IG-NB: A Hybrid Method for Data Classification

Mohammad Reza Keyvanpour, and Fahimeh Baesi

***Abstract*—**These Naïve Bayesian (NB) classifiers are a well-known and powerful type of classifiers. If we ignore conditional independence assumption of attributes, the simplicity and accuracy of this method will decrease. In this Paper, we propose a novel model Based on Decision Tree, called "Information Gain-Naïve Bayesian"(IG-NB). The objective of this work is to improve the results of the NB algorithm. IG-NB starts by selecting the attributes with the least ability of separation based on Information Gain in the first level of ID3. The remained attributes will send to Naïve Bayesian Algorithm. The preprocessing attributes in our method improves Naïve Bayesian efficiency. Moreover, IG-NB method can increase rate and accuracy of Naïve Bayesian algorithm because of deleting low importance attributes. The IG-NB method has been tested in five different UCI datasets and results indicate acceptable efficiency.

***Keywords*—**Classification, Naive Bayesian, Decision Tree, Entropy

I. Introduction

TWO of the most widely used and successful methods of classification are ID3 Decision Trees and Naïve Bayesian (NB) [1][2].

Several researchers have emphasized on the issue of redundant attributes, as well as advantages of feature selection for the Naïve Bayesian classifier, not only for induction learning [2]. Pazzani [16,17 explorers the methods of joining two (or more) related attributes into a new compound attributes where the attributes dependence are present [2]. Another method, boosting on Naïve Bayesian Classifier [18] has been experimented by applying series of classifiers to the problem and paying more attention to the example misclassified by its predecessor [1]. However, it was shown that it fail on average in a set of natural domain [3]. Hall and Frank investigated a simple semi-naïve Bayesian ranking method that combines Naïve Bayesian with induction of decision tables. Langley and Sage [19] use a wrapper approach for the subset selection to only select relevant feature for NB.

Naïve Bayesian can suffer from oversensitivity to redundant and/or irrelevant attributes [2]. If two or more attributes are highly correlated, they receive too much weight in the final decision as to which class an example belongs to [2].This lead to a decline in accuracy of prediction in domains with correlated features [2]. Unlike the Naïve Bayesian, ID3 does not suffer from this problem, because if two attributes are correlated, it will not be possible to use both of them to split the training set, since this would lead to exactly the same split, which makes no difference to the existing tree. This get is one of the main reasons ID3 performs better than NB on domains with correlated attributes. It has been shown that Naïve Bayesian Classifier is extremely effective in practice and difficult to improve upon [2,20].

In this Paper, we present a new method (Information Gain-Naïve Bayesian) to improve efficiency of the Naïve Bayesian classifier.

IG-NB method increase rate and accuracy of Naïve Bayesian algorithm because of deleting low importance attributes. ID3's calculations gain these attributes. Figure 1 show the architecture of IG-NB method for Improvement Naïve Bayesian Classifier.

The paper is organized as follows: the background of classifier, including features and problems is explained in section2. In section 3, we give a brief introduction to ID3 Decision Tree. Section 4 describes proposed method (IG-NB). Experimental evaluation and conclusion are given in section 5 and 6, respectively.

II. Procedure For Paper Submission

The Naive Bayesian Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayesian can often outperform more sophisticated classification methods. In this section we describe Naïve Bayesian algorithm, features and problem.

A. Algorithm Description

Naïve Bayesian classifier is one of the most effective and efficient classification algorithms for machine learning and data mining. This Classifier is a straightforward and frequently used method for supervised learning [5]. It provides a flexible way for dealing with any number of attributes or classes, and is based on probability theory. It is the asymptotically fastest learning algorithm that examines all its training input. It has been demonstrated to perform surprisingly well in a very wide variety of problems in spite of the simplistic nature of the model [2].

