Naïve Bayes Variants in Classification Learning

Khadija Mohammad Al-Aidaroos[1], Azuraliza Abu Bakar[2] and Zalinda Othman[3]
[1,2,3]Faculty of Information and Science Technology
Universiti Kebangsaan Malaysia
Selangor, Malaysia
[1]bintalameen@gmail.com, [2,3]{aab|zalinda}@ftsm.ukm.my

Abstract—**Naïve Bayesian classifier is one of the most effective and efficient classification algorithms. The elegant simplicity and apparent accuracy of naive Bayes (NB) even when the independence assumption is violated, fosters the on-going interest in the model. This paper discusses issues on NB along with its advantages and disadvantages. We also present an overview of NB variants and provide a categorization of those methods based on four dimensions. These include manipulating the set of attributes, allowing interdependencies, employing local learning and adjusting the probabilities by numeric weights. Examples for each category are discussed based on 18 variants reviewed in this paper.**

Keywords- Classification learning; Naïve Bayes (NB) classifier; NB variants.

I. Introduction

Naïve Bayes classifier is a straightforward, frequently used method for supervised learning. Data classification with naïve Bayes is the task of predicting the class of an instance from a set of attributes describing that instance and assume that all the attributes are conditionally independent given the class. The elegant simplicity and apparent accuracy of naive Bayes even when the independence assumption is violated, fosters the on-going interest in the model. It has proven its effective application, in text classification, medical diagnosis and systems performance management [1, 2].

A problem with some NB alternative models, is that, they brought modest improvements at the cost of considerable increase in complexity. However, a variety of adaptations to NB in the literature have been studied in order to improve upon its good performance while maintaining its efficiency and simplicity.

In this paper we investigate issues on naïve Bayes classifier and its variants. We begin with basic background of naïve Bayes, highlighting its features and discussing its main problem. We reviewed several extensions of NB, proposed in the last 15 years, and outlined a categorization of those methods. We compare them in terms of their basic components, approach used and domain of application. However this paper is not intended to provide performance comparison of the different algorithms discussed, since they have different experimental designs and methodologies, which make a direct comparison infeasible. For an empirical comparison of some of the reviewed methods, the reader is referred to [3]. The authors presented a comparison study of eight semi-naïve Bayesian algorithms and performed error analysis using the bias-variance decomposition.

This paper is organized as follows. The background of naive Bayes classifier, including its features and problems, is explained in Section 2. Different extensions to the original NB classifier are categorized in Section 3. Discussion of the reviewed NB variants is presented in Section 4, and finally conclusion is given in Section 5.

II. Naïve Bayes Classifier

Naïve Bayesian classifier, or simply naive Bayes (NB), is one of the most effective and efficient classification algorithms. It is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions.

Given a set of training instances with class labels and a test case E represented by n attribute values (a_1, a_2... a_n), Bayesian classifiers use the following equation to classify E:

$$c(E) = arg_{c \in C} max\, P(c)P(\, a_1, a_2, \dots, a_n|c) \qquad (1)$$

The central assumption of NB classification is that, within each class, the values of attributes are all independent of each other. Then by the laws of independent probability:

$$P(a_1, a_2, \dots a_n|c) = P(a_1|c)P(a_2|c) \dots P(a_n|c)$$

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

By replacing (2) in (1), the resulting classifier is called naive Bayesian classifier, as shown below:

$$c_{NB}(E) = arg_{c \in C}\, max\;\; P(c) \prod_{i=1}^{n} P(a_i|c) \qquad (3)$$

where $c_{NB}(E)$ denotes the classification given by NB on test case E.

From (3), each of the probabilities can be determined directly from the training data. At training time NB generates a one-dimensional table of class probability estimates, indexed by the classes, and a two-dimensional table of conditional attribute-value probability estimates, indexed by the classes and attribute-values.

