Bayes to estimate $P(C|S_1)$ and $P(S_2|C)$, respectively.;
  return the classification accuracy of B on D

Figure 2: Our proposed algorithm: HBayes-NB

Computational Complexity

Since Logistic Regression dominates other computations in the procedure **Accuracy**, HBayes-NB has time complexity $O(m^2l)$ where m is the number of attributes and l is the time it takes for Logistic Regression to estimate $P(C|S_1)$. l depends on implementation. The exact time complexity of Logistic Regression used in our algorithm is not available. HBayes-TAN has time complexity $O(ml)$.

Experimental Results

We compared our algorithms HBayes-NB and HBayes-TAN with naive Bayes (NB), Logistic Regression (LR), TAN, and C4.5(J48) (Quinlan 1993). All of them are implemented in Weka (Witten & Frank 1999).

We ran our experiments on 20 data sets from the UCI repository (Newman *et al.* 1998). We carried out preprocessing stages for numeric attributes and missing values. Numeric attributes were discretized into ten equal-length intervals. Each occurrence of a missing value was replaced by the most frequently occurring value in that attribute. These preprocessing stages were carried out by the filters in Weka.

Classification accuracies were measured via 10-fold cross validation for the smaller data sets, and via 3-fold cross validation for the larger data sets (Chess, Hypothyroid, Segment and Wave).

Table 1 shows the classification errors of each algorithm on the data sets. Figures 3 and 4 compare HBayes-NB and HBayes-TAN with the competing algorithms. Points above the $y = x$ diagonal are data sets for which our algorithms outperform the competing algorithms. Also, the figures show one-standard-deviation bars. In our experiments, HBayes-NB outperformed naive Bayes and Logistic Regression in the Wilcoxon paired-sample signed-ranks test. HBayes-NB provided a modest, but not statistically significant improvement against TAN and C4.5. The significance values appear in the figures. While HBayes-TAN is much faster than HBayes-NB, it produced slightly worse classification accuracy.

We also tested a variant of HBayes-TAN, which starts by examining every pair of attributes as in HBayes-NB. Interestingly, we found that it did not produce better results than those obtained by HBayes-NB.

Conclusion

This paper introduces a new restricted Bayesian network classifier that relaxes some conditional independence assumptions made by naive Bayes. We show that the proposed classifier can be seen as a hybrid generative/discriminative classifier. We present an algorithm that learns the classifier by combining naive Bayes and Logistic Regression in a greedy fashion, and show that HBayes-NB, the resulting classifier, outperforms a purely generative classifier (naive Bayes) and a purely discriminative classifier (Logistic Regression) and has performance comparable to other classifiers such as C4.5 and TAN. We also propose another version of the restricted Bayesian network classifier, HBayes-TAN, that combines TAN and Logistic Regression. HBayes-TAN achieves slightly worse classification accuracy than HBayes-NB, but is computationally more efficient.

References

Ezawa, K., and Schuermann, T. 1995. Fraud/uncollectible debt detection using a bayesian network learning system. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 157–166. Morgan Kaufmann.

Friedman, N., and Goldszmidt, M. 1996. Learning bayesian networks with local structure. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence*, 252–262. Morgan Kaufmann.

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

Grossman, D., and Domingos, P. 2004. Learning bayesian network classifiers by maximizing conditional likelihood. In *Proceedings of the twenty-first international conference on Machine learning*, 361–368. ACM Press.

Keogh, E., and Pazzani, M. 1999. Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, 225–230.

Kohavi, R., and John, G. 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97(1-2):273–324.

Kononenko, I. 1991. Semi-naive bayesian classifier. In *Proceedings of sixth European Working Session on Learning*, 206–219. Springer-Verlag.

Langley, P., and Sage, S. 1994. Induction of selective bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, 399–406. Morgan Kaufmann.

le Cessie, S., and van Houwelingen, J. 1992. Ridge estimators in logistic regression. *Applied Statistics* 41(1):191–201.

Newman, D.; Hettich, S.; Blake, C.; and Merz, C. 1998. UCI repository of machine learning databases [http://www.ics.uci.edu/∼mlearn/mlrepository.html]. University of California, Irvine, Dept. of Information and Computer Sciences.

Ng, A., and Jordan, M. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems* 14.

Niblett, T. 1987. Constructing decision trees in noisy domains. In *Proceedings of the Second European Working Session on Learning*, 67–78. Bled, Yugoslavia: Sigma.

Pazzani, M. 1996. Searching for dependencies in bayesian classifiers. In *Proceedings of the Fifth InternationalWorkshop on Artificial Intelligence and Statistics*, 239–248. Springer-Verlag.

Quinlan, J. 1993. *C4.5: programs for machine learning*. Morgan Kaufmann.

Table 1: Classification error.