Let $X=(X_1,...,X_n)$ be a vector of observed random variables, called features, where each feature takes values from its domain D_i. The set of all feature vectors (examples, or states), is denoted $\Omega = D_1 \times ... \times D_n$. Let C be an unobserved random variable denoting the class of an example, where C can take one of *m* values $c \in \{0,...,m-1\}$. Capital letters, such as *Xi*, will

Mohammad Reza Keyvanpour, and Fahimeh Baesi, Department of Computer Engineering, Alzahra University, Tehran, Iran.

denote variables, while lower-case letters such as x_i , will denote their values; boldface letters will denote vectors [21].

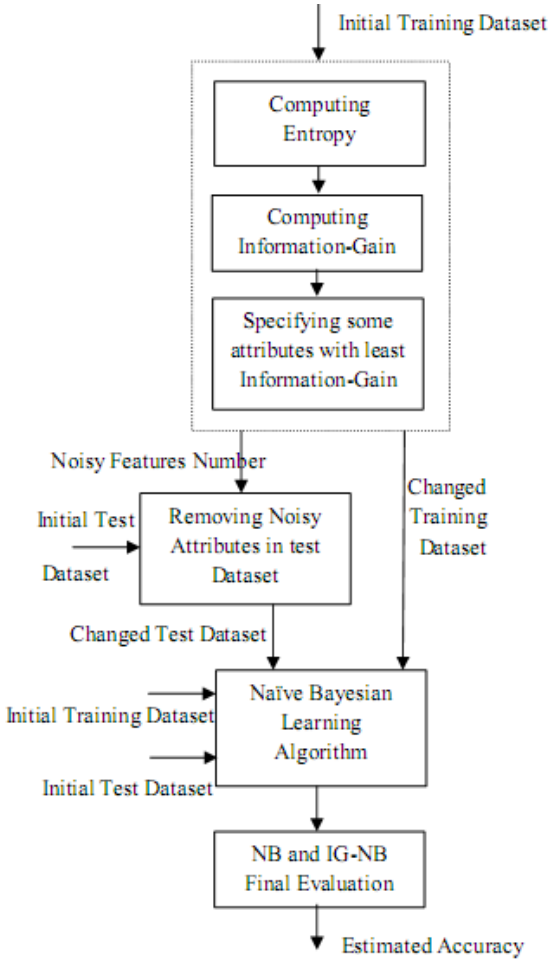

Fig.1 The system architecture of IG-NB method

A function $g:\Omega\to\{0,...,m-1\}$, where $g(x)=C$, denotes a concept to be learned. Deterministic g(x) corresponds to a concept without noise, which always assigns the same class to a given example (e.g., disjunctive and conjunctive concepts are deterministic). In general, however, a concept can be noisy, yielding a random function g(x) [21].

A classifier is defined by a (deterministic) function $h:\Omega\to\{0,...,m-1\}$ (a hypothesis) that assigns a class to any given example. A common approach is to associate each class I with a discriminate function $f_i(x), i=0,...,m-1$, and let the classifier select the class with maximum function on a given example: $h(x)=\arg\max_{i\in\{0,...,m-1\}} f_i(x)$.

The Bayesian classifier $h^*(x)$ (that we also call Bayesian-optimal classifier and denote BO(x)), uses as discriminate functions the class posterior probabilities given a feature vector, i.e. $f_i^*(x)=P(C=i\,|\,X=x)$. Applying Bayesian gives $P(C=i\,|\,X=x)=\frac{P(X=x\,|\,C=i)P(C=i)}{P(X=x)}$, where P(X=x) is identical for all classes, and therefore can be ignored. This yields Bayesian discriminate functions

$$f_i^*(x)=P(X=x\,|\,C=i)P(C=i) \quad (1)$$

Where $P(X=x\,|\,C=i)$ is called the class-conditional probability distribution (CPD). Thus, the Bayesian classifier find the maximum a posterior probability (MAP) hypothesis given example *x*.

$$h^*(x)=\arg\max_i P(X=x\,|\,C=i)P(C=i) \quad (2)$$

However, direct estimation of P(X=x |C=i) from a given set of training examples is hard when the feature space high-dimensional. Therefore, approximations are commonly used, such as using the simplifying assumption that features are independent given the class. This yields the naïve Bayesian classifier NB(x) defined by discriminate functions.