Although independence is generally a poor assumption, in practice NB often competes well with much more sophisticated techniques and its performance often reported by many researchers as "surprisingly accurate. In a large-scale comparison of naïve Bayes classifier with state-of-the-art algorithms for decision tree induction, instance-based learning, and rule induction, conducted by [4] on standard

978-1-4244-5651-2/10/$26.00 ©2010 IEEE

benchmark datasets; the authors found it to be sometimes superior to the other learning schemes, even on datasets with substantial feature dependencies.

A. *Features of NB*

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. In this section we concentrate more on the features of this classifier.

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].

B. *Problems with NB*

NB is extremely effective in practice and difficult to improve upon, however NB can suffer from oversensitivity to redundant or irrelevant attributes. 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 [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.

Two more issues related to NB classifier need to be mentioned here, namely, zero probabilities and continuous attributes [4, 8]. We discuss the former in this section while the latter is discussed in detail in the next section.

It is possible that a particular attribute value in the test set never occurs together with a class in the training set. This is problematic because it will result in a zero probability, which wipes out all the other probabilities when they are multiplied according to (3). A principled solution to this problem is to incorporate a small-sample correction into all probabilities to avoid zero counts.

A typical approach is to use the Laplace estimate. If n_{ijk} is the number of instances that have both class c_i and attribute value a_{jk}. Let n_i be the number of instances with class c_i, and n_j is the number of values of attribute A_j, then the following is the corrected estimate:

$$P(A_j = a_{jk} | C = c_i) = \frac{n_{ijk} + f}{n_i + f n_j} \qquad (4)$$

Where f is a multiplicative factor, which is commonly set to $f = 1/n$, where n is the total number of training instances.

C. *Discretization for NB Classifier*

In real life data sets there are both qualitative (nominal) and quantitative (numeric) attributes and each type has different method to estimate probabilities required for NB classification. For a qualitative attribute, its probabilities can be estimated from corresponding frequencies. For a quantitative attribute, either probability density estimation or discretization can be employed to estimate its probabilities [16].

Research study shows that NB classification works best for discretized attributes and discretization effectively approximates a continuous variable [16-18]. Dougherty J. et al. found that the performance of NB algorithm significantly improved when features were discretized using an entropy-based method as compared to using probability density estimation. Another comparison study conducted by Abraham R. et al. indicates that, on an average, MDL (Minimum Description Length) discretization seems to be the best performer compared to that of EWD (Equal Width) and EFD (Equal Frequency) discretizations.

However, because qualitative data have a lower level of measurement scale than quantitative data, discretization might suffer information loss. Based on this viewpoint, some researchers argue that the more continuous attributes used to predict, the more information to be lost by pre-discretization. In [19] the authors propose Self-adaptive NBTree which can mitigate the negative effect of information loss by applying post-discretization strategy.

Typical approaches to estimate probability density estimation are assuming a Gaussian distribution or kernel density estimation. On the other hand, discretization did not make assumptions about the forms of quantitative attributes' probability distribution. Therefore, in practice, discretization is more popular than probability density estimation and many of the researchers proposing new variants of NB prepare their data by pre-discretization. For example, if we refer to the NB variants listed in Table 1, most of the researchers used the MDL discretization, given by [20], to discretize their data as a pre-processing step.

III. VARIANTS OF NB CLASSIFIER

The basic independent Bayes model has been modified in various ways and new algorithms have been proposed in attempts to improve its performance. In this paper we review the following variants of NB: SNB (Selective Naïve Bayes) [13], KDB (K-Dependence Bayesian classifier) [21], NBTree [9], TAN (Tree Augmented Naïve Bayes) [1], LBR (Lazy Bayesian Rule) [22], SBC (Selective Bayesian Classifier) [14], HLBR (Heuristic LBR) [23], SNNB (Selective Neighborhood based NB) [12], LWNB (Locally Weighted NB) [5], WNB (Weighted NB) [24], LNB (Lazy NB) [25], AODE (Averaged One-Dependence Estimators) [26], LE-AODE (Lazy Elimination for AODE) [15], S-NBTree (Self-adaptive NBTree) [19], DKNAW (Dynamic K-Nearest-Neighbor NB with Attribute Weighted) [27], BPET (Bayesian estimate with Probability Estimation Trees) [28], RoughTree [6] and AWNB (Attribute Weighted NB) [29].

