# A novel Naive Bayes model: Packaged Hidden Naive Bayes

Yaguang Ji
School of Computer Engineering & Science, Shanghai University
Yanchang Rd. 149, 200072 Shanghai, China
{Jiyaguang46@yahoo.com.cn}

Songnian Yu
School of Computer Engineering & Science, Shanghai University
Yanchang Rd. 149, 200072 Shanghai, China
{ snyu@staff.shu.edu.cn}

Yafeng Zhang
School of Computer Engineering & Science, Shanghai University
Yanchang Rd. 149, 200072 Shanghai, China
{zhangyafeng6362@gmail.com}

***Abstract*—Naive Bayes classifier has good performance on many datasets, however, the performance is very poor on some datasets which have a strong correlation between attributes due to the conditional independence assumption is not always true in the real world. In the latest Hidden Naive Bayes (HNB) algorithm, each attribute corresponds to a hidden parent which combines the influences of all other attributes. Compared to other Bayesian algorithms, its performance is significantly improved, but too much test time on high-dimensional datasets cost. In this paper, to find the optimal combination between Naive Bayes and HNB, a novel model Packaged Hidden Naive Bayes (PHNB), which the number of attributes in the hidden parent is controlled through packaging idea, is proposed. Our experiments show that compared to HNB, PHNB significantly reduces the test time on many high-dimensional datasets, and has higher accuracy on some particular datasets. (Abstract)**

***Keywords*：-Naive Bayes; HNB; classification; test time**

## I. INTRODUCTION

Bayesian classification algorithm, which based on perfect Bayesian theory, is one classical statistical algorithm of many classification algorithms. Bayesian classifier is constructed from a training dataset with class labels. Assuming n attributes $A_1, A_2, ..., A_n$, an instance $E$ is represented by a vector $<a_1, a_2, ..., a_n>$, where $a_i$ is the value of $A_i$, C is used to represent the class variable, $c$ is the value of $C$, and $c(E)$ denotes which the class label E belonging to. So Bayesian classifier is defined in (1) [1]:

$$c(E) = \arg\max_{c \in C} P(c)P(a_1, a_2, ..., a_n \mid c) \tag{1}$$

Assuming that all attributes are independent given the class label, thus:

$$P(E \mid c) = P(a_1, a_2, ..., a_n \mid c) = \prod_{i=1}^{n} P(a_i \mid c) \tag{2}$$

So the resulting classifier we get is called Naive Bayesian classifier [2], or simply Naive Bayes(NB) :

$$c(E) = \arg\max_{c \in C} P(c)\prod_{i=1}^{n} P(a_i \mid c) \tag{3}$$

NB has many advantages, such as simplicity, efficient computation and very good performance on many domains. But when the assumption is violated, the performance become very bad due to the assumption is not always satisfied in the real world. In recent 20 years, relaxing the assumption was paid great attention to and many new models are proposed. Related works can be broadly divided into four approaches: Feature selection, Structure extension, Local learning, Data Expansion [1].

In structure extension aspect, Bayesian network is proposed. Its node represents attribute and the arcs correspond to attribute dependencies. So learning the structure of Bayesian network is unavoidable. However, learning the optimal structure is a NP-hard problem [3]. Later, many researches are developed based on adding some restrictions to Bayesian network. TAN [4] is an extended tree-like Naive Bayes in which each class node points all attribute nodes and each attribute node except for a class node as its parent node, still has at most one other attribute node as its parent node. The performance of TAN is better than Naive Bayes, however, to TAN, the structure must be learned in training the classifier. ODANA [5] avoids the process of learning the structure, in which each attribute selects an attribute from other attributes as its parent attribute that has the maximum conditional mutual information with it. AODE [6] is consisted of many sub models in which each attribute only has a correlative attribute. The result is the average of the predictions of all sub models. WAODE [7] is an improved model of AODE, assigning weight to the sub model in terms of the dependence between the attribute and class.