$$f_i^{NB}(x)=\Pi_{j=1}^{n}P(X_j=x_j\,|\,C=i) \quad (3)$$

The probability of a classification error or risk of a classifier h is defined as

$$R(h)=P(h(X)\neq g(X))=\sum_{x\in\Omega}P(h(x)\neq g(x))P(X=x)=E_x\{P(h(x)\neq g(x))\},\ R^*=R(h^*) \quad (4)$$

Where E_x is the expectation over x. denotes the Bayesian error (Bayesian risk).

We say that classifier h is optimal on a given problem if its risk coincides with the Bayesian risk. Assuming there is no noise (i.e. zero Bayesian risk), a concept is called separable by a set of function $S=\{f_c(x)\,|\,c=0...,m-1\}$ if every example x is classified correctly when using each $f_c(x)$ as discriminate functions.

As a measure of dependence between two features X_k and X_j we use the class-conditional mutual information [22], which can be defined as

$$I(X_k;X_j\,|\,C)=H(X_k\,|\,C)+H(X_j\,|\,C)-H(X_k,X_j\,|\,C) \quad (5)$$

,

Where $H(A\,|\,C)$ is the class-conditional entropy of A, defined as:

$$-\sum_i P(C=i)\sum_t P(A=t\,|\,C=i)\log P(A=t\,|\,C=i) \quad (6)$$

Mutual information is zero when X_k and X_j are mutually independent given class C, and increase with increasing level of dependence, reaching the maximum when one of feature is deterministic function of the other [21].

In naive Bayesian classifiers, every feature gets a say in determining which label should be assigned to a given input value. To choose a label for an input value, the naive Bayesian classifier begins by calculating the prior probability of each label, which is determined by checking frequency of each label in the training set. The contribution from each feature is then combined with this prior probability, to arrive at a likelihood estimate for each label. The label whose likelihood estimate is the highest is then assigned to the input value. Figure 2 illustrates this process.

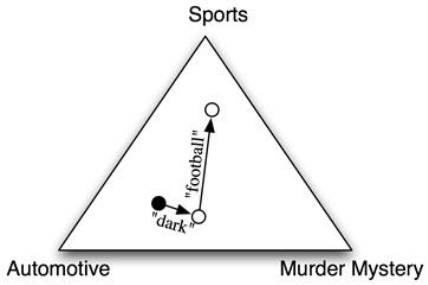

Fig.2 An abstract illustration of the procedure used by the naive Bayesian classifier to choose the topic for a document.

Naive Bayesian begins by calculating the prior probability of each label, based on how frequently each label occurs in the training data. Every feature then contributes to the likelihood estimate for each label, by multiplying it by the probability that input values with that label will have that feature. The resulting likelihood score can be thought of as an estimate of the probability that a randomly selected value from the training set would have both the given label and the set of features, assuming that the feature probabilities are all independent.

B. Feature of Naïve Bayesian

Surprisingly, simple Naïve Bayesian classifier with strong assumption of independence among attributes is competitive with state-of-the-art classifiers and has good performance in a wide variety of domains including many domains where there are clear dependencies between the attributes.

NB classification is simple and computationally efficient. All the probabilities required to build a NB classifier can be found in one scan and the model can be updated easily. Therefore training is linear in both the number of instances and attributes, which is one of the great strengths of NB [5, 6]. In addition, since the model has the form of a product, it can be converted into a sum through the use of logarithms with significant consequent computational advantages [7].

Compared to other classifiers, NB requires relatively little data for training. It trains very quickly, requires little storage space during both training and classification, is easily implemented, and do not have lot of parameters such as Neural Networks and Support Vector Machines [7, 8].

