EECS E6690 Statistical Learning

Final Project Report

Yingling Wang (yw3152) Xin Geng (xg2294) Anran Li (al3804)

Instructor: Professor Predrag Jelenkovic

Abstract

In this project, we implement data classification using different methods. The problem we consider is doing data classification to achieve high accuracy. We show and compare classification results for data using different approaches such as Naïve Bayesian Classifier, C4.5 Decision Trees and Selective Bayesian Classifier. They are all mentioned in paper *Scaling up the Naive Bayesian Classifier: Using Decision Trees for Feature Selection.* On the basis of the analysis of these three approaches, we propose a new approach named as Improved Selective Bayesian Classifier, which takes advantages of Decision Trees and Selective Bayesian Classifier. In this project, we access data classification quality based on prediction accuracy and running time.

Keywords: Naïve Bayesian Classifier, Selective Bayesian Classifier, Feature Selection, C4.5 Decision Trees

1. Introduction

Main structure of this project:

- Paper and Data Details
- Main Classification Tools
  - Naïve Bayesian Classifier
  - ID3 Algorithm
  - C4.5 Decision Trees
  - Selective Bayesian Classifier
- Result Reproduction
- Improved Selective Bayesian Classifier
- Conclusion and Discussion

Among many classification tools, Naïve Bayesian Classifier (NBC) and C4.5 Decision Trees are the two most widely used ones. C4.5 constructs decision trees by using features to try and split the training set into positive and negative examples until it achieves high accuracy on the training set. NB represents each class with a probabilistic summary, and finds the most likely class for each example it is asked to classify. Naïve Bayesian model originated from classical mathematical theory, which has a solid mathematical foundation and stable classification efficiency. At the same time, NBC model needs few parameters to estimate, it is not sensitive to missing data, and the algorithm is relatively simple. Theoretically, NBC model has the smallest error rate compared with other classification methods. But this is not always the case, the NBC model assumes that attributes are independent of each other. This assumption is often unsuccessful in practical applications, which has a certain impact on the correct classification of NBC model. When the number of attributes is large or the correlation between attributes is large, the classification efficiency of NBC model is lower than that of decision tree model. The NBC model has the best performance when the correlation of attributes is small. This is the reason why in practice, Naïve Bayesian Classifier (NBC) works very well on some domains, and poorly on some. The performance of NB suffers in domains which involve correlated features. C4.5 Decision Trees, on the other hand, typically perform better than the Naïve Bayesian algorithm on such domains.

For small-scale data set, Naïve Bayesian Model can handle multi-classification tasks and perform well. It is suitable for incremental training, especially when the amount of data exceeds internal memory, we can do incremental training in batches. The algorithm of Naïve Bayesian Model is relatively simple. It is often used in text categorization. The main shortcoming of Naïve Bayesian Classifier is that it 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. C4.5 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, this is obvious since this would lead to exactly the same split, which makes no difference to the existing tree. This is one of the main reasons C4.5 outperform Naïve Bayesian Classifier on domains with correlation attributes.

In order to overcome this weak point of Naïve Bayesian Classifier, Ratanamahana and Gunopulos propose Selective Bayesian Classifier, which is an improvement of Naïve Bayesian Classifier. They conjecture that the performance of Naïve Bayesian Classifier will improve if it only uses those features that C4.5 would use in its decision tree when learning a small example of a training set, then it is a combination of the two different natures of the classifiers. They conduct this approach on 9 datasets of the UCI repository, 5 of which C4.5 achieves higher accuracy than NBC, and 4 on which NBC outperforms C4.5 and the result is quite satisfying. Their SBC approach outperforms NBC in each of domains and outperforms C4.5 Decision Trees in over half of the domains.

However, when we tried to apply Selective Bayesian Classifier approach on the 9 domains, we did not get the results as expected. It turned out that Selective Bayesian Classifier outperforms Naïve Bayesian Classifier in only 4 domains, and perform the same as Naïve Bayesian Classifier in 2 domains. In these domains C4.5 Decision Trees perform obviously better than Naïve Bayesian Classifier.

Due to unsatisfying results we reproduce in the way proposed in the paper, we decide to analyze the reasons and come up with a new approach on the basis of Naïve Bayesian Classifier. We think the feature selection process in this Selective Bayesian Classifier is not appropriate enough. It runs C4.5 Decision Trees on a 10% sample of a training set and selects a set of attributes that appear only in the first 3 levels of the simplified decision tree as relevant. Then repeat this process for 5 times. When we do this feature selection process on a dataset for 5 times, the sets of attributes in the first 3 levels each time are different from each other. In one hand, given the size of the datasets, the size of a 10% sample is so small, which easily leads to the loss of ubiquity. The attributes after the third level of a simplified decision are likely to contribute to the final decision.

In this project, we propose a new approach called Improved Selective Bayesian Classifier, where we mainly improve the feature selection part. We compute Information Gain Ratio at the root node of a decision tree and select features with highest ratios. In order to avoid suffering from redundant attributes, we compute Pearson Correlation Coefficient and draw heat graphs out. We remove the irrelevant attributes and then apply Naïve Bayesian Classifier. We present experimental evidence that this method of feature selection leads to improved performance of Naïve Bayesian Classifier, and improves performance of C4.5 to some extent.

2. Paper and Data Details

The paper *Scaling up the Naive Bayesian Classifier: Using Decision Trees for Feature Selection*[1] was published by Chotirat Ann Ratanamahatana Dimitrios Gunopulos in Computer Science Department, University of California. It used 9 domains from the UCI repository, 5 of which Naïve Bayesian classifier outperforms C4.5 and the other 4 of which C4.5 outperforms Naïve Bayesian classifier. The datasets are shown in Table 1.

Table 1. Descriptions of domains used

Dataset	#Attributers	#Classes	#Instances
Ecoli	8	8	336
GermanCredit	20	2	1,000
KrVsKp	37	2	3,198
Monk	6	2	554
Mushroom	22	2	8,124
Pima	8	2	768
Promoter	57	2	106
Soybean	35	19	307
Wisconsin	9	2	699
Vote	16	2	435

The paper analyzed pros and cons of Naïve Bayesian Classifier and C4.5 Decision Trees and proposed Selective Bayesian Classifier. A simple method that uses C4.5 decision trees to select features has been described. Figure 1 shows the algorithm for feature selecting of the Selective Bayesian classifier. Note that they apply tree pruning in C4.5 Decision Trees.

1. Shuffle the training data and take a 10% sample.
2. Run C4.5 on data from step 1.
3. Select a set of attributes that appear only in the first 3 levels of the simplified decision tree as *relevant* features.
4. Repeat 5 times (step 1-3)
5. Form a union of all the attributes from the 5 rounds.
6. Run Naïve Bayesian classifier on the training and test data using *only* the final features selected in step 5.

Figure 1. Selective Bayesian Classifier Algorithm: Feature Selecting

In order to compare the quality of these three approaches, the paper conduct Naïve Bayesian Classifier, C4.5 Decision Trees and Selective Bayesian on the 9 domains. They first shuffle each dataset and then produce disjoint training. Then, run NBC, C4.5

and SBC on the datasets. In the paper, the results confirm their hypotheses. The asymptotic accuracy of SBC is as good as (or slightly better than) the better of C4.5 and NB on each of the domains.

