

# One Dependence Augmented Naive Bayes <sup>\*</sup>

Liangxiao Jiang<sup>1</sup>, Harry Zhang<sup>2</sup>, Zhihua Cai<sup>1</sup>, and Jiang Su<sup>2</sup>

<sup>1</sup> Faculty of Computer Science, China University of Geosciences  
Wuhan, Hubei, P.R.China, 430074  
ljjiang@cug.edu.cn

<sup>2</sup> Faculty of Computer Science, University of New Brunswick  
P.O.Box 4400, Fredericton, NB, Canada E3B 5A3  
hzhang@unb.ca

**Abstract.** In real-world data mining applications, an accurate ranking is same important to a accurate classification. Naive Bayes (simply NB) has been widely used in data mining as a simple and effective classification and ranking algorithm. Since its conditional independence assumption is rarely true, numerous algorithms have been proposed to improve Naive Bayes, for example, SBC[1] and TAN[2]. Indeed, the experimental results show that SBC and TAN achieve a significant improvement in term of classification accuracy. However, unfortunately, our experiments also show that SBC and TAN perform even worse than naive Bayes in ranking measured by AUC[3, 4](the area under the Receiver Operating Characteristics curve). This fact raises the question of whether can we improve Naive Bayes with both accurate classification and ranking? In this paper, responding to this question, we present a new learning algorithm called One Dependence Augmented Naive Bayes (simply ODANB). Our motivation is to develop a new algorithm to improve Naive Bayes' performance not only on classification measured by accuracy but also on ranking measured by AUC. We experimentally tested our algorithm, using the whole 36 UCI datasets recommended by Weka[5], and compared it to NB, SBC[1] and TAN[2]. The experimental results show that our algorithm outperforms all the other algorithms significantly in yielding accurate ranking, yet at the same time outperforms all the other algorithms slightly in terms of classification accuracy.

## 1 Introduction

Classification is one of the most important tasks in data mining. Learning Bayesian classifiers is a process of constructing a special Bayesian networks from a given set of preclassified instances, each of which is represented by a vector of attribute values. Assume  $A_i, i = 1, 2, \dots, n$  are  $n$  attributes which take values  $a_i, i = 1, 2, \dots, n$  respectively. Those attributes will be used collectively to

---

<sup>\*</sup> This work was supported by Excellent Youth Foundation of China University of Geosciences(No.CUGQNL0505) and Natural Science Foundation of Hubei of China(No.2001ABB006 and No.2003ABA043).

predict the value  $c$  of the class  $C$ . Thus, the Bayesian classifier represented by a Bayesian network can be defined as:

$$\arg \max_{c \in C} P(c)P(a_1, a_2, \dots, a_n|c) \quad (1)$$

Assume all attributes are independent given the class. That is:

$$P(a_1, a_2, \dots, a_n|c) = \prod_{i=1}^n P(a_i|c) \quad (2)$$

The resulting classifier is called a naive Bayesian classifier, or simply Naive Bayes:

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

It is obvious, Naive Bayes is a probability-based classification model which is based on the assumption that attributes are conditionally mutually independent given the class label. Although Naive Bayes has conceptual and computational simplicity etc many advantages, its unrealistic attribute independence assumption leads to its probability estimations will not be correct [6], if there exists some strong dependent relations among attributes.

Thinking of the limitation of Naive Bayes, in real-world data mining applications, many researchers propose to learn an optimal Bayesian networks to overcome Naive Bayes's limitation of unrealistic attribute independence assumption throughout. Unfortunately, however, it has been proved that learning an optimal Bayesian networks is NP-hard [7]. Therefore, researchers have made a substantial amount of effort to improve Naive Bayes. Research work to improve the Naive Bayes can be broadly divided into two approaches: 1) selecting attributes subsets in which attributes are mutual conditionally independent at most; 2) relaxing the conditional independence assumption by extending the structure of Naive Bayes to represent the dependencies among attributes.

In classification, the predictive ability of a classifier is typically measured by its predictive accuracy on the testing examples. In fact, most classifiers (including Naive Bayes) can also produce probability estimations or "confidence" of the class prediction. Unfortunately, however, this information is completely ignored in classification. This is often taken for granted since the true probability is unknown for the testing examples anyway.