Other advantages of NB classifier involve its implementation and transparency. NB classification takes into account evidence from many attributes to make the final prediction and is very transparent; it is easily grasped by users like physicians who find that probabilistic explanations replicate their way of diagnosing [9, 10]. NB is naturally robust to missing values since these are simply ignored in computing probabilities and hence have no impact on the final decision [7, 11]. In addition to missing values, NB is also robust to noise and irrelevant attributes and therefore it has attracted much attention from researchers [12].

C. Problems with Naïve Bayesian

The central assumption in Naïve Bayesian classification is that given a particular class membership, the probabilities of particular attributes having particular values are independent of each other [2]. However, this assumption is often violated in reality. For example, in demographic data, many attributes have obvious dependencies, such as age and income [2].

NB is extremely effective in practice and difficult to improve upon, however NB can suffer from oversensitivity to redundant or irrelevant attributes [5]. If two or more attributes are highly correlated, they receive too much weight in the final decision as to which class an example belongs to. This leads to a decline in accuracy of prediction in domains with correlated features. Since NB may place too much weight on the influence from the two attributes, and too little on the other attributes, which can result in classification bias [5] [13-15].

Although it has been explained by [4] that NB can work well in some cases where the attribute independence assumption is violated, but the fact remains that probability estimation is less accurate and performance degrades when attribute independence does not hold. Therefore, many techniques have been developed to reduce the "naivety" of the NB classifier and previous research has shown that variants of NB technique that explicitly adjust the naïve strategy can improve upon the prediction accuracy of the NB classifier in many domains [5].

However, some researchers have shown that although irrelevant features should theoretically not hurt the accuracy of Naïve Bayesian, they do degrade performance in practice [2][13]. This paper illustrates that if those redundant and/or irrelevant attributes are eliminated, the performance of Naïve Bayesian classifier can significantly increase.

III. Decision Tree

A decision tree is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a decision.

Decision tree are commonly used for gaining information for the purpose of decision making. Decision tree starts with a root node on which it is for users to take actions. From this node, users split each node recursively according to decision tree learning algorithm. The final result is a decision tree in which each branch represents a possible scenario of decision and its outcome. Decision tree learning is one of the most widely used and practical methods for inductive inference [6].

Decision trees classify instances by traverse from root node to leaf node. We start from root node of decision tree, testing the attribute specified by this node, and then moving down the tree branch according to the attribute value in the given set. This process is the repeated at the sub tree level.

The three widely used decision tree learning algorithms are: ID3, ASSISTANT and C4.5. We will use ID3 in this paper.

The basic decision tree structure is shown in figure 3.

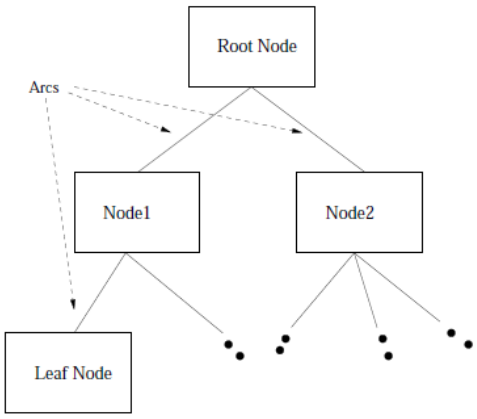

Fig.3 Basic decision tree structure

A. Decision Tree Learning Algorithm-ID3 Basic

ID3 is a simple decision tree learning algorithm developed by Ross Quinlan (1983). The basic idea of ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node. In order to select the attribute that is most useful for classifying a given sets use a metric-information gain.

To find an optimal way to classify learning set, we need to do is to minimize the questions asked (i.e. minimizing the depth of the tree). Thus, we need some function which can measure which questions provide the most balanced splitting. The information gain metric is such a function.