They also show the result for NBC, C4.5, and SBC learning algorithms using 80% of the data for training and 20% for testing (5-fold cross- validation). SBC outperforms the original NBC in nearly every domain, giving the accuracy improvement up to 7.9%. SBC also outperforms both C4.5 in almost all the domains, giving the accuracy improvement up to 33.1%. Even though, SBC cannot beat C4.5 in some datasets, it still gives quite big improvement over the Naïve Bayes (7.8%, 1.4%, and 6.0%) on such cases.

As for the number of features selected for Selective Bayesian classifier. On almost all the datasets, more than half of the original attributes were eliminated. Finally, they analyze the running times of SBC, NBC and that of C4.5 because Bayesian classifier only needs to go through the whole training data once. They are also space efficient because they build up a frequency table in size of the product of the number of attributes, number of class values, and the number of values per attribute. [2].

3. Main Classification Tools

3.1 Naïve Bayesian Classifier

The Bayesian School is very old, but it was not the mainstream from its birth to a hundred years ago. The mainstream is the frequency school. Both Pearson and Fisher, the authority of the Frequency School, disdained the Bayesian School, but the Bayesian School has won its high status by virtue of its outstanding application in modern specific fields. The idea of Bayesian school can be summarized as prior probability + data = posterior probability. That is to say, the posterior probability we need in practical problems can be synthesized by the prior probability and data. It is easy to understand that the data are attacked by the frequency school by the prior probability. Generally speaking, the prior probability is our historical experience in the field of data, but this experience is often difficult to quantify or model. So the Bayesian school boldly assumes the model of prior distribution, such as normal distribution, beta distribution and so on. This hypothesis has no specific basis, so it has been considered ridiculous by the frequency school. Although it is difficult to deduce the logic of Bayesian School from the rigorous mathematical logic, in many practical applications, Bayesian theory is very useful, such as spam classification, text classification. Before we can understand the Naive Bayesian algorithm, we need to know the necessary statistical knowledge.

Before we can understand the Naive Bayesian algorithm, we need to review the necessary statistical knowledge. If X and Y are independent of each other, $P(X,Y) = P(X)P(Y)$. Formula of conditional probability is $P(Y|X) = P(X|Y)P(Y)/P(X)$. Formula of full probability is $\sum_k P(X|Y = Y_k)P(Y_k)$, where $\sum_k P(Y_k) = 1$. The

formula for Bayes Rule is

$$P(Y_k|X) = \frac{P(X|Y_k)P(Y_k)}{P(X|Y=Y_k)P(Y_k)}.$$

Say our sample for classification model is

$$(x_1^{(1)}, x_2^{(1)}, \ldots x_n^{(1)}, y_1) \ , (x_1^{(2)}, x_2^{(2)}, \ldots x_n^{(2)}, y_2), \ldots (x_1^{(m)}, x_2^{(m)}, \ldots x_n^{(m)}, y_n)$$

It means that we have m samples, each sample has n attributes, the output of attributes have K classes, defined as $C_1, C_2, ..., C_k$. From the sample wo can learn the prior distribution of Naïve Bayesian $P(Y = C_k)(k = 1, 2, ...K)$, and then we can get joint distribution P(X,Y). Joint distribution P(X,Y) is defined as:

$$P(X, Y = C_k) = P(Y = C_k)P(X = x \mid Y = C_k) = P(Y = C_k)P(X_1 = x_1, X_2 = x_2, ...X_n = x_n \mid Y = C_k)$$

From the above formula, we can see *P(Y=Ck)* is easy to obtain through the Maximum Likelihood Method, which is the times C_k appears in class G.

But $P(X_1 = x_1, X_2 = x_2, ...X_n = x_n \mid Y = C_k)$ is hard to obtain here, for this is a super-complex conditional distribution with n dimensions. The naive Bayesian model here makes a bold assumption that the N dimensions of X are independent of each other, so that it can be concluded that:

$$P(X_1 = x_1, X_2 = x_2, ...X_n = x_n \mid Y = C_k) = P(X_1 = x_1 \mid Y = C_k)P(X_2 = x_2 \mid Y = C_k)...P(X_n = x_n \mid Y = C_k)$$

As can be seen from the above formula, this difficult conditional distribution is greatly simplified, but it may also lead to inaccuracy of prediction. If the attributes really very dependent, it's better to consider other classification methods. But in general, the condition that the features of samples are independent is weakly established, especially when the amount of data is very large. Although we sacrifice accuracy, the advantage is that the calculation of conditional distribution of the model is greatly simplified, which is the choice of Bayesian model. Our goal is that given a new data $(x_1^{(test)}, x_2^{(test)}, ...x_n^{(test)})$, we determine which class it belongs to. Since it is a Bayesian model, the posterior probability maximization is used to judge classification. We only need to calculate all k conditional probabilities $P(Y = C_k \mid X = X^{(test)})$, then find out the category corresponding to the maximum conditional probability, which is Naive Bayesian prediction.

3.2 ID3 Algorithm

Decision tree is a basic classification and regression method. Decision tree model is a tree structure. In classification problem, it represents the process of case classification based on features. It can be considered as a set of if-then rules or a conditional probability distribution defined in feature space and class space. Its main advantage is that the model has readability and fast classification speed. In learning, a decision tree model is established based on the principle of minimizing loss function using training data. When forecasting, the new data are classified by decision tree model. Decision tree learning usually consists of three steps: feature selection, decision tree generation and decision tree pruning. These decision tree learning ideas are mainly derived from the ID3 algorithm proposed by Quinlan in 1986, the C4.5 algorithm proposed in 1993, and the CART algorithm proposed by Breiman et al. in 1984.

The core of ID3 algorithm is to construct decision tree recursively by using information gain criterion to select features at each node of decision tree. Specific methods are as follows: starting from the root node, calculating the information gain of all possible features for the node, selecting the feature with the greatest information gain as the feature of the node, and establishing the sub-node according to the different values of the feature; then recursively invoking the above method for the sub-node to construct the decision tree; until the information gain of all features is very small or no feature can be selected. Finally, we get a decision tree, ID3 is equivalent to using maximum likelihood method to select probability model.

3.2.1 Information Gain

Before we can understand *information gain*, we need to know another related term *entropy*. In general, entropy is a measure of the purity in an arbitrary collection of examples. Let S be a set consisting of s data samples. Suppose the class label attribute has m distinct values defining m distinct classes, C_k. Let s_i be the number of samples of S in class C_k. The expected information needed to classify a given sample is given by

$$I(s_1, s_2, ..., s_m) = -\sum_{k=1}^{m} p_k \log_2(p_k),$$

where p_k is the probability that an arbitrary sample belongs to class C_k and is estimated by s_k / s.

Let attribute A have v distinct values, $\{a_1, a_2, ..., a_v\}$. Attribute A can be used to partition S into v subsets, $\{S_1, S_2, ..., S_v\}$, where S_j contains those samples in S that have value a_j of A. Let s_{kj} be the number of samples of class C_k in a subset S_j. The entropy, or expected information based on the partitioning into subsets by A, is given by

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + ... + s_{mj}}{s} I(s_{1j}, ..., s_{mj}).$$

The term acts as the weight of the j^{th} subset and is the number of samples in the subset

divided by the total number of samples in S. For a given subset, S_j,

$$I(s_{1j}, s_{2j}, ..., s_{mj}) = -\sum_{k=1}^{m} p_{kj} \log_2(p_{kj})$$

where $p_{kj} = s_{kj} / |S_j|$ and is the probability that a sample in Sj belongs to class C_k. The entropy is zero when the sample is pure, i.e. when all the examples in the sample S belong to one class. Entropy has a maximum value of 1 when the sample is maximally impure, i.e. there are same proportions of positive and negative examples in the sample S.