In many data mining applications, however, the classifier's accuracy are not enough, because they cannot express the information how "far-off" (be it 0.45 or 0.01?) is the prediction of each example from its target. For example, in direct marketing, we often need to promote the top X% of customers during gradual roll-out, or we often deploy different promotion strategies to customers with different likelihood of buying some products. To accomplish these tasks, we need more than a mere classification of buyers and non-buyers. We often need a ranking of customers in terms of their likelihood of buying. Thus, a ranking is more desirable than just a classification.

A natural question is how to evaluate a classifier in terms of its ranking performance, rather than classification accuracy. Recently, the area under the Receiver Operating Characteristics curve [3, 4], or simply AUC, has been used for this purpose and received a considerable attention. AUC compares the classifiers' performance across the entire range of class distributions and error costs and is a good "summary" for comparing two classifiers. Hand and Till [8] show that, for binary classification, AUC is equivalent to the probability that a randomly chosen example of class  $-$  will have a smaller estimated probability of belonging to class  $+$  than a randomly chosen example of class  $+$ . They present a simple approach to calculating the AUC of a classifier  $G$  below.

$$\hat{A} = \frac{S_0 - n_0(n_0 + 1)/2}{n_0 n_1}, \quad (4)$$

where  $n_0$  and  $n_1$  are the numbers of negative and positive examples respectively, and  $S_0 = \sum r_i$ , where  $r_i$  is the rank of  $i_{th}$  positive example in the ranked list. From Equation 4, it is clear that AUC is essentially a measure of the quality of a ranking. For example, the AUC of a ranking is 1 (the maximum value of AUC) if there is no positive example preceding a negative example.

In this paper, Our motivation is to develop a new algorithm to improve Naive Bayes' performance not only on classification measured by accuracy but also on ranking measured by AUC. In order to achieve our goal, we present a new learning algorithm called One Dependence Augmented Naive Bayes (simply ODANB). The experimental results show that we have learned improved Naive Bayes with both accurate classification and ranking.

The rest of the paper is organized as follows. In Section 2, we introduce the related work on improving Naive Bayes. In Section 3, we present our new learning algorithm called One Dependence Augmented Naive Bayes (simply ODANB) and make a simple analysis on its advantages and disadvantages. In Section 4, we describe the experimental setup and results in detail. In Section 5, we draw a conclusion.

## 2 Related Work

Naive Bayes is a simple but effective classifier. Although its conditional independence assumption is often violated, it performs surprisingly well in classification [9]. This fact raises the question of whether a Naive Bayesian classifier with less restrictive assumptions can perform even better.

In order to tackle this question effectively, we need an appropriate language and efficient machinery to represent and manipulate independence assertions. Both are provided by Bayesian networks [10]. So, learning Bayesian networks from data has become a rapidly growing field of research. Unfortunately, however, it has been proved that learning an optimal Bayesian networks is NP-hard [7]. In order to escape Bayesian networks's learning complexity, learning improved Naive Bayes has attracted much attention from researchers. Research work to

improve Naive Bayes can be broadly divided into two categories just as described in introduction.

The first category aims to improve Naive Bayes by selecting attributes subsets in which attributes are mutual conditionally independent. For example, Langley and Sage[1] presented an algorithm called Selective Bayesian Classifiers(simply SBC). They used a forward greedy search method to select an attribute subset through the whole space of attributes. They use Naive Bayes' accuracy to evaluate alternative subsets of attributes and consider adding each unselected attribute which can improve the classifier's accuracy at most on each iteration. Their experimental results proved their hypotheses that their algorithm will improve Naive Bayes' accuracy in domains that involve correlated attributes without reducing Naive Bayes' accuracy in domains that don't.

The second category aims to improve Naive Bayes by extending the structure of Naive Bayes to represent dependencies among attributes. For example, Friedman and Goldszmidt[2] singled out an algorithm called Tree Augmented Naive Bayes(simply TAN). They hypothesized the structure among all attributes only is tree-like structure, in which the class node directly points to all attributes nodes and each attribute except the root node of the tree has only one parent from another attribute node. As a result, significant improvement in accuracy is achieved for some datasets compared to Naive Bayes.

### 3 One Dependence Augmented Naive Bayes: ODANB

At first, let us introduce an important definition of conditional mutual information used in our classification algorithm.