The main ideas behind the ID3 algorithm are:

- Each non-leaf node of a decision tree corresponds to an input attribute, and each arc to a possible value of that attribute. A leaf node corresponds to the expected value of the output attribute when the input attributes are described by the path from the root node to that leaf node.
- In a "good" decision tree, each non-leaf node should correspond to the input attribute which is the most informative about the output attribute amongst all the input attributes not yet considered in the path from the root node to that node. This is because we would like to predict the output attribute using the smallest possible number of questions on average.
- Entropy is used to determine how informative a particular input attribute is about the output attribute for a subset of the training data. Entropy is a measure of uncertainty in communication systems introduced by Shannon (1948). It is fundamental in modern information theory.

B. Entropy[6]

Entropy measure homogeneity of learning set. Without loss of generality assume that the resulting decision tree classifies instances into two categories. We can call them P (positive) and N (negative).

S is a set that contain positive and negative targets. The entropy of S related to this Boolean classification is:

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_- \quad (7)$$

p_+ is **proportion of positive examples in S**.

p_- is **proportion of negative examples in S.**

Entropy is a measure of the impurity in a collection of training sets. You will see at the following that how it is related to the optimization of our decision making in classifying the instances.

C. Information Gain [6]

Information Gain measure the expected reduction in Entropy. To minimize the decision tree depth, we need to select the optimal attribute for splitting the tree node, which we can easily imply that the attribute with the most entropy reduction is the best choice.

The information gain, Gain(S, A) of an attribute A is:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (8)$$

We can use this notion of gain to rank attributes and to build decision trees where at each node is located the attribute with greatest gain among the attributes not yet considered in the path from the root.

IV. Proposed Method (Information Gain-Naïve Bayesian)

As described in section 2, one of the Naïve Bayesian problems is that, if some attributes have high dependence to each other, they will get high weight for dependency of a sample to a special class. This will decrease prediction accuracy in Domains with dependent attributes. But, ID3 learning method doesn't have this problem. Because, if we have two dependent attributes, this method won't use them simultaneously for splitting training set.

On the other hand, Naïve Bayesian learning algorithm is sensitive to initial parameters and must survive a learning stage. This phase is done by human or an algorithm and is very time consuming.

In this paper, we suggest a new method for improvement Naïve Bayesian Algorithm. In the proposed method, we use ID3 learning algorithm scales .i.e. first we calculate Information Gain for all attributes in the first level, then because of this issue that Naïve Bayesian is dependent on attribute values and they values is high effectiveness in final result, we delete attributes with less information- gain values within all attributes. Then remained attributed will send to Naïve Bayesian Classifier. Number of attributes that can be deleted is diverse and dependent to number, importance and correlation within attributes.

V. Experimental Evaluation

Table 1 compares IG-NB and NB Algorithms on five UCI datasets (monk1, monk2, monk3, wine and abalone). Result show that proposed method improve NB algorithm on four

dataset. The sum of removed noisy feature for abalone dataset can change final results.

Table 2 and Figure 6 compare IG-NB and NB Algorithms on three UCI datasets. These three sets are 2-classes. The classifier accuracy is determined by False Negative (FN), False Positive (FP) and Accuracy criterion. False positives result when a test falsely or incorrectly reports a positive result. On the other hand, false negatives result when a test falsely or incorrectly reports a negative result and accuracy is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative

As mentioned, NB classification is simple and computationally efficient. For these five datasets, Train is very quickly and requires little storage space during both training and classification. Also, NB is robust to noise and irrelevant attributes. The Proposed method by removing the noisy attributes can have good effect on training and testing time.

In Figure 4 and 5 the performance of Naïve Bayesian algorithm and its improved version for four UCI dataset is shown.

The results confirm the initial hypotheses. The performance of the proposed IG-NB classifier is quite impressive. Its accuracy is better than the NB on the presented domains.