The encoding information would be gained by branching on *A* is

InformationGain(A) = I(s1,s2,...,sm) – E(A)

The attribute with the highest information gain is chosen as the test attribute for the current node. Such approach minimizes the expected number of tests needed to classify an object and guarantees that a simple (but may not be the simplest) tree is found.

3.2.2 Algorithm

Input: Training Data Set D, Feature Set A, Threshold Euro

Output: Decision Tree T

(1) If all instances in D belong to the same class of C_k, then T is a single-node tree, and class C_k as the class marker of the node, return T.

(2) If A=∅, then T is a single node tree and the class C_k with the largest number of instances in D as the class marker of the node, return T.

(3) Otherwise, calculate the *information gain* of each feature in A to D, and select the feature A_g with the greatest information gain.

(4) If the information gain of A_g is less than the threshold, then T is set as a single node tree and the class C_k with the largest number of instances in D is set as the class marker of the node, return T.

(5) Otherwise, for Ag each possible value a_i, according to $A_g=a_i$, Divide D into several non-empty subsets D_i, D_i The classes with the largest number of instances are used as markers to construct sub-nodes, which form tree T and return T.

(6) For the first sub-node, D_i for training set, take A{A_g} For feature set, step (1) ~ (5) is called recursively to get subtree T_i. return T_i .

3.3 C4.5 Decision Trees

C4.5 algorithm is one important data mining algorithms. It is an improvement of ID3

algorithm. Compared with ID3 algorithm, it has the following improvements.

(1) Selecting attributes by information gain ratio

(2) pruning trees in the process of constructing decision trees

(3) Non-discrete data can also be processed.

(4) Ability to process incomplete data

3.3.1 Information Gain Ratio

The information gain measure is biased in that it tends to prefer attributes with many values rather than those with few values. C4.5 suppresses this bias by using an alternative measure called Information Gain Ratio, which considers the probability of each attribute value. The Split Information takes into account the factor of an attribute having many values. *SplitInformation* and *GainRatio* are defined as

$$SplitInformation(A) = -\sum_{j=1}^{v} \frac{|S_j|}{|S|} \log_2 \frac{|S_j|}{|S|}$$

$$GainRatio(A) = \frac{InformationGain(A)}{SplitInformation(A)}$$

By using *SplitInformation(A)*, which is proportional to the number of values an attribute A can take, *GainRatio(A)* effectively removes the bias of information gain towards features with many values. To resolve the issue when *SplitInformation(A)* becomes very small, C4.5 lists the set of attributes with the *InformationGain(A)* above the average information gain for that node and then it uses the Gain Ratio to select the best attribute from the list .[3].

3.3.2 Tree Pruning

C4.5 builds a tree so that most of the training examples are classified correctly. Though this approach is correct when there is no noise, accuracy for unseen data might degrade in cases where there is a lot of noise associated with the training examples and/or the number of training examples is very small. To alleviate this so-called overfitting problem, C4.5 uses the post-pruning method. This approach allows C4.5 to grow a complete decision tree first, and then post-prune the tree. It tries to shorten the tree in order to overcome overfitting. This generally involves removal of some of the nodes or subtrees from the original decision tree. Its goal is to improve (by pruning) the accuracy on the unseen set of examples.

As a result, C4.5 achieves further elimination of features through pruning. It uses rule-post pruning to remove some of the insignificant nodes (and hence, some not so relevant features) from the tree.

3.4 Selective Bayesian Classifier

3.4.1 Description

The features that C4.5 chose in constructing its decision tree are likely to be the ones that are most descriptive in terms of the classifier, in spite of the fact that a tree structure inherently incorporates dependencies among attributes, while Naïve Bayesian works on a conditional independence assumption. C4.5 will naturally construct a tree that does not have an overly complicated branching structure if it does not have too many examples that need to be learned. As the number of training examples increases, the attributes that are considered will usually be the ones that are not correlated. This is mainly because C4.5 will use only one of a set of correlated features for making good splits in training set. However, sometimes many of the branches may reflect noise or outliers (overfitting) in the training data. "Tree pruning" procedure in C4.5 attempts to identify and remove those least reliable branches, with the goal of improving classification accuracy on unseen data. SBC only picks attributes contained in the first few levels of the tree as the most representative attributes. This is supported by the fact that by the selection of attributes that split the data in the best possible way at every node, C4.5 will try to ensure that it encounters a leaf at the very earliest possible point, i.e. it prefers to construct shorter trees. And by its algorithm, C4.5 will find trees that have attributes with higher information gain nearer to the root.

3.4.2 Algorithm

SBC first shuffles the training data and use 10% of that to run C4.5 on. This is to make sure that all the subsamples are not biased toward any particular classes unseen data. Once we run C4.5 and obtain the decision tree, we only pick attributes that only appear in the first three levels of the decision trees as the most relevant features. The paper hypothesize that if a feature in the deeper levels on any one execution of C4.5 is relevant enough, it will finally rises up and appear in one of the top levels of the tree in some other executions of C4.5. Selective Bayesian Classifier forms a union of all the attributes in the first three levels from each run, and finally, run the Naïve Bayesian classifier on the training and test data using only those features selected in the previous step.

4. Result Reproduction

4.1 Reproduction Process

- Shuffle each dataset randomly.
- Produce disjoint training and testing sets as follows

  10% training and 90% testing data

  20% training and 80% testing data

  30% training and 80% testing data

  ....

  80% training and 10% testing data

  90% training and 10% testing data

  99% training and 1% testing data
- For each set of training and testing data, run NBC, C4.5 Decision Trees and SBC.
- Repeat for 15 times.

The classifier accuracy is determined by Random Subsampling method, i.e. the holdout method that is repeated k times. The overall accuracy estimate is the mean of the accuracies obtained from all iterations.

4.2 Reproduction Results

The results did not come out as expected. The asymptotic accuracy of SBC is not as good as NBC and C4.5 on each of the domains.

Figure 2 – 10 depict the learning curves for the 10 UCI datasets. It is clear that SBC performs slightly better than NBC in nearly half of the domains, and performs better than C4.5 Decision Trees in less than half domains. It shows that SBC learns faster than both C4.5 and NBC on all the dataset, i.e. with small number of training data (e.g. 10%), the prediction accuracy for SBC is higher.

Note that all the C4.5 accuracies considered in this experiment are based on the simplified decision tree with pruning. This accuracy is usually higher on the test (unseen) data, in comparison to the accuracy based on unpruned decision trees.

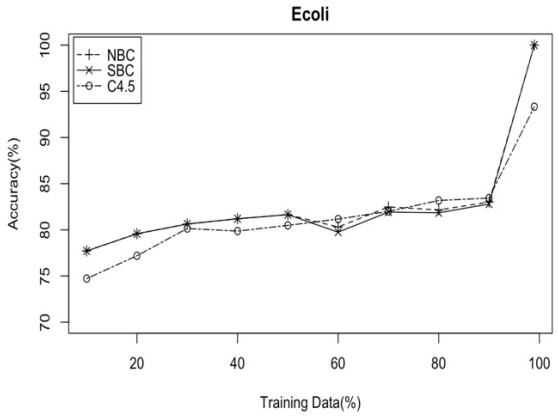

Figure 2. Ecoli dataset. 336 instances, 8 attributes, 8 classes. Attributes selected by SBC = 6.