HNB [8] is the latest Naive Bayes model, in which each attribute has a hidden parent which combines the influences of other attributes. It is different from Naive Bayes, the hidden parent is represented by $a_{hpi}$. So HNB is defined:

$$c(E) = \arg\max_{c \in C} P(c)\prod_{i=1}^{n} P(a_i \mid a_{hpi}, c) \tag{4}$$

The weight between attributes $A_i$ and $A_j$ is represented by $W_{ij}$, it is defined in (5). The conditional mutual information between attributes is defined in (6). Then the conditional probability of the attribute and its hidden parent given class is defined in (7) [8].

**978-1-4244-8625-0/11/$26.00 ©2011 IEEE**

$$W_{ij} = \frac{Ip(A_i; A_j \mid C)}{\sum_{j=1, j\neq i}^{n} Ip(A_i; A_j \mid C)} \quad (5)$$

$$Ip(A_i; A_j \mid C) = \sum_{a_i, a_j, c} P(a_i, a_j, c) \log \frac{P(a_i, a_j \mid c)}{P(a_i \mid c) P(a_j \mid c)} \quad (6)$$

$$P(a_i \mid a_{hpi}, c) = \sum_{j=1, j\neq i}^{n} W_{ij} \times P(a_i \mid a_j, c) \quad (7)$$

Compared to other state-of-the-art models for augmenting Naive Bayes, HNB has a better overall performance [8]. However, when it is applied on high-dimensional datasets, it consumes too much test time due to all attributes is concerned in the hidden parent. In fact, the number of correlative attributes with a specific attribute is further lower than the number of all attributes, especially when the dataset dimension is high. Our motivation is to develop a novel algorithm based on HNB to reduce the test time and still has high classification accuracy. So Packaged Hidden Naive Bayes (PHNB), which controls the number of correlative attributes to find the optimal combination of HNB model and Naive Bayes model, is presented.

The rest of this paper is organized as follows: In section 2, we present PHNB and analyze its performance in time cost. Section 3 is the experiment setup and result in detail. Finally, we conclude this paper and propose the future work.

## II. PACKAGED HIDDEN NAIVE BAYES: PHNB

### A. Algorithm Idea

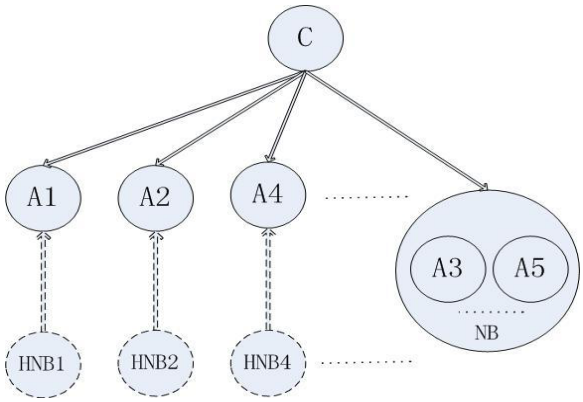


Fig 1. The structure of PHNB

As discussed in previous sections, to HNB, all attributes are concerned in the hidden parent, so the efficiency is not so good on high-dimensional datasets. In this paper, a novel Bayes model PHNB, which can reduce the test time and has a comparative accuracy to HNB model, is proposed.

The structure of PHNB is shown in Fig 1, where C is the class node which points all attribute nodes. Each attribute either has a bag HNB, represented by a dashed circle, as its hidden parent node or be contained in the bag NB. At classification phase, the attributes which have bag HNB use HNB algorithm, and other attributes in the bag NB correspond to NB algorithm. The arc from the hidden parent node to attribute node is represented by a dashed arc to distinguish it from the regular arcs.

Now we state how to get PHNB model as Fig 1. At training phase, each attribute selects the attributes whose dependences with it are greater than the threshold to put into its corresponding bag HNBi (the bag HNBi is hidden parent). However, if there is none of attributes selected, the attribute is put into the bag NB. The process is called package phase.