The above mentioned studies have shown that it is possible to improve upon the general performance of NB classifier and they have extended it using different approaches and combining different techniques. For example the NB variant presented by [1], attempts to overcome the independence assumption by adding extra edges to include some of the dependencies between the features with the limitation that each feature can be related to only one other feature. From a different perspective, an alternative generalization, SBC, has been explored by [14]. The idea was to improve classification performance by using attributes that appear in only the top three levels of a decision tree and thus removing the irrelevant features. For more details on each of the mentioned variants, the reader can refer to the corresponding papers as shown in Table 1.

Based on the research work to improve NB classifier, those approaches can be broadly divided into four groups:

1. Manipulating the set of attributes.
2. Allowing interdependencies between attributes.
3. Employing the principle of local learning.
4. Adjusting the probabilities by numeric weights.

In the following subsections we discuss each of the four groups in more detail with examples from the studies listed in Table 1.

A. Manipulating the Set of Attributes

First group involves those methods which manipulate the set of attributes by performing some sort of feature subset selection or extraction, to remove or join attributes, and then applying NB with the new attribute set.

Given a set of attributes $a = \{a_1, a_2 ... a_n\}$, the proposed algorithms under this category would usually have two groups of attributes which are derived from the original set of attributes. The selected attributes $s = \{s_1, s_2 ... s_g\}$ and remaining attributes $r = \{r_1, r_2 ... r_{n-g}\}$. Then the algorithm would select the class with maximum posterior probability, to classify a new test case, as shown below:

$$arg_{c_i} \max\left(P(c_i|s)\prod_{j=1}^{n-g}P(r_j|c_i,s)\right) \tag{5}$$

For example, in LBR the set s denotes the selected attributes in the antecedent of the rule, while in NBTree, s denotes the set of test attributes on the path to the leaf of NBTree.

Other approaches such as SBC, discussed above, would simply remove those redundant or irrelevant attributes, using decision tree, and apply NB to only the remaining attributes. The selective naïve Bayesian classifier [13] is also a straightforward application of feature subset selection and it classifies a test instance using the following formula ($k \leq n$ is the number of selected attributes):

$$c_{NB}(E) = \arg_{c\in C}\max\ P(c)\prod_{i=1}^{k}P(a_i|c) \tag{6}$$

B. Allowing Interdependencies between Attributes

Second group of research improves NB by addressing its main weakness explicitly and therefore allowing interdependencies between attributes. This covers the *n-dependence* Bayesian classifiers, where $n \geq 1$. The original NB is *0-dependence* classifier because its attributes only depend on the class attribute and no other attributes.

Most classifiers which come under this category are based on Bayesian networks. Reference [21] presented KDB algorithm, that is used to construct *k-dependence* Bayesian classifier (KDB) which contains the structure of the NB and allows each attribute A_i to have a maximum of k attribute nodes as parents.

However, to maintain efficiency it appears desirable for many researchers to restrict their proposed methods as being *1-dependence* classifiers. The Tree-Augmented NB (TAN), proposed by [1], allows each attribute to depend on the class and on at most one other attribute, and therefore is referred as a *1-dependence* classifier. LBR [22] and AODE [26] are also *1-dependence* classifiers.

C. Employing the Principle of Local Learning

This group employs the principle of local learning to extend NB and constructs a number of NB classifiers on different subsets of training instances. This can improve its performance in case the conditional independence assumption is violated on the whole training data but is satisfied within each local NB.

So the key point here is to divide the whole data set into subgroups which hold the independence assumption (to a certain level) and build local NB on each subset. In this case, it's common to have local NB classifiers embedded into another model, such as decision tree. A typical example in this category is the NBTree, which partitions the whole instance space into several subspaces, using decision tree techniques, and then trains a local NB classifier for each leaf node [9]. RoughTree also produces a tree-like model, but unlike NBTree, it uses the attribute dependence measure (based on rough sets) as its splitting criteria to alleviate the interdependences of the local NB at each leaf [6].