Let  $X, Y, Z$  are three variables, then the conditional mutual information between  $X$  and  $Y$  given  $Z$  can be defined by the Equation 5. Roughly speaking, this function measures the information that  $Y$  provides about  $X$  when the value of  $Z$  is known. Some more information about conditional mutual information can be found in [2].

$$I_P(X; Y|Z) = \sum_{x, y, z} P(x, y, z) \log \frac{P(x, y, z)P(z)}{P(x, z)P(y, z)} \quad (5)$$

In order to improve Naive Bayes' performance measured by accuracy and AUC, we present a novel classification algorithm called One Dependence Augmented Naive Bayes (simply ODANB) to weaken the attribute independence assumption by adding a parent  $A_{ip}, i = 1, 2, \dots, n$  for some attributes  $A_i, i = 1, 2, \dots, n$ . Our algorithm classifies instance using the formulation:

$$\arg \max P(C) \prod_{i=1}^n P(A_i|A_{ip}, C) \quad (6)$$

where

$$P(A_i|A_{ip}, C) = \begin{cases} P(A_i|A_m, C) & I_P(A_i; A_m|C) \geq \text{average} \\ P(A_i|C) & I_P(A_i; A_m|C) < \text{average} \end{cases} \quad (7)$$

where  $A_m$  satisfies

$$I_P(A_i; A_m|C) = \max I_P(A_i; A_j|C), j \neq i = 1, 2, \dots, n \quad (8)$$

and *average* is defined as

$$average = \frac{1}{n(n-1)} \sum_{i=1}^n \sum_{j \neq i=1}^n I_P(A_i; A_j|C) \quad (9)$$

Now, let us look back how to calculate  $P(A_i|A_{ip}, C)$ . At first, we calculate the conditional mutual information  $I_P(A_i; A_j|C), i \neq j$  between each pair of attributes, and calculate their average conditional mutual information. Secondly, we take the attribute with maximum conditional mutual information and above (including equal) the average conditional mutual information as one attribute's attribute parent. At last, we calculate this attribute's conditional probability given class and its attribute parent. Of course, if all conditional mutual information between the one attribute and all the other attributes are below the average conditional mutual information, then this attribute hasn't attribute parent. So, we only need to calculate its conditional probability given class.

Compared to Tree Augmented Naive Bayes(TAN)[2], our classification algorithm(ODANB) has the following advantages at least:

1. Our experimental results measured by classification accuracy and AUC have already proved our classification algorithm has better performance than all the other algorithm used to compare.
2. ODANB's effectiveness and efficiency is higher than TAN, because it substitutes calculating each attribute's maximum conditional mutual information for TAN's searching a maximum conditional mutual information weighted spanning tree. ODANB's time complexity only is  $o(n^2 \cdot N + n^2)$ , where  $n$  is the number of attributes and  $N$  is the number of training instances. It is lower than TAN's time complexity  $o(n^2 \cdot N + n^2 \cdot \log n)$ .
3. ODANB is easier for researchers to understand and implement, because its learning process doesn't suffer from TAN's searching a maximum conditional mutual information weighted spanning tree.

## 4 Experimental Methodology And Results

We run our experiments on the 36 UCI data sets recommended by Weka[5]. All the preprocessing stages of data sets are carried out by the Weka[11]. They mainly include the following three processes:

1. We use the filter of ReplaceMissingValues in Weka to replace the missing values of attributes.
2. We use the filter of Discretize in Weka to discretize numeric attributes.
3. It is well-known that, if the number of values of an attribute is almost equal to the number of instances in the data set, this attribute does not contribute

any information to classification. So we use the filter of Remove in Weka to delete these attributes. In these 36 data sets, there only exists three this type of attributes, namely Hospital Number in colic.ORIG, Instance Name in Splice and Animal in zoo.

We conduct our experiments to compare ODANB on accuracy and AUC with NB, SBC and TAN. In our all experiments, the accuracy of each classifier is based on the percentage of successful predictions on the test sets of each data set, and multi-class AUC has been calculated by measure[8]. The accuracy and AUC of each classifier was measured via the ten-fold cross validation for all data sets. Runs with the various classifiers were carried out on the same training sets and evaluated on the same test sets. In particular, the cross-validation folds are the same for all the experiments on each data set. Throughout, we compare our algorithm with each other algorithm via two-tailed t-test with significantly different probability of 0.95, because we speak of two results for a dataset as being “significantly different” only if the difference is statistically significant at the 0.05 level according to the corrected two-tailed t-test.

Table 1 show the accuracy of each classifier on the test sets of each data set, the average accuracy are summarized at the bottom of the table. Table 2 show the AUC of each classifier on the test sets of each data set, the average AUC are summarized at the bottom of the table. Table 3 shows the results of two-tailed t-test between each pair of algorithms, each entry  $w/t/l$  means that the algorithm at the corresponding row wins in  $w$  datasets, ties in  $t$  datasets, and loses in  $l$  datasets, compared to the algorithm at the corresponding column.