There are two approaches to get the threshold: computing the average of dependences of all pairs of attributes, or empirical value. On high-dimensional datasets, the threshold can be set a bigger value, so there are many attributes in the bag NB. The more the attributes in the bag NB, the less the classification time is cost. When the threshold is big enough, all attributes are contained in the bag NB, PHNB evolves into NB. On the contrary, when the dataset dimension is low, the threshold should be set a smaller value, and then there are more attributes having its corresponding HNB bag and more attributes contained in the bag HNB. So the accuracy is high. Though the classification time costs a little longer, it works still very well because the dataset dimension is low. When the threshold is small enough (such as a negative value), this model evolves into HNB. So if the threshold is set an optimal value, the model would be a perfect combination of HNB and NB, which would have a high accuracy and short classification time.

In addition, we theoretically analyze the accuracy of PHNB model. The real correlative attributes of a specific attribute are just some attributes, not all. If all attributes are concerned, many accidentals may be involved, thus when dataset dimension is very high, the accuracy may be lower. So we predict that if just the most correlative attributes are concerned, it is not only reducing the classification time but also improving the accuracy. Our experiments show that the prediction is right on some high-dimensional datasets.

The classifier corresponding to PHNB on an instance E is defined as follows:

$$c(E) = \arg\max_{c \in C} P(c) \prod_{i=1}^{numberof\ HNBbags} P(a_i \mid a_{hpi}, c) \prod_{j=1}^{numberof\ attributesinNB} P(a_j \mid c) \quad (8)$$

The conditional probability of the attribute and its hidden parent is defined in (9), which is similar to HNB, but $a_{hpi}$ just contains the dependences of the attributes in bag $HNB_i$. $W_{ij}$ is defined in (10)

$$P(a_i \mid a_{phi}, c) = \sum_{j=1, j\neq i}^{numberof\ attributesin\ HNBi} W_{ij} \times P(a_i \mid a_j, c) \quad (9)$$

$$W_{ij} = \frac{Ip(A_i; A_j \mid C)}{\sum_{i=1, j\neq i}^{numberof\ attributesin\ HNNBi} Ip(A_i; A_j \mid C)} \quad (10)$$

In this paper, the dependence of two attributes is represented by the conditional mutual information. The conditional mutual information between attributes $Ip(A_i;A_j \mid C)$ is defined in (6).

The threshold can be set the average of the dependences of all pairs of attributes, which is defined in (11), where $n$ is the number of attributes.

$$average = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} Ip(A_i; A_j \mid C) \qquad (11)$$

### B. Algorithm Description

This algorithm mainly packages the correlative attributes for each attribute in classifier training phase. The algorithm PHNB is depicted as follows:

Training phase: (Input: a set $D$ of training examples)

```
For each value c of C
    Compute P(c) from D;
For each pair of Ai and Aj in D
    For each assignment ai, aj and c
    to Ai,Aj and C
    Compute P(ai|aj,c) from D;
For each pair of attributes Ai and
Aj
    Compute Ip(Ai;Aj|C) from D;
For each Ai
    For each Aj≠Aj in D
    If Ip(Ai;Aj|C)≥threshold, put Aj
    into the bag HNBi;
    else put Ai into the bag NB;
```

Testing phase: (Input: an instance $E$)

```
For each value c of C
    For each Ai in E
    If HNBi exists, computed by the
    equation 9;
    Apply NB algorithm in the bag NB
    Compute c(E);
```

### C. Computational Complexity

Assuming $t$ is the number of instances in the training examples, $n$ is the number of attributes, v is the average number of values for an attribute, and $k$ is the number of different class labels, then the time complexity, which computing the counts for each pair of attribute values given class in each instance, is $O(tn^2)$, the time complexity, which computing the conditional mutual information $Ip(A_i;A_j|C)$ for each pair of attributes, is $O(kn^2v^2)$, and the time complexity, which computing the average for threshold and packaging each attribute, both are $O(n^2)$. Thus, the training time complexity of PHNB is $O(kn^2v^2+tn^2+2n^2)= O(kn^2v^2+tn^2)$. At classification time, assumed the averaged number of attributes in HNB bags is $w(0\leq w\leq n)$, then the time complexity for computing HNB bags is $O(kw^2)$ and computing the bag NB is $O(k(n-w))$, thus the classification time complexity of PHNB is $O(kw^2+k(n-w))=O(kw^2)$. To HNB, the training time complexity is $O(kn^2v^2+ tn^2)$, the classification time complexity is $O(kn^2)$ [8]. So at training phase, the time complexity of PHNB model is similar to HNB model; at classification time, the time complexity is related to the number of attributes which are included in HNB bags, when there are more attributes but less attributes in HNB bags, the efficiency of PHNB is more significantly improved. The test phase is a classification process, so the test time is just influenced by the classification time and the number of instances in test datasets.