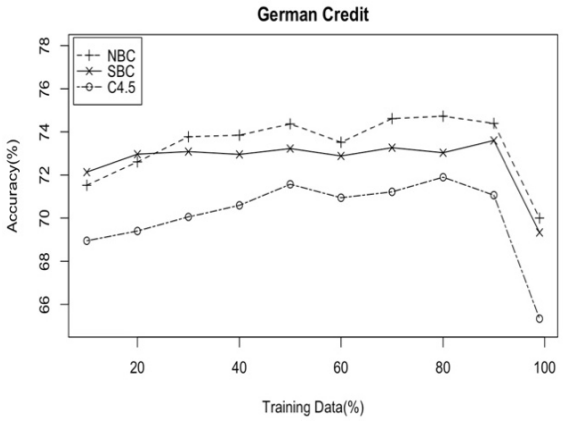

Figure 3. German Credit dataset. 1,000 instances, 20 attributes, 2 classes. Attributes selected by SBC = 5.

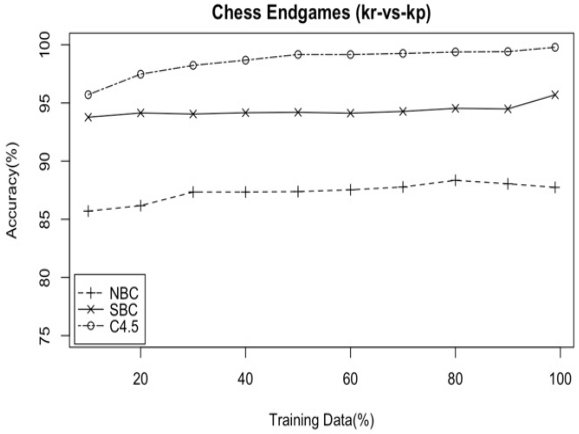

Figure 4. Kr-vs-Kp dataset. 3,198 instances, 37 attributes, 2classes. Attributes selected by SBC = 4.

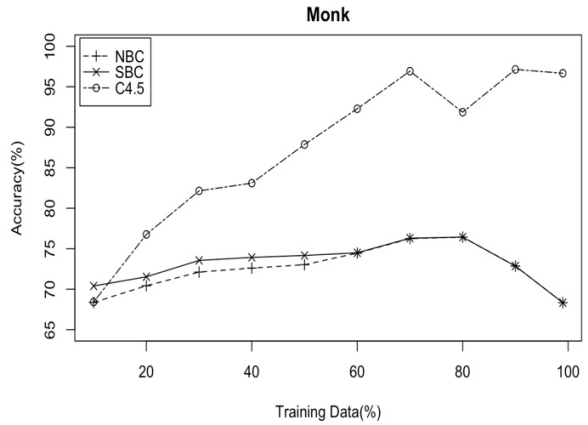

Figure 5. Monk dataset (prob.3). 554 instances, 6 attributes, 2 classes. Attributes selected by SBC = 4.

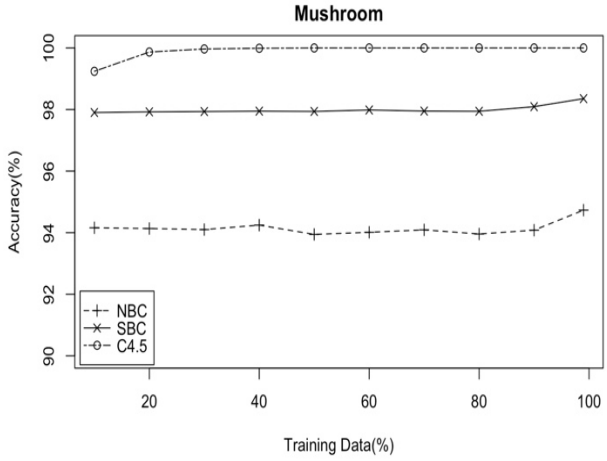

Figure 6. Mushroom dataset. 8,124 instances, 22 attributes, 2 classes. Attributes selected by SBC = 4.

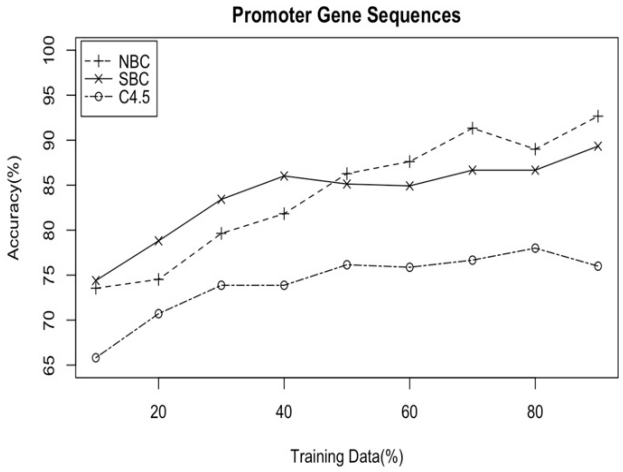

Figure 7. Gene Promoter dataset. 106 instances, 57 attributes, 2 classes. Attributes selected by SBC = 5.

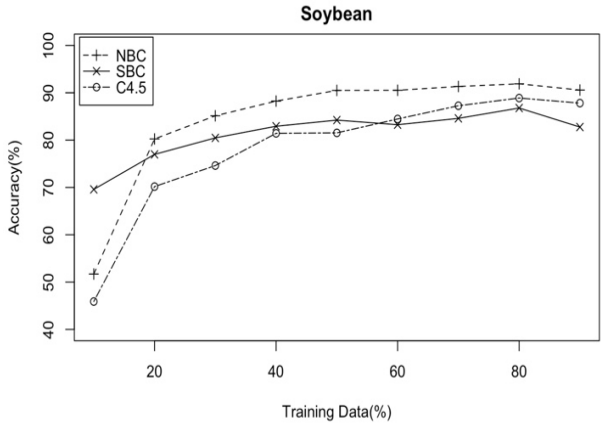

Figure 8. Soybean-large dataset. 307 instances, 35 attributes, 19 classes. Attributes selected by SBC = 12.

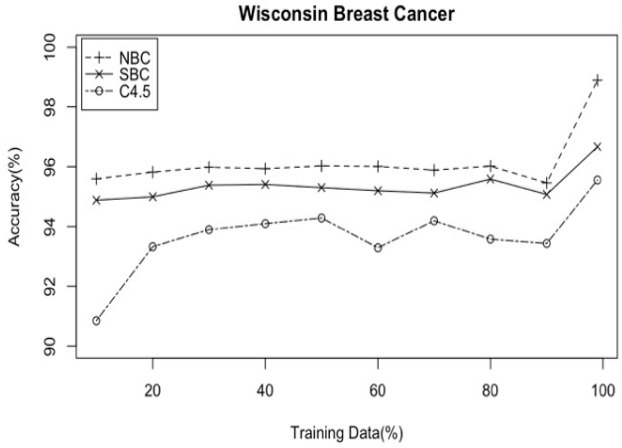

Figure 9. Wisconsin Breast Cancer dataset. 699 instances, 9 attributes, 2 classes. Attributes selected by SBC = 6.