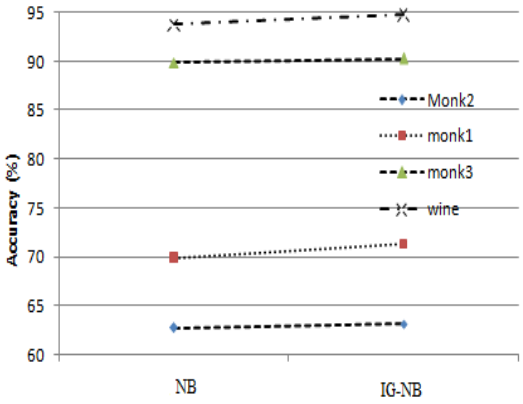

Fig.4 NB and IG-NB Comparison on 4 UCI Datasets (1)

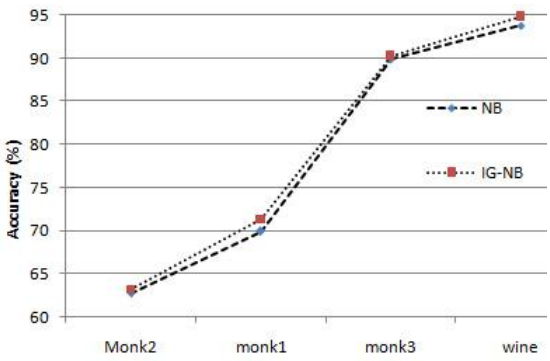

Fig.5 NB and IG-NB Comparison on 4 UCI Datasets (2)

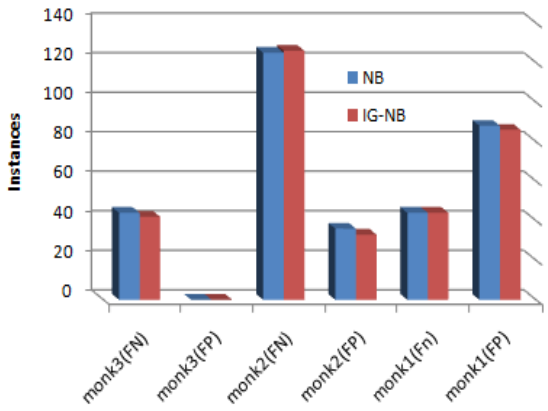

Fig.6 FP and FN Criterion Evaluation on three UCI Datasets

VI. Conclusion

Naïve Bayesian classifier is one of the most effective and efficient classification algorithms based on applying Bayesian theorem with strong (naive) independence assumptions. Many techniques have been developed to reduce the "naivety" of the NB classifier and previous research has shown that variants of NB can improve upon the prediction accuracy of the NB classifier in many domains. On the other hand, we believe that ID3 does not use redundant attributes in constructing decision trees, since they cannot generate different splits of training data.In this paper we suggest a simple method that use ID3 decision tree for specifying the noisy features. After deleting the selective attributes, remained attributes send to Naïve Bayesian classifier for classification. This is to be used to improve Naïve Bayesian learning. The empirical evidence shows that this method is very fast and surprisingly successful. This Selective Naïve Bayesian classifier (we called IG-NB) is asymptotically at least as accurate as the basic Naïve Bayesian on each of the domains on which the experiments were performed. Also, it learns faster than both ID3 and NB on each of these domains.

Table I
Experimental Results For Naïve Bayesian And Ig-Naïve Bayesian

Dataset	NB Accuracy	IGNB Accuracy	Training Set number	Test Set Number	Sum of Removed Features
Monk1	69.91	71.30	123	429	2
Monk2	62.73	63.19	168	430	2
Monk3	89.81	90.28	121	431	2
Wine	93.75	94.82	50	128	3
Abalone	45.99	45.51	400	3777	2
Abalone	45.99	45.88	400	3777	3

TABLE II
EXPERIMENTAL RESULTS (NB AND IGNB) OF THE 2 CLASSES DATA SETS WITH FN AND FP CRITERION

Dataset	Accuracy	FN	FP
Monk1(NB)	69.91	44	88
Monk1(IGNB)	71.30	44	86
Monk2(NB)	62.73	125	36
Monk2(IGNB)	63.19	126	33
Monk3(NB)	89.81	44	0
Monk3(IGNB)	90.28	42	0

REFERENCES

[1] Elkan, C., "BOOSTING AND NAIVE BAYESIAN LEARNING". September 1997.