The detailed results displayed in Table 1–Table 3 show that our algorithm outperforms all the other algorithms used to compare measured by accuracy and AUC. Now, we summarize the highlights as follows:

1. Our algorithm’s performance on classification measured by accuracy outperforms all the other algorithms. ODANB’s average classification accuracy is 83.37, but the best algorithm of the other algorithms is TAN with average classification accuracy of 82.96. Moreover, the  $w/t/l$  value between ODANB and TAN is 5/27/4.
2. Our algorithm’s performance on ranking measured by AUC outperforms all the other algorithms. Although Our algorithm’s average AUC(89.30) is a little lower than that of NB(89.61), the  $w/t/l$  value between ODANB and NB is 9/25/2.
3. TAN improves NB’s performance on classification measured by accuracy, but they perform even worse than Naive Bayes in ranking measured by AUC. The average AUC of NB(89.61) is higher than that of TAN(80.58) significantly. Moreover, the  $w/t/l$  value between TAN and NB is 1/12/23.

## 5 Conclusions

Naive Bayes delivers fast and effective classification with a clear theoretical foundation. However, It is hampered by the limitations of the attribute independence

**Table 1. Experimental results on classification accuracy and standard deviation.** ODANB: One Dependence Augmented Naive Bayes; NB: Naive Bayes; SBC: Selective Bayesian Classifiers; TAN: Tree Augmented Naive Bayes with smoothed parameter of 5.0.

| Datasets      | ODANB       | NB          | SBC         | TAN         |
|---------------|-------------|-------------|-------------|-------------|
| anneal        | 96.55±1.11  | 94.32±2.38  | 96.88±2.5   | 96.66±2.35  |
| anneal.ORIG   | 90.31±2.57  | 87.53±4.69  | 88.75±3.72  | 87.98±3.62  |
| audiology     | 62.27±10.02 | 71.23±7.03  | 76.01±7.05  | 75.16±8.45  |
| autos         | 78.55±6.92  | 64.83±11.18 | 67.71±11.27 | 76.07±10.01 |
| balance-scale | 91.36±1.38  | 91.36±1.38  | 91.36±1.38  | 86.08±3.18  |
| breast-cancer | 69.61±8.45  | 72.06±7.97  | 73.45±8.91  | 66.82±7.01  |
| breast-w      | 96.99±1.85  | 97.28±1.84  | 96.42±2.26  | 96.71±1.79  |
| colic         | 81.25±5.46  | 78.81±5.05  | 81.77±4.89  | 77.18±7.04  |
| colic.ORIG    | 68.76±4.55  | 75.26±5.26  | 75.53±6.15  | 75.51±7.15  |
| credit-a      | 82.9±3.54   | 84.78±4.28  | 85.51±4.16  | 84.64±5.03  |
| credit-g      | 73.4±4.58   | 76.3±4.76   | 74.1±3.87   | 73.4±4.12   |
| diabetes      | 73.84±7.31  | 75.4±5.85   | 75.53±5.07  | 75.13±4.71  |
| glass         | 60.28±9.31  | 60.32±9.69  | 57.99±6.89  | 55.71±10.81 |
| heart-c       | 80.46±10.31 | 84.14±4.16  | 82.47±7.61  | 77.53±7.41  |
| heart-h       | 79.66±5.97  | 84.05±6.69  | 79±9.77     | 79.97±6.39  |
| heart-statlog | 80±11.07    | 83.7±5      | 79.26±9.75  | 81.11±3.68  |
| hepatitis     | 85.13±7.36  | 83.79±8.79  | 80.63±6.8   | 83.83±8.05  |
| hypothyroid   | 92.63±0.82  | 92.79±1.02  | 93.53±0.66  | 92.79±1.06  |
| ionosphere    | 90.9±5.1    | 90.89±3.49  | 91.17±4.12  | 90.6±3.83   |
| iris          | 94.67±8.2   | 94.67±8.2   | 97.33±4.66  | 90.67±11.42 |
| kr-vs-kp      | 90.52±1.54  | 87.89±1.81  | 94.34±1.23  | 93.18±1.6   |
| labor         | 90±14.05    | 93.33±11.65 | 77±11.91    | 88±11.46    |
| letter        | 77.89±0.89  | 70±0.81     | 70.57±0.88  | 80.45±0.91  |
| lymph         | 82.43±7.18  | 85.67±9.55  | 79±6.84     | 84.38±9.1   |
| mushroom      | 99.94±0.09  | 95.57±0.45  | 99.67±0.23  | 99.77±0.12  |
| primary-tumor | 44.26±4.06  | 46.89±4.32  | 46.02±5.19  | 48.37±5.83  |
| segment       | 94.2±1.12   | 88.92±1.95  | 90.43±1.96  | 86.36±2.36  |
| sick          | 97.59±0.48  | 96.74±0.53  | 97.59±0.69  | 97±0.4      |
| sonar         | 77.02±11.28 | 77.5±11.99  | 70.71±12.97 | 71.62±12.64 |
| soybean       | 91.51±3.94  | 92.08±2.34  | 91.79±2.72  | 93.41±2.1   |
| splice        | 93.07±2.34  | 95.36±1     | 94.76±1.6   | 95.39±1.35  |
| vehicle       | 71.04±2.8   | 61.82±3.54  | 60.65±4.73  | 69.86±3.47  |
| vote          | 94.04±3.9   | 90.14±4.17  | 95.18±3.93  | 93.12±4.02  |
| vowel         | 91.82±2.31  | 67.07±4.21  | 68.69±3.47  | 83.43±3.84  |
| waveform-5000 | 81.26±0.91  | 79.96±1.92  | 81.32±1.54  | 81.52±1.21  |
| zoo           | 95.18±8.15  | 94.18±6.6   | 93.18±7.93  | 97.09±4.69  |
| Mean          | 83.37±5.03  | 82.41±4.88  | 82.09±4.98  | 82.96±5.06  |