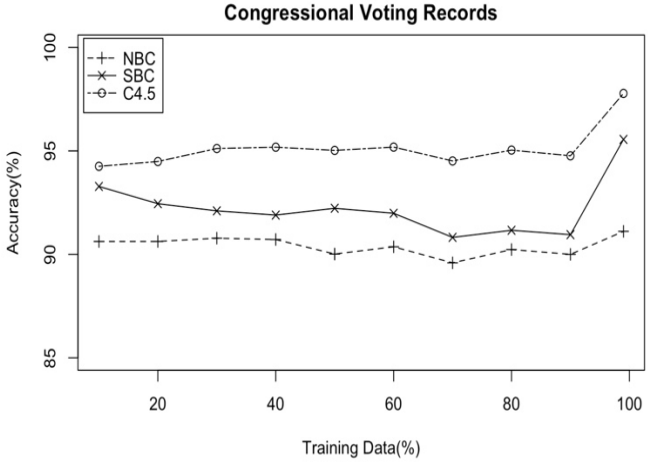

Figure 10. Congressional Voting dataset. 435 instances, 16 attributes, 2 classes. Attributes selected by SBC = 3.

From the figures we can see that nearly in each of the domains, SBC is not the approach which has the best performance. In Figure 4, 5, 6 and 10, the descending order of approaches depending on accuracy is C4.5 Decision Trees, Selective Bayesian Classifier, Naïve Bayesian Classifier. In Figure 3, 8, 9, the descending order is Naïve Bayesian Classifier, Selective Bayesian Classifier, C4.5 Decision Trees. In Figure 2, Naïve Bayesian Classifier performs nearly same as Selective Bayesian Classifier, they both perform better than C4.5 Decision Trees. In figure 7, when training data is small, Selective Bayesian Classifier outperforms C4.5 Decision Trees, but when training data grows larger, C4.5 begins to outperform SBC.

Table 2 shows the results for NBC, C4.5, and SBC learning algorithms using 80% of the data for training and 20% for testing (5-fold cross- validation). The figures reported in bold reflect the winning method on each dataset. The last two columns show the difference between SBC with NBC, C4.5, and ABC, respectively. The last row of the table gives the mean accuracies and differences for each learning algorithm.

Table 2. Accuracy of each learning method using 5-fold cross-validation

Dataset	NBC	C4.5	SBC	SBC vs NBC	SBC vs C4.5
Ecoli	81.53%	**81.83%**	81.53%	0.00%	-0.37%
GerCredit	**74.80%**	71.50%	73.50%	-1.74%	2.80%
KrVsKp	87.96%	**99.31%**	94.09%	+6.97%	-5.26%
Monk	75.00%	**91.90%**	75.00%	0.00%	-18.38%
Mushroom	94.08%	**100.00%**	97.93%	+4.10%	-2.07%
Promoter	**88.65%**	68.83%	85.80%	-3.21%	24.65%
Soybean	**91.20%**	87.32%	84.71%	-7.12%	-2.99%
Wisconsin	**96.00%**	93.84%	95.43%	-0.60%	1.68%
Vote	90.11%	**94.49%**	91.26%	+1.28%	-3.42%
Mean	86.59%	**87.67%**	86.58%	-0.01%	-1.24%

Table 3 shows the number of features selected for Selective Bayesian classifier. On almost all the datasets, surprisingly more than half of the original attributes were eliminated. 30% or less of all attributes selected were shown in bold. It means that we can actually pay no attention to more than 70% of the original data and still achieve very high accuracy in classification.

Table 3. Number of features selected

Dataset	#Attributes	# of Attributes selected
Ecoli	8	6
GermanCredit	20	**5**
KrVsKp	37	**4**
Monk	6	4
Mushroom	22	**4**
Promoter	57	**5**
Soybean	35	12
Wisconsin	9	6
Vote	16	**3**

Table 4 shows the running time for each classifier on the 9 datasets. The last row of the table gives the mean running for each learning algorithm. The running time of SBC is much smaller than NBC, the running time of SBC and NBC is much smaller than that of C4.5 in over half of the domains because Bayesian classifier only needs to go through the whole training data once.

Table 4. Running Time of each learning method(unit:second)

Dataset	C4.5	NBC	SBC
ecoli	25.07	4.92	5.79
german	10.24	23.13	11.35
krkp	16.40	113.23	26.42
monk	6.16	4.00	4.37
mushroom	15.99	174.64	63.50
gene	21.30	7.23	2.38
soybean	15.71	13.32	14.50
cancer	6.10	7.66	7.07
vote	8.99	8.23	3.72
mean	14.00	39.60	16.66

4.3 Reproduction Discussion

Due to unsatisfying reproduced result, we analyze the reasons. We think the main reason is that feature selection process in this Selective Bayesian Classifier is not appropriate enough. It runs C4.5 Decision Trees on a 10% sample of a training set and selects a set of attributes that appear only in the first 3 levels of the simplified decision tree as relevant. Then repeat this process for 5 times. The paper says 10% of the training is a good size for our feature selection process. If too small a portion is used, the decision tree may not be representative enough for the unseen data. And if too large a portion is used, it unnecessarily takes longer to construct a decision tree and the tree also are likely to be too complex. However, the fact proves that 10% sample of a training set is so small that it affects the accuracy. When we do this feature selection process on a dataset for 5 times, the sets of attributes in the first 3 levels each time are different from each other. On one hand, given the size of the datasets, the size of a 10% sample is so small, which easily leads to the loss of ubiquity. The attributes after the third level of a simplified decision are likely to contribute to the final decision. On the other hand, since the feature selecting algorithm forms a union of all the attributes from the 5 rounds, given the fact that the sets of attributes obtained in each round are different，two or more correlated attributes are likely to be selected together. If we do not remove redundant attributes, it will totally violate the main idea of SBC. We will talk about an improvement later to overcome these shortcomings.

5. Improved Selective Bayesian Classifier

5.1 Description

In this project, we propose a new approach called Improved Selective Bayesian Classifier(ISBC), where we mainly improve the feature selection part. We compute *Information Gain Ratio* and *chi-square value* at the root node of a decision tree and select features with highest values. In order to avoid suffering from redundant attributes, we compute Pearson Correlation Coefficient and draw heat graphs out. We remove the redundant attributes and then apply Naïve Bayesian Classifier. We present experimental evidence that this method of feature selection leads to improved performance of Naïve Bayesian Classifier, and improves performance of C4.5 to some extent.

5.2 Chi-Square Test

Chi-square test is a widely used hypothesis test method. It is applied in statistical inference of classification data, including: chi-square test for comparison of two ratios

or two constituent ratios, chi-square test for comparison of multiple ratios or multiple constituent ratios, and correlation analysis of data classification.