Dataset	HBayes-NB	HBayes-TAN	NB	LR	C4.5	TAN
Annealing	.1116	.0827	.1267	.0914	.1040	.0952
Balance-scale	.0273	.0354	.0800	.0145	.3452	.1455
Chess	.0291	.0329	.1208	.0260	.0069	.0805
Credit	.1580	.1553	.1538	.1581	.1481	.1523
Glass	.3803	.3749	.3933	.4614	.4198	.3951
Hepatitis	.1886	.2089	.1420	.2587	.1674	.1878
Hypothyroid	.0276	.0269	.0323	.0288	.0295	.0279
Ionosphere	.0884	.0629	.0941	.1366	.1335	.0800
Iris	.0667	.0667	.0534	.0734	.0400	.0667
Labor	.0500	.0834	.0167	.0500	.1600	.0867
Liver disease	.3216	.3390	.3593	.3276	.4032	.3448
Lymphography	.1767	.2032	.1606	.1954	.2336	.1430
Post-operative	.3146	.3278	.3237	.3903	.2928	.3278
Segment	.0650	.0589	.0936	.0810	.0784	.0563
Soybean	.0574	.0613	.0786	.0659	.0765	.0526
Vehicle	.3125	.2650	.3847	.3476	.2934	.2660
Voting records	.0577	.0483	.0968	.0462	.0367	.0552
Wave	.1556	.1563	.1918	.1539	.2690	.1798
Wine	.0393	.0904	.0334	.0442	.1461	.0904
Zoo	.0177	.0555	.0177	.0289	.0177	.0461
Average	.1323	.1368	.1477	.1490	.1701	.1440

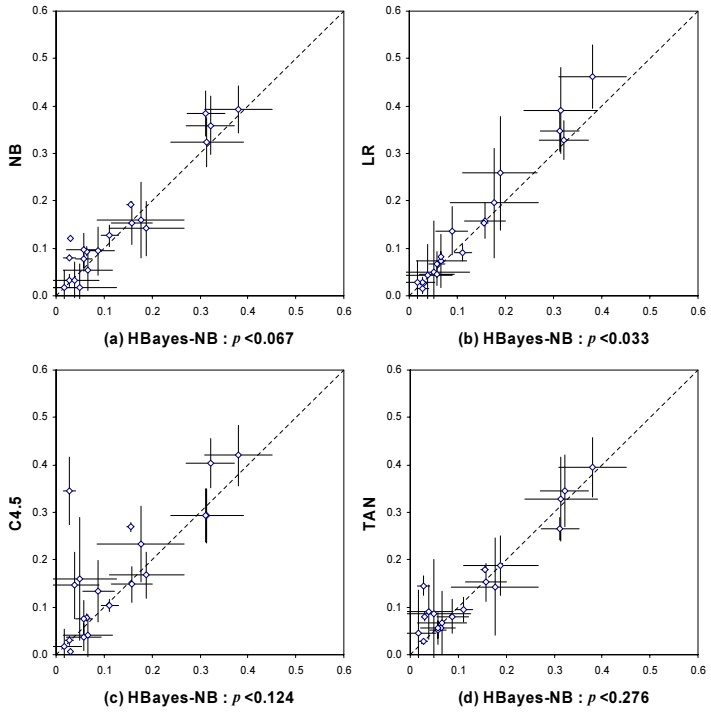

Figure 3: HBayes-NB vs. competing algorithms: classification error.

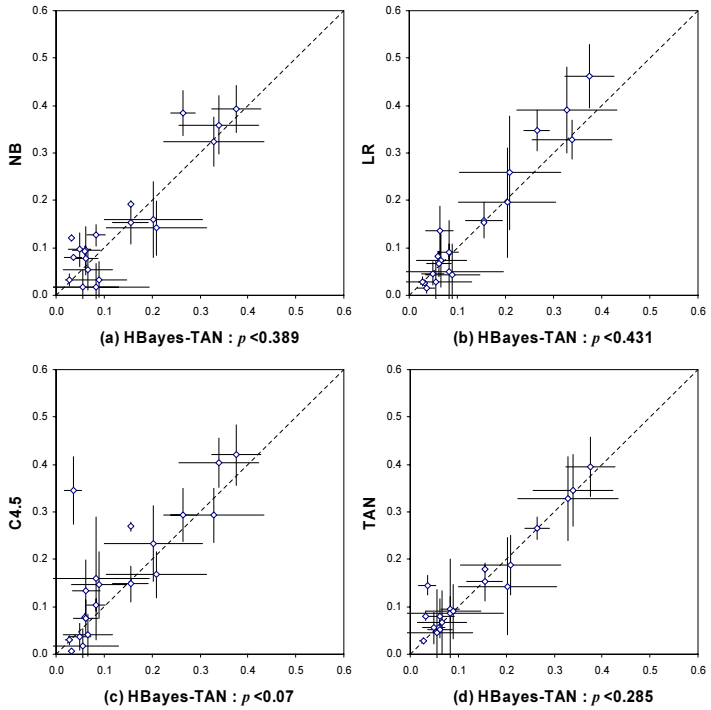

Figure 4: HBayes-TAN vs. competing algorithms: classification error.

R. Raina, Y. Shen, A. Y. N., and McCallum, A. 2004. Classification with hybrid generative/discriminative models. *Advances in Neural Information Processing Systems* 16.

Rubinstein, Y., and Hastie, T. 1997. Discriminative vs. informative learning. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, 49–53.

Sahami, M. 1996. Learning limited dependence bayesian classifiers. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 334–338. AAAI Press.

Singh, M., and Provan, G. M. 1996. Efficient learning of selective bayesian network classifiers. In *Proceedings of the Thirteenth International Conference on Machine Learning*, 453–461. Morgan Kaufmann.

Witten, I., and Frank, E. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco, CA: Morgan Kaufmann.