[2] Gunopulos, C.A.R.a.D., "Scaling up the Naive Bayesian Classifier:Using Decision Trees for Feature Selection". in Proceedings of Workshop on Data Cleaning and Preprocessing (DCAP 2002), IEEE International Conference on Data Mining (ICDM 2002).

[3] Zheng, K.M.T.a.Z., "Improving the Performance of Boosting for Naive Bayesian Classification ", in In Proceedings of the PAKDD-99, pp.296-305, Beijing, China. 1999.

[4] Frank, M.H.a.E., "Combining Naive Bayes and Decision Tables", in Association for the Advancement of Artificial Intelligence. 2008.

[5] Khadija Mohammad Al-Aidaroos, A.A.B.a.Z.O., "Naïve Bayes Variants in Classification Learning". IEEE 2010.

[6] m.mitchell, T., "Machine Learning", ed. 16. March 1,1997: McGraw-Hill Science/Engineering/Math; (March 1, 1997).

[7] E. Frank, M.H., and B. Pfahringer, "Locally Weighted Naive Bayes", in In: Proceedings of the Conference on Uncertainty in Artificial Intelligence, 2003: pp. 249–256. Morgan Kaufmann.

[8] Shang, Y.J.a.L., "RoughTree: A Classifier with Naïve-Bayes and Rough Sets Hybrid in Decision Tree Representation". 2007 IEEE International Conference on Granular Computing, 2007: pp. 221-226.

[9] S.B. Kotsiantis, I.D.Z., and P.E. Pintelas, "Machine Learning: A Review of Classification and Combining Techniques". Artificial Intelligence Review, 2006. 26(3): pp. 159-190.
http://dx.doi.org/10.1007/s10462-007-9052-3

[10] Liu, B., "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data". Data-Centric Systems and Applications. 2007: Springer.

[11] R. Abraham, J.B.S., and S.S. Iyengar, "Medical Datamining with a New Algorithm for Feature Selection and Naïve Bayesian Classifier". in ICIT. 2007: IEEE Computer Society, pp.44-49.

[12] Sage, P.L.a.S., "Induction of Selective Bayesian Classifiers", in in Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann. 1994: pp. 399–406.

[13] Webb, F.Z.a.G.I., "Efficient Lazy Elimination for Averaged One-Dependence Estimators", in in Proceedings of the 23rd International Conference on Machine Learning, 2006: pp. 1113-1120.

[14] S.Weiss, C.A.a., "Data Mining with Decision Trees and Decision Rules". Future Generation Computer Systems, 1997. 13:197-210.
Mitchell, T.M., Machine Learning. 1997.
http://dx.doi.org/10.1016/S0167-739X(97)00021-6

[15] Pazzani, M. (1995). Searching for dependencies in Bayesian classifiers. Preliminary Papers of the 5th International Workshop on Artificial Intelligence and Statistics. Ft. Lauderdale, FL.

[16] Pazzani, M. (1996). Constructive Induction of Cartesian Product Attributes. Information, Statistics and Induction in Science. Melbourne, Australia.

[17] Elkan, C. Boosting and Naïve Bayesian Learning. Technical Report No. CS97-557, Department of Computer Science and Engineering, University of California, San Diego, Spetember 1997.

[18] Langley, P. and Sage, S. Induction of Selective Bayesian Classifiers. Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (1994). Seattle, WA: Morgan Kaufmann

[19] Domingos, P. and Pazzani, M. On the Optimality of the Simplie Bayesian Classifier under Zero-One Loss. Machine Larning, 29(2/3): 103-130, November/December 1997.

[20] Irina Rish IBM Research Division, " An empirical study of the naive Bayes classifier ", RC 22230(W0111-014) November2,2001 computer Science.
http://dx.doi.org/10.1023/A:1007413511361

[22] T.M. Cover and J.A. Thomas. Elements of information theory. New York:John Wiley & Sons, 1991