The classical chi-square test is to test the correlation between qualitative independent variables and qualitative dependent variables. Considering the difference between the observed values and expectations of sample frequencies with independent variables equal to I and dependent variables equal to j, the meaning of this statistic is simply the correlation between independent variables and dependent variables. In our project, we compute *chi-square value* as long as *information gain ratio* at the root node of the decision tree. The greater the *chi-square value*, the more relevance an attribute has with the final classification. Thus, there is mutual authentication between *information gain ratio* and *chi-square value.*

5.3 Pearson Correlation Coefficient

In order to avoid suffering from redundant attributes, we introduce pearson correlation coefficient to find them. Pearson correlation coefficient is also called Pearson product-moment correlation coefficient. It is a linear correlation coefficient. Pearson correlation coefficient is used to reflect the degree of linear correlation between two variables. The correlation coefficient is expressed by r, where n is the sample size, which is the observation value and the mean value of two variables respectively. R describes the degree of linear correlation between two variables. The greater the absolute value of r, the stronger the correlation. Note that we can only compute pearson correlation coefficients for numerical attributes, thus we can only find redundant attributes which is numerical type, but not factor type. In our Improved Selective Bayesian Algorithm, we compute pearson correlation coefficients for the attributes in a data set and draw the heat graph, then remove the redundant attributes from the attributes we have selected according to information gain ratio and chi-square value.

5.4 Algorithm

(1)Shuffle the training data.

(2)Compute information gain ratio along with chi-square value at the root node of the decision tree.

(3)Select a set of attributes with the highest information gain ratios and chi-square values. The decision mainly depend on the information gain ratio, we use chi-square test to mutually verify with it. As for the attribute selection, we divide the data sets into three part: The first part: some of the attributes have obviously higher information ratio than the others. Then select them directly. The second part: some of the attributes have obviously lower information ratio than the others. Then delete them and select the rest

ones. The third part: If it is hard to decide which ones to select, then we may selectively delete or reserve uncertain attributes, and test the hypothesized attributes to see if they are important.

(4) With selected attributes in step 3, compute pearson correlation coefficients and draw heat graphs. Remove redundant attributes.

(5)Run Naïve Bayesian Classifier on the training and testing data using only the final features selected in step 4.

5.5 Experimental Design and Process

- Shuffle each dataset randomly.
- Produce disjoint training and testing sets as 20% training and 80% testing data.
- For each set of training and testing data, run Improved Selected Bayesian Classifier.
- Repeat for 15 times.

The classifier accuracy is determined by Random Subsampling method, i.e. the holdout method that is repeated k times. The overall accuracy estimate is the mean of the accuracies obtained from all iterations.

Figure 11-19 show the result of calculating information gain ratios in 9 domains. The red bars denote that the attributes are of factor type and the blue bars denote that the attributes are of numeric type.