Other than decision trees, k-nearest neighbour has also been embedded with NB for local learning. SNNB [12] constructs multiple NB classifiers on multiple neighbourhoods by using different radius values for the input new object and then selects the most accurate one to classify the new object. In [5] the LWNB uses the nearest neighbours of the test instance to build a local NB. Although LWNB is relatively insensitive to the choice of k but it's still considered as a k-related algorithm.

D. *Adjusting the Probabilities by Attribute Weights*

Unlike the first group of NB classifiers, this group alters NB through adjusting the probabilities by assigning numeric weights to the attributes. These weights can correct incorrect ranking of the conditional probabilities produced by the NB classifier due to certain violations of the attribute independence assumption.

However, if we compare attribute weighting to attribute selection, which is a special case of the former with weights zero or one, it is clear that assigning different weights to the attributes based on their contribution can help to improve the final model. So, instead of removing some attributes, which may cause information loss, it is more advantage to retain them but reduce their affect to the final model with lower weights.

Naïve Bayes with attributes weighted differently results in a model called weighted naïve Bayes (WNB) [24], which is formally defined as follow:

$$c_{WNB}(E) = arg_{c\in C} \max P(c) \prod_{i=1}^{n} P(a_i|c)^{w_i} \qquad (7)$$

Where $c_{WNB}(E)$ denotes the classification given by the WNB for the test instance E, and w_i is the weight of attribute A_i.

Based on the various variants of NB listed in Table 1, we can find a number of studies which come under this group. For example, [24] explored various methods to learn WNBs (such as gain ratio, hill climbing and Markov Chain Monte Carlo methods). Another study which comes under this group it the Attribute Weighted Bayesian classifier (AWNB), presented by [29], which used decision trees to estimate the weights of attributes based on the minimum depth at which each attribute is tested in the tree.

However, it is important here to distinguish attribute weighting from instance weighting. While the above mentioned weighted NB classifiers are meant for those setting weights for each attribute; other approaches may apply weighting scheme for instances instead of attributes. LWNB model uses a weighted set of training instances in the locale of the test instance [5].

TABLE I. VARIANTS OF NAÏVE BAYES CLASSIFIER

Author(s), year	**Bayesian Classifier**	**Techniques/ components**	**Approach**	**Evaluation Criteria**	**Domain of Application (data sets)**
Langley P. & Sage S., 1994	SNB	Naïve Bayes with Forward Selection	Eager	Compared with C4.5 and NB - using accuracy	6 datasets from UCI Repository.
Sahami M., 1996	KDB	Bayesian classifier with features dependency	Eager	Compared with NB - using accuracy	5 datasets from UCI Repository and a text classification domain.
Kohavi R., 1996	NBTree	Hybrid of Decision Tree and Naïve Bayes	Eager	Compared with C4.5 and NB - using accuracy	10 datasets from UCI Repository.
Friedman N. et al., 1997	TAN	Restricted Bayesian Network	Eager	Compared with C4.5, NB and Selective-NB. - using accuracy	25 datasets from UCI Repository.
Zheng Z. & Webb G., 2000	LBR	Lazy learning with Bayesian Tree Induction	Lazy	Compared with NB, C4.5, BT, BSEJ, BSE and LazyTree - using error rate	29 datasets from UCI Repository.
Ratanamahatana C. & Gunopulos D., 2002	SBC	Decision Tree and Naïve Bayes	Eager	Compared with C4.5, NB and ABC - using accuracy & mean elapsed time	10 data sets from UCI Repository and one large synthetic data set
Wang Z. & Webb G., 2002	HLBR	LBR and TAN	Combines eager & lazy learning	Compared with NB, LBR and TAN - using error rate and runtime	35 data sets from UCI Repository
Xie Z. et al., 2002	SNNB	NB and k-NN	Lazy	Compared with NB, NBTree, CBA and C4.5 - using error rate	26 data sets from UCI Repository
Frank E. et al., 2003	LWNB	Locally weighted NB(using k-NN) with lazy approach	Lazy	Compared with K-NN, KNNDW and NB - using accuracy	2 artificial datasets and 37 UCI datasets
Zhang H. & Sheng S., 2004	WNB	NB with Gain Ratio, Hill Climbing and Markov Chain Monte Carlo	Eager	Compared each variant of WNB with NB and C4.5 - using accuracy & AUC	8 data sets from UCI Repository
Jiang L. & Guo Y., 2005	LNB	Lazy learning with NB and Ranking	Lazy	Compared with C4.4 and NB - using AUC	36 data sets from UCI Repository
Webb G. et al., 2005	AODE	Averaging a constrained class of classifiers	Eager	Compared with NB, ODE, bagODE, LBR, TAN, SP-TAN and C4.5 -using error, bias, variance, training time & classification time	Using 37 data sets