**Table 2. Experimental results on AUC and standard deviation.** ODANB: One Dependence Augmented Naive Bayes; NB: Naive Bayes; SBC: Selective Bayesian Classifiers; TAN: Tree Augmented Naive Bayes with smoothed parameter of 5.0.

| Datasets      | ODANB       | NB          | SBC         | TAN         |
|---------------|-------------|-------------|-------------|-------------|
| anneal        | 96.53±0.22  | 95.9±1.3    | 94.7±3.92   | 92.97±2.51  |
| anneal.ORIG   | 95.17±2.76  | 94.49±3.67  | 94.35±4.31  | 85.42±7.04  |
| audiology     | 70.84±0.58  | 70.96±0.73  | 70.98±0.67  | 70.16±0.55  |
| autos         | 93.07±4.06  | 89.18±4.93  | 90.43±3.43  | 90.28±2.59  |
| balance-scale | 84.46±4.1   | 84.46±4.1   | 84.46±4.1   | 76.47±7.56  |
| breast-cancer | 66.57±11.08 | 69.71±15.21 | 67.67±12.63 | 67.4±10.4   |
| breast-w      | 99.04±1.05  | 99.19±0.87  | 99.16±0.62  | 98.74±1.32  |
| colic         | 84.48±5.99  | 83.71±5.5   | 84.86±7.13  | 50.6±8.29   |
| colic.ORIG    | 72.53±5.61  | 80.67±6.98  | 81.82±4.9   | 62.89±7.73  |
| credit-a      | 90.19±3.79  | 92.09±3.43  | 87±3.75     | 63.3±13.3   |
| credit-g      | 75.65±6.28  | 79.27±4.74  | 77.41±4.67  | 60.18±6.84  |
| diabetes      | 80.88±5.95  | 82.31±5.17  | 82.79±5.04  | 74.18±5.87  |
| glass         | 79.94±6.88  | 80.5±6.65   | 80.97±8.37  | 84.79±4.34  |
| heart-c       | 83.85±0.79  | 84.1±0.54   | 83.87±0.64  | 82.96±1.12  |
| heart-h       | 83.23±0.84  | 83.8±0.7    | 82.83±1.38  | 82.69±0.72  |
| heart-statlog | 88.18±9.27  | 91.3±4.19   | 87.98±6.91  | 80.12±11.94 |
| hepatitis     | 86.04±12.18 | 88.99±8.99  | 83.62±12.29 | 53.83±14.97 |
| hypothyroid   | 86.5±8.64   | 87.37±8.52  | 85.25±8.16  | 84.03±12.22 |
| ionosphere    | 97.67±1.71  | 93.61±3.36  | 92.26±5.26  | 72.05±7.4   |
| iris          | 98.58±2.67  | 98.58±2.67  | 99±1.46     | 94.17±5.51  |
| kr-vs-kp      | 97.13±0.9   | 95.17±1.29  | 96.41±0.78  | 87.21±1.49  |
| labor         | 91.67±18    | 98.33±5.27  | 65.83±32.5  | 68.33±40.41 |
| letter        | 98.45±0.16  | 96.86±0.24  | 97.03±0.23  | 94.5±0.25   |
| lymph         | 89.02±2.62  | 89.69±1.49  | 88.14±3.35  | 85.56±6.98  |
| mushroom      | 100±0       | 99.79±0.04  | 99.98±0.02  | 99.87±0.04  |
| primary-tumor | 78.18±0.78  | 78.85±1.35  | 78.88±1.45  | 76.39±1.9   |
| segment       | 99.55±0.23  | 98.51±0.46  | 98.93±0.42  | 95.35±1.06  |
| sick          | 97.48±0.88  | 95.91±2.35  | 94.5±4.28   | 73.25±2.73  |
| sonar         | 81.64±12.5  | 85.48±10.82 | 79.89±13.1  | 67.4±13.83  |
| soybean       | 99.46±0.72  | 99.53±0.6   | 99.08±0.74  | 96.73±1.59  |
| splice        | 99.05±0.57  | 99.41±0.22  | 99.14±0.36  | 97.72±0.68  |
| vehicle       | 87.97±3.13  | 80.81±3.51  | 81.31±4.02  | 76.86±3.8   |
| vote          | 98.16±1.47  | 96.56±2.09  | 94.26±4.14  | 93.49±1.38  |
| vowel         | 99.49±0.2   | 95.81±0.84  | 96.12±0.59  | 92.33±1.23  |
| waveform-5000 | 94.38±0.62  | 95.27±0.58  | 95.12±0.76  | 78.9±2.03   |
| zoo           | 89.88±4.05  | 89.88±4.05  | 89.06±4.49  | 89.88±4.05  |
| Mean          | 89.30±3.92  | 89.61±3.54  | 87.92±4.75  | 80.58±5.99  |