## III. Experiments

Our experiments were based on uci-20070111 datasets [9] under the framework of Weka [10] which represents a wide range of data dimension from 9 to 263 and are described in Table 1, thus, we can compare the performance of HNB and PHNB at this wide range. We select nine representative datasets whose dimensions are from low to high. In the experiments, the performance of an algorithm on each dataset has been obtained by 10 runs of 10-fold stratified cross validation. The experimental computer configuration is: RAM 4G, AMD 4-core CPU.

TABLE I. Description Of Dataset Used In The Experiments

| NO | Dataset | Examples | Attributes | Classes | Numeric Value |
|---|---|---|---|---|---|
| 1 | nursery | 12960 | 9 | 5 | N |
| 2 | sick | 3772 | 28 | 2 | Y |
| 3 | Kr_vs_kp | 3196 | 37 | 2 | N |
| 4 | meat_zernike | 2000 | 48 | 10 | Y |
| 5 | optdigits | 5620 | 65 | 10 | Y |
| 6 | Mfeat_fourier | 2000 | 77 | 10 | Y |
| 7 | Arrhythmia | 452 | 263 | 13 | Y |
| 8 | Mfeat_pixel | 2000 | 241 | 10 | N |
| 9 | Mfeat_factors | 2000 | 217 | 10 | Y |

We adopted three preprocessing steps for datasets. Firstly, we used the unsupervised filter named ReplaceMissing Values in Weka to replace the missing values in each dataset. Secondly, we used the unsupervised 10-bin filter Discretize in Weka to discretize all numeric attributes. Finally, we removed all useless attributes using the unsupervised filter Remove in Weka.

The accuracy and test time of the classifiers are shown in Tabel 2, where the performance of PHNB is divided into two statuses: the threshold is average and the optimal value.

Firstly, we compared the performance of HNB and PHNB where the threshold was set the average from Table 2. Compared to HNB, we found that the test time of PHNB was generally comparative, and the time was shorter when the number of attributes was high, otherwise the time was longer, the accuracy of PHNB was comparative to HNB, and on some datasets the accuracy was even higher.

Secondly, we compared the performance of HNB and PHNB where the threshold was set the optimal value from Table 2. Compared to HNB, we found that as the data dimension rising, the test time of PHNB saved, the time was a little longer only when the number of attributes became very low (such as 9), the accuracy of PHNB was generally a little higher, and on some datasets the accuracy became equal or a little lower.

TABLE II. THE EXPERIMENT PERFORMANCE OF HNB AND PHNB

| Dataset NO | HNB | | PHNB(average) | | PHNB(optimal value) | | |
|---|---|---|---|---|---|---|---|
| | Accuracy | Test time | Accuracy | Test time | Threshold | Accuracy | Test time |
| 1 | 94.56% | 0.22s | 94.28s | 0.25s | 0.0058 | 95.01% | 0.25s |
| 2 | 97.79% | 0.17s | 97.73s | 0.09s | 0.138 | 97.79% | 0.16s |
| 3 | 92.25% | 0.17s | 92.52s | 0.16s | 0.024 | 92.67% | 0.14s |
| 4 | 75.45% | 2.06s | 75.4s | 1.56s | 0.28 | 75.5% | 1.14s |
| 5 | 95.88% | 13.31s | 95.92s | 15.84s | 0.12 | 96.19% | 9.92s |
| 6 | 80.05% | 8.43s | 80.3s | 10.41s | 0.25 | 80.15% | 3.19s |
| 7 | 66.15% | 22.81s | 68.58s | 23.38s | 0.2 | 69.02% | 16.08s |
| 8 | 96.05% | 50.58s | 96.05s | 44.78s | 0.15 | 96.05% | 30.66s |
| 9 | 96.25% | 49.58s | 96.0s | 38.42s | 0.45 | 96.1% | 11.8s |

Finally, we compared the performance of PHNB where the threshold was set different value to HNB. Compared to HNB, we found that when setting as the optimal value on most datasets not only the test time was shorter, but the accuracy was also higher. In addition, the threshold is always gotten nearby the average.