Zheng F. & Webb G., 2006	LE-AODE	AODE with Lazy Elimination	Combines eager & lazy learning	Compared with AODE, LE-AODE and BSE-AODE - using error, bias & variance	56 data sets from UCI Repository
Wang L.-M. et al., 2006	S-NBTree	Hybrid of DT and NB	Eager	Compared with C4.5 and NBTree - using accuracy	12 data sets from UCI Repository
Jiang L. et al., 2006	DKNAW	NB, k-NN and Attribute Weighting	Combines eager & lazy learning	Compared with NB, LWNB, k-NN, KNNDW - using accuracy	36 data sets from UCI Repository
Qin Z., 2006	BPET & FPET	Naïve Bayes and PET (Probability Estimation Trees)	Eager	Compared with PET and NB - using accuracy	9 data sets from UCI Repository
Ji Y. & Shang L., 2007	RoughTree	Semi-NB, DT and Rough Sets	Eager	Compared with NB, C4.5 and NBTree - using accuracy	13 data sets from UCI Repository
Hall M., 2007	AWNB	NB, DT and Attribute Weighting	Eager	Compared with NB, GRW, RW, CFS, SB,SBC and NBTree - using RRSE and AUC	28 data sets from UCI Repository

IV. DISCUSSION

From a different perspective, and depending on when the major computation occurs; the discussed methods can be categorized into two groups: eager classifiers, e.g. [1, 6, 9, 26], and lazy classifiers, e.g. [12, 22, 25]. However, some approaches, such as HLBR [23] and DKNAW [27] combine both eager and lazy learning.

However, some of the NB variants may combine more than one approach at the same time. For example, most of the Bayesian classifiers based on (5), i.e. manipulating the set of attributes, also employ local learning on the remaining attributes. Therefore, we mentioned NBTree as an example in the first group as well as in the third group.

For improving NB, based on the reviewed variants, many researchers have paid considerable attention to investigate the use of decision tree and k-NN (more than half of the listed methods in Table 1 make use of DT or k-NN). Other techniques from other disciplines, such as soft computing, still need more investigation.

Rough Set Theory introduced by Pawlak in 1982, which is a mathematical tool that deals with imprecision and uncertainty of information, has the ability to discover attributes' dependency [30, 31]. However, RoughTree is the only Bayesian classifier which makes use of rough sets to improve NB and this encourages for more research in this area. Thus investigating the use of rough set to improve NB's performance is our main direction for future work.

V. CONCLUSION

Naïve Bayesian classifier is one of the most effective and efficient classification algorithms based on applying Bayes' 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. In this paper we discussed the main features as well as the problems and issues related to NB classifier. We reviewed several methods for improving NB and outlined a categorization framework for the different approaches based on four dimensions. We discussed each dimension and described examples based on the 18 NB variants reviewed in this study.

REFERENCES

[1] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian Network Classifiers," Machine Learning, 1997. 29(2-3): pp. 131-163.

[2] I. Rish, "An Empirical Study of the Naïve Bayes Classifier," In: Proceedings of the Int. Joint Conf. on Artificial Intelligence, Workshop on "Empirical Methods in AI", 2001.

[3] F. Zheng and G. I. Webb, "A Comparative Study of Semi-naïve Bayes Methods in Classification Learning," Proceeding 4th Australasian Data Mining conference (AusDM05), 2005: pp. 141-156.