**Table 3. Results of two-tailed t-test on accuracy and AUC.**

|          |       | NB      | SBC     | TAN     |
|----------|-------|---------|---------|---------|
| accuracy | SBC   | 6/29/1  |         |         |
|          | TAN   | 6/26/4  | 4/28/4  |         |
|          | ODANB | 10/25/1 | 6/27/3  | 5/27/4  |
| AUC      | SBC   | 4/31/1  |         |         |
|          | TAN   | 1/12/23 | 0/16/20 |         |
|          | ODANB | 9/25/2  | 8/26/2  | 22/14/0 |

assumption. The current work is motivated by the desire not only to improve Naive Bayes' performance not only on classification measured by accuracy but also on ranking measured by AUC. In this paper, we present a novel classification algorithm called One Dependence Augmented Naive Bayes (simply ODANB) by adding an attribute parent for some attributes. Our experimental results show that our classification algorithm outperforms NB, SBC and TAN measured by accuracy and AUC. In a word, we believe that we have been successful in our goal of developing a data mining algorithm that retains the computational simplicity and direct theoretical foundation of naive Bayes while alleviating the limitations of its attribute independence assumption.

## References

- Langley, P., Sage, S. Induction of selective Bayesian classifiers. in Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, 1994, pp. 339-406.
- Friedman, Geiger, and Goldszmidt. "Bayesian Network Classifiers", Machine Learning, Vol. 29, 131-163, 1997.
- Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition,30,1145-1159.
- Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: comparison under imprecise class and cost distribution. Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. AAAI Press (1997) 43-48
- <http://prdownloads.sourceforge.net/weka/datasets-UCI.jar>
- Bennett, P.N., Assessing the Calibration of Naive Bayes' Posterior Estimates. In Technical Report No. CMU-CS100-155. 2000.
- Chickering, D. M. (1996). Learning Bayesian networks is NP-Complete. In Fisher, D. and Lenz, H., editors, Learning from Data: Artificial Intelligence and Statistics V, pages 121-130. Springer-Verlag.
- Hand,D. J., and Till,R. J., A simple generalisation of the area under the ROC curve for multiple class classification problems. Machine Learning, 45:171-186, 2001.
- Domingos, P., Pazzani M.: Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. Machine Learning **29** (1997) 103-130
- Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. San Francisco, CA: Morgan Kaufmann.
- Witten,I. H., and Frank,E., Data Mining-Practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann, 2000.