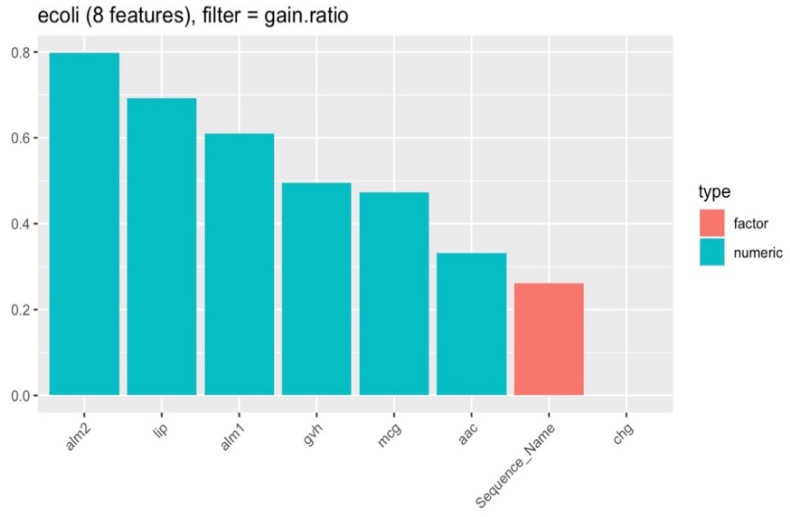

Figure 11 Ecoli dataset. Information Gain Ratio

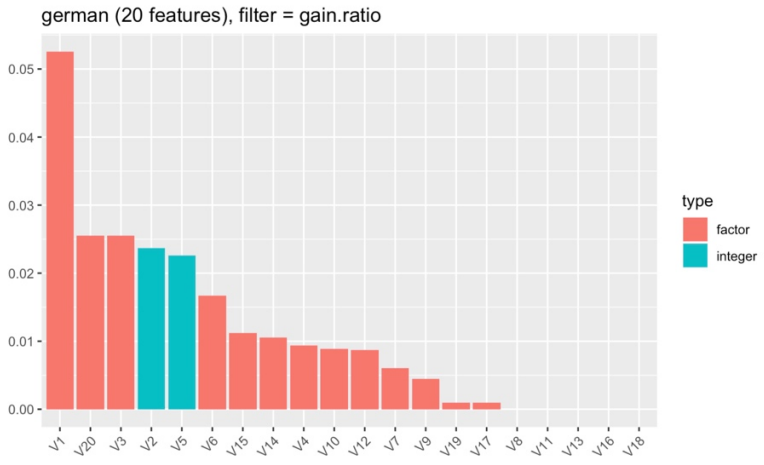

Figure 12 German dataset. Information Gain Ratio

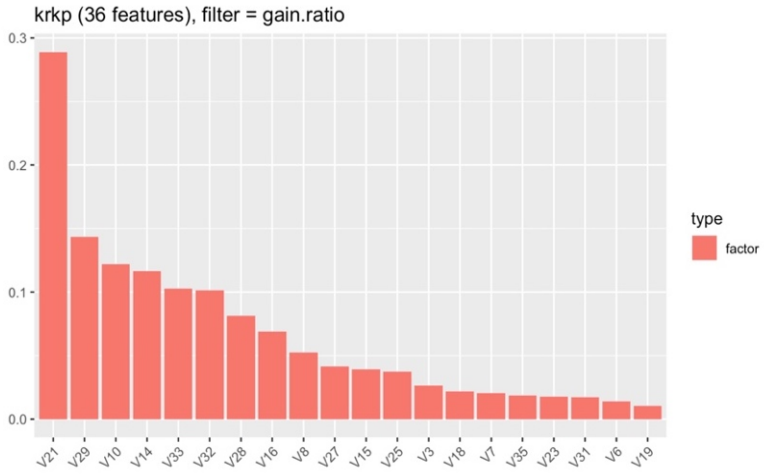

Figure 13 Krkp dataset. Information Gain Ratio

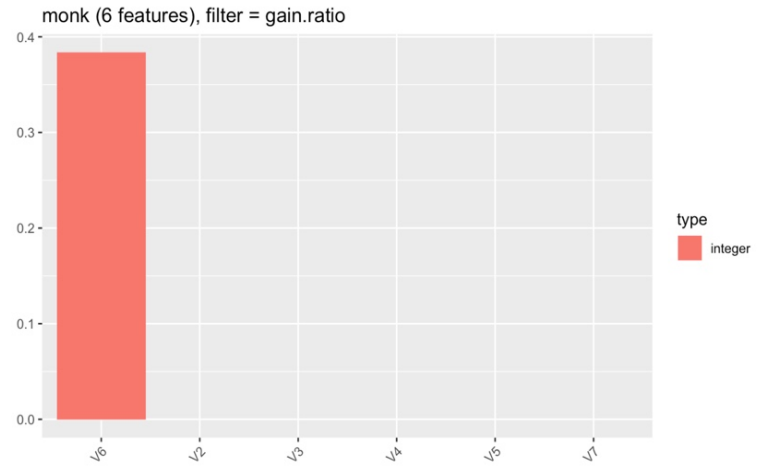

Figure 14 Monk dataset. Information Gain Ratio

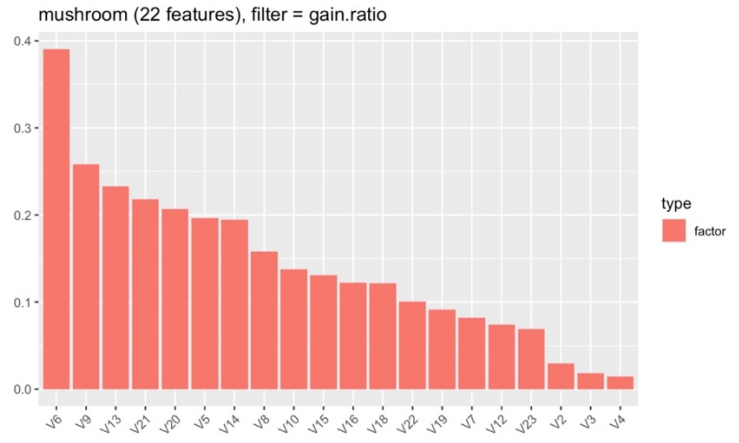

Figure 15 Mushroom dataset. Information Gain Ratio

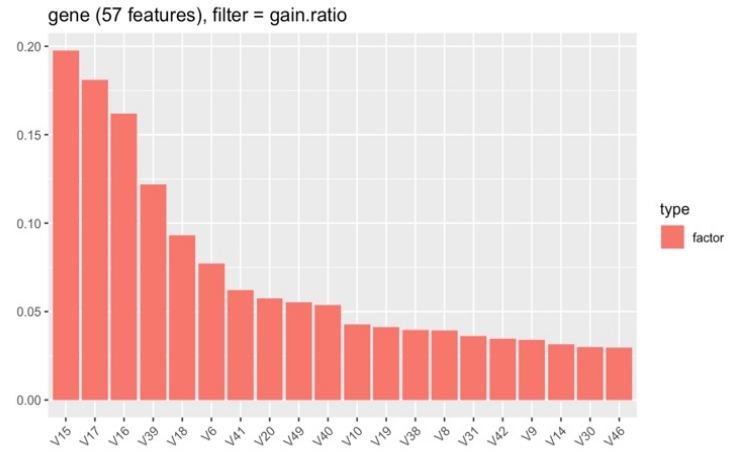

Figure 16 Promoter dataset. Information Gain Ratio

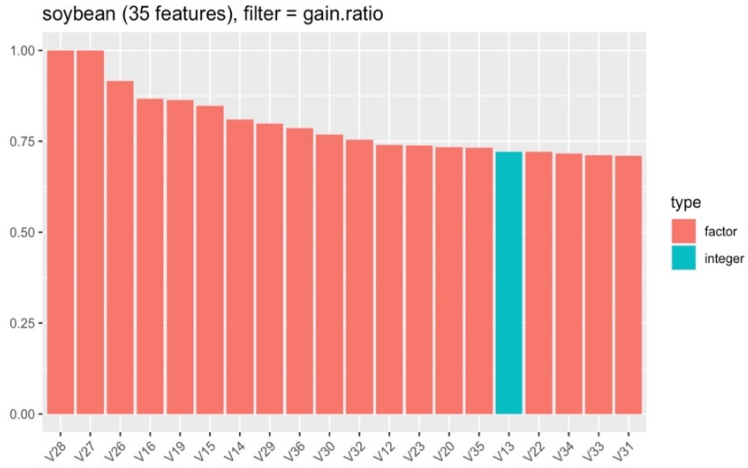

Figure 17 Soybean dataset. Information Gain Ratio

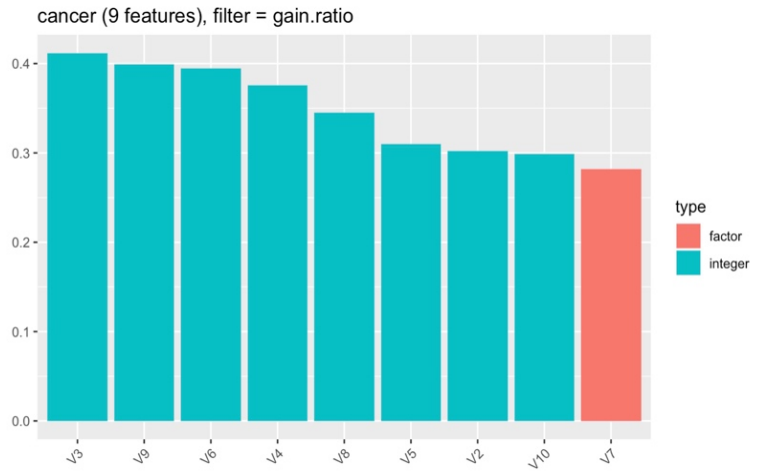

Figure 18 Wisconsin breast cancer dataset. Information Gain Ratio

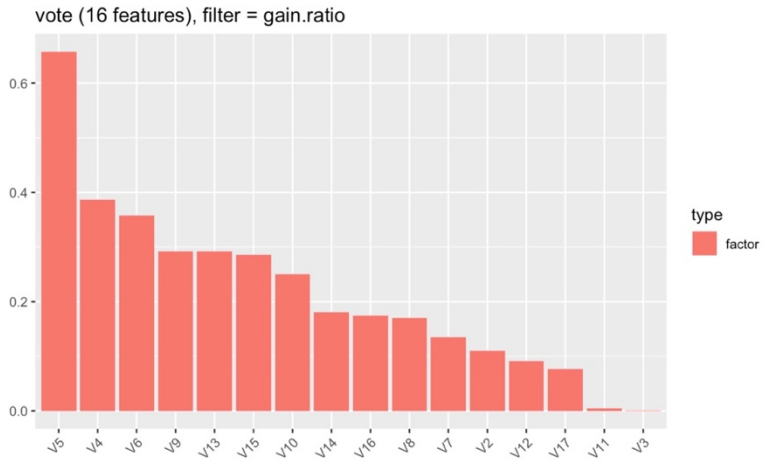

Figure 19 Vote dataset. Information Gain Ratio

Figure 20-22 show the heat graphs of dataset Ecoli, data Monk and dataset Wisconsin breast cancer. Note that blue points mean that attributes are negatively correlated, the red points mean that attributes are positively correlated. The darker the point, the stronger their relationship is.

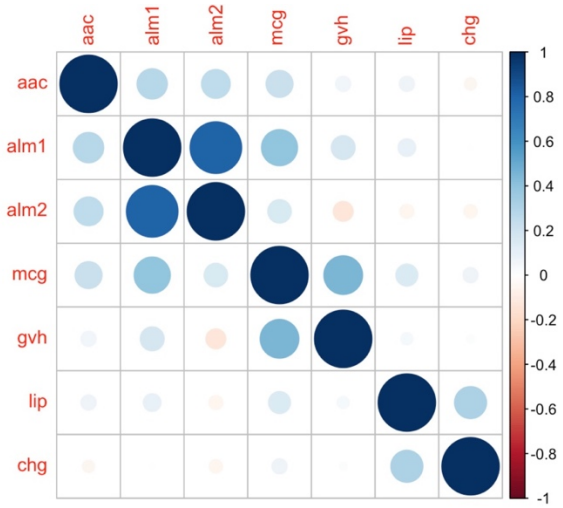

Figure 20. Dataset Ecoli. Heat Graph.