On the whole, PHNB significantly reduces the test time of HNB on high-dimensional datasets, meanwhile has a comparative even higher accuracy than HNB. Though on some datasets the accuracy is a little lower, the test time is markedly shorter than HNB. In addition, the threshold is a critical factor in PHNB, when the optimal value set, the advantage of PHNB can be significantly shown. From the experimental results we infer that when there are more attributes in dataset the performance would be much better than HNB and so that also proves the improved idea we proposed in section 2.

## IV. CONCLUSIONS AND FUTURE WORK

Aiming at the high computational complexity of HNB on high-dimensional datasets, PHNB model, which combines the model of HNB and NB, is proposed, and has a better performance than the single model. Our experiments show that the test time of PHNB is significantly reduced on high-dimensional datasets, meanwhile the accuracy of which is comparative to HNB, even higher on some particular datasets.

There are some future works as following:

1. The computation between bags may proceed in parallel. So if the algorithm is transplanted to parallel environment, the efficiency may be higher. That is especially suitable for processing very high-dimensional datasets.
2. The optimal value which is critical to the performance of PHNB, is gotten nearby the average. We get the value through several experiments. It is need to research how to get the optimal value automatically.
3. Due to the restriction of experiment condition, there are no experiments on higher dimensional datasets. That is the focus for future works.

## ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant No.60873129, Shanghai Leading Academic Discipline Project under Grant No.J50103.

## REFERENCES

[1] L. Jiang, D. Wang, Z. Cai and X. Yan : Survey of Improving Naive Bayes for Classification. In: Lecture Notes in Computer Science, Vol. 4632, pp. 134-145, Springer-Verlag , Berlin Heidelberg , 2007 .

[2] Langley, P., Iba, W., Thomas, K.: An analysis of Bayesian classifiers. In: Proceedings of the Tenth National Conference of Artificial Intelligence. AAAI Press, Stanford , pp. 223–228, 1992.

[3] Chickering, D.M.: Learning Bayesian networks is NP-Complete. In: Fisher, D.,Lenz, H. (eds.) learning from Data: Artificial Intelligence and Statistics V. Springer-Verlag, Berlin Heidelberg, pp. 121–130, 2007.

[4] Friedman, Geiger, Goldszmidt.: Bayesian Network Classifiers. In: Machine Learning, Vol. 29. Springer-Verlag, Berlin Heidelberg , pp. 131–163, 1997.

[5] Jiang, L., Zhang, H., Cai, Z., Su, J.: One Dependence Augmented Naive Bayes. In: Lecture Notes in Computer Science, Vol. 3584. Springer-Verlag, Berlin Heidelberg, pp. 186–194, 2005.

[6] Webb, G.I., Boughton, J., Wang, Z.: Not so naive bayes: Aggregating one dependence estimators. In: Machine Learning , Vol. 58. Springer-Verlag, Berlin Heidelberg, pp. 5–24, 2005.

[7] Jiang, L., Zhang, H.: Weightily Averaged One-Dependence Estimators. In: Lecture Notes in Computer Science, Vol. 4099. Springer-Verlag, Berlin Heidelberg, pp. 970–974, 2006.

[8] L. Jiang, H. Zhang, and Z. Cai. A Novel Bayes Model: Hidden Naive Bayes. In: IEEE Transactions on Knowledge and Data Engineering, Vol. 21(10), pp. 1361-1371, 2009.

[9] http://sourceforge.net/projects/weka/files/datasets/UCI%20and%20StatLib/uci-20070111.tar.gz//download

[10] http://www.cs.waikato.ac.nz/ml/weka/