[4] P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," Machine Learning, 1997. 29(2-3): pp. 103-130.

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

[6] Y. Ji and L. Shang, "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.

[7] S.B. Kotsiantis, I.D. Zaharakis, and P.E. Pintelas, "Machine Learning: A Review of Classification and Combining Techniques," Artificial Intelligence Review, 2006. 26(3): pp. 159-190.

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

[9] R. Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid," in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. 1996: pp. 202–207.

[10] I. Zelič, I. Kononenko, N. Lavrač, and V. Vuga, "Induction of Decision Trees and Bayesian Classification Applied to Diagnosis of Sport Injuries," Journal of Medical Systems, 1997. 21(6): pp. 429-444.

[11] R. Abraham, J.B. Simha, 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] Z. Xie, W. Hsu, Z. Liu, and M. Lee, "SNNB: A Selective Neighborhood Based Naive Bayes for Lazy Learning," Lecture Notes in Computer Science, 2002: pp. 104–114.

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

[14] C.A. Ratanamahatana and D. Gunopulos, "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).

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

[16] Y. Yang and G. I. Webb, "On why discretization works for naive-Bayes classifiers," Lecture Notes in Computer Science, 2003: pp. 440–452.

[17] R. Abraham, J.B. Simha and S.S. Iyengar, "A Comparative Analysis of Discretization Methods for Medical Data Mining with Naïve Bayesian Classifier," in 9th International Conference on Information Technology (ICIT'06), 2006: pp. 235-236.

[18] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features," in Proc. 12th International Conference on Machine Learning. 1995: Morgan Kaufmann.

[19] L.-M. Wang, X.-L. Li, C.-H. Cao and S.-M. Yuan, "Combining Decision Tree and Naïve Bayes for Classification," Knowledge_Based Systems, 2006. 19(7): pp. 511-515.

[20] U. M. Fayyad and K. B. Irani, "Multi-interval Discretization of Continuous-valued Attributes for Classification Learning," in Proceedings of the 13th International Joint Conference on Artificial Intelligence, 1993: pp. 1022–1027.

[21] M. Sahami, "Learning Limited Dependence Bayesian Classifiers," in KDD-96 Proceedings, AAAI Press. 1996: pp. 335–338.

[22] Z. Zheng and G.I. Webb, "Lazy Learning of Bayesian Rules," Machine Learning, 2000. 41(1): pp. 53–84.

[23] Z. Wang and G. I. Webb, "A Heuristic Lazy Bayesian Rule Algorithm," The Australasian Data Mining Workshop, 2002.

[24] H. Zhang and S. Sheng, "Learning Weighted Naïve Bayes with Accurate Ranking," in Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04), 2004.

[25] L. Jiang and Y. Guo, "Learning Lazy Naive Bayesian Classifiers for Ranking," in Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence, ICTAI. 2005: pp. 412–416.

[26] G.I. Webb, J. Boughton, and Z. Wang, "Not so Naive Bayes: Aggregating One-Dependence Estimators," Machine Learning, 2005. 58(1): pp. 5–24.

[27] L. Jiang, H. Zhang and Z. Cai, "Dynamic K-Nearest-Neighbor Naive Bayes with Attribute Weighted," in FSKD. LNCS, 2006: Springer. pp. 365–368.

[28] Z. Qin, "Naïve Bayes Classification Given Probability Estimation Trees," in Proceedings of the 5th International Conference on Machine Learning and Applications (ICMLA'06), 2006.

[29] M. Hall, "A Decision Tree-Based Attribute Weighting Filtering for Naïve Bayes," Knowledge-Based Systems, 2007. 20(2): pp. 120-126.

[30] Z. Pawlak, "Rough Sets and Intelligent Data Analysis," Information Sciences, Elsevier. 2002: pp. 1-12.

[31] A. Skowron, "Rough Sets in KDD," in Proceedings of the 16th World Computer Congress (IFIP'2000). 2000: pp. 1-14.