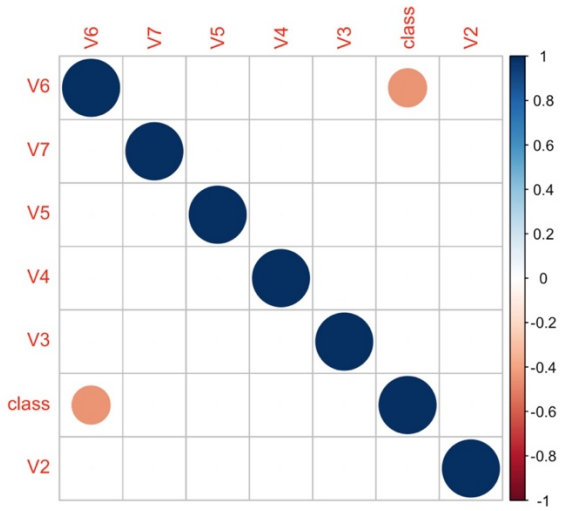

Figure 21. Dataset Monk. Heat Graph.

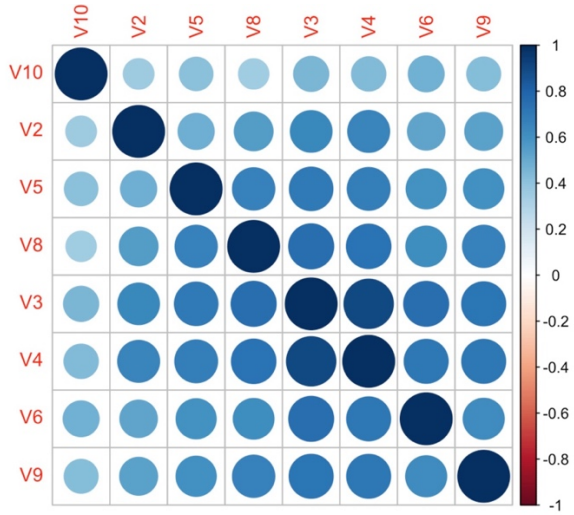

Figure 22. Dataset Wisconsin Cancer. Heat Graph.

5.6 Experimental Results

The results confirm the initial hypotheses. The performance of the Improved Selective Bayesian classifier is quite impressive. Its asymptotic accuracy is as good as (or slightly better than) C4.5 and NB on nearly each of the domains. Table 5 shows the results for NBC, C4.5, SBC, and ISBC learning algorithms using 80% of the data for training and 20% for testing (5-fold cross validation). The figures reported in bold reflect the winning method on each dataset. The last two columns show the improvement of ISBC over NBC and C4.5, respectively. The last row of the table gives the mean accuracies and improvements for each learning algorithm.

Table 5. Accuracy of each learning method using 5-fold cross-validation

Dataset	NBC	C4.5	SBC	ISBC	ISBC vs NBC	ISBC vs C4.5
Ecoli	81.53%	**81.83%**	81.53%	78.57%	-3.63%	-3.99%
GerCredit	74.80%	71.50%	73.50%	**76.10%**	+1.74%	+6.43%
KrVsKp	87.96%	**99.31%**	94.09%	94.34%	+7.26%	-5.01%
Monk	75.00%	**91.90%**	75.00%	75.00%	0.00%	-18.38%
Mushroom	94.08%	**100.00%**	97.93%	98.92%	+5.14%	-1.08%
Promoter	88.65%	68.83%	85.80%	**94.23%**	+6.30%	+36.91%
Soybean	91.20%	87.32%	84.71%	**93.17%**	+2.16%	+6.70%
Wisconsin	96.00%	93.84%	95.43%	**96.14%**	+0.15%	+2.45%
Vote	90.11%	94.49%	91.26%	**94.71%**	+5.11%	+0.24%
Mean	86.59%	87.67%	86.58%	**89.02%**	+2.81%	+1.54%

From table 5, it is apparent that ISBC outperforms the original NBC in every domain, giving the accuracy improvement up to 7.26%. SBC also outperforms C4.5 in more than half of the domains, giving the accuracy improvement up to 36.91%. Even though, SBC cannot beat C4.5 in some datasets, it still gives quite big improvement over the Naïve Bayesian (7.26% and 5.14%,) on such cases. From the mean accuracy of these four approaches, we could see that ISBC performs the best.

Table 6 shows the number of features selected for Selective Bayesian classifier. On over half of the datasets, surprisingly more than half of the original attributes were eliminated. 30% or less of all attributes selected were shown in bold. It means that we can actually pay no attention to more than 70% of the original data and still achieve very high accuracy in classification. As for this part, ISBC performs as well as SBC.

Table 6. Number of features selected

Dataset	#Attributes	# of Attributes selected SBC	# of Attributes selected ISBC
Ecoli	8	6	5
GermanCredit	20	**5**	13
KrVsKp	37	**4**	**6**
Monk	6	4	**1**
Mushroom	22	**4**	**3**
Promoter	57	**5**	**4**
Soybean	35	12	33
Wisconsin	9	6	8
Vote	16	**3**	**3**

6. Conclusion and Discussion

In this project, we implement Naïve Bayesian Classifier, C4.5 Decision Trees, Selective Bayesian Classifier and Improve Selective Bayesian Classifier to classify 9 datasets from the UCI repository. Selective Bayesian Classifier is proposed in paper *Scaling up the Naive Bayesian Classifier: Using Decision Trees for Feature Selection*, this is a combination of the two different natures of classifiers. The reproduction result shows that the algorithm described in the paper is not satisfying enough. This Selective Bayesian Classifier outperforms Naïve Bayesian Classifier only in half of the cases. The reason is that the original algorithm may lose ubiquity and introduce redundant attributes. Thus, we propose a new approach which uses information gain ratio, chi-square value and pearson correlation coefficient to select features. The empirical evidence shows that this method is very successful. This improved Selective Bayesian Classifier is more accurate than Naïve Bayes and SBC on each of the domains, and more accurate than C4.5 Decision Trees on over half of the domains. On those domains where C4.5 beat ISBC, we find that C4.5 performs better than NBC.

This work suggests that when the attributes are independent from each other, Naïve Bayesian Classifier can work very well, and C4.5 works well in selecting good features which do not contain redundant ones, because the tree remains the same if C4.5 uses a redundant value to construct a decision tree. The fact that ISBC achieves high accuracy suggests that information gain ratio is a useful measure to select the best attributes for classification. Pearson correlation coefficient catches numerical attributes which are correlated with each other and helps ISBC to remove redundant features. Besides, the running times of different approaches suggest that the feature selection would learn

more quickly, it would need fewer training examples to reach high classification accuracy.

7. Reference

[1] C.A. Ratanamahatana, D. Gunopulos, "Scaling up the Naive Bayesian Classifier: Using Decision Trees for Feature Selection", *Proc. Workshop Data Cleaning and Preprocessing (DCAP '02) at IEEE Int'l Conf. Data Mining (ICDM '02)*, 2002.

[2] Zhang, H., Ling, C.X., and Zhao, Z. The Learnability of Naïve Bayes. Canadian Conference on AI 2000: 432-411.

[3] Quinlan, J.R.(1990). Induction of Decision Trees. In Reading in Machine Learning. Morgan Kaufmann, Dordrecht, The Netherlands. Originally published in Machine Learning 1:81-106, 1986.