Comparison of Naive Bayes and Decision Tree on Feature Selection Using Genetic Algorithm for Classification Problem

S Rahmadani, A Dongoran, M Zarlis, Zakarias
Magister Teknik Informatika, Fasilkom - TI, Universitas Sumatera Utara, Medan - Indonesia

Email: dhani.harahap@gmail.com

Abstract. This paper discusses the problem of feature selection using genetic algorithms on a dataset for classification problems. The classification model used is the decision tree (DT), and Naive Bayes. In this paper we will discuss how the Naive Bayes and Decision Tree models to overcome the classification problem in the dataset, where the dataset feature is selectively selected using GA. Then both models compared their performance, whether there is an increase in accuracy or not. From the results obtained shows an increase in accuracy if the feature selection using GA. The proposed model is referred to as GADT (GA-Decision Tree) and GANB (GA-Naive Bayes). The data sets tested in this paper are taken from the UCI Machine Learning repository.

1. Introduction
The issue of the huge dataset feature is still an important issue in data mining task. The main problem is how to choose or even reduce one or more features on the dataset that may affect the quality of data. Features that may interfere with data, irrelevant, correlated, or even overlapping must be cleaned so that the quality and performance of the data can be increased [1].

There are two common tasks related to feature reduction, Feature Selection and Feature Composition. There are two feature selection methods, feature-ranking algorithm and minimum subset algorithm. Feature selection algorithm is very helpful data mining process to seek new knowledge in finding models of data patterns that have high accuracy. One of feature selection technique is to compare means and variance. The trivial technique is its unknown feature distribution. If the distribution is normal then the results will be better statistically but weak on the assumption. If the distribution is not known then means and variance are heuristic so the value is not too exact. In this paper, the technique used is non statistical technique but using heuristic method, Genetic Algorithm.

The feature selection is a fairly complex task that can affect the performance of classification. The most popular approach is based on Evolutionary Computation (EC) such as Genetic Algorithm [12] and Swarm Intelligence called Particle Swarm Optimization (PSO).

2. Naive Bayes
Naive Bayes is a classic model that has been studied since the 1950s. It became popular after being tested for text retrieval in the 1960s [8], combined with Support Vector Machine [9], and also applied to medical diagnosis [10]. Naive Bayes is one of the statistical approach of inductive-inferencing in the classification problem that uses bayes theorem.
Let H be a hypothesis and X is a data residing in a certain C class. Then P (H / X) is called the posterior probability that expresses our confidence level on a hypothesis H after X data is given. P (H) represents the H prior probability for all sample data. P (H / X) is certainly more informative than P (H). Bayes's theorem describes the relationship between P (H / X), P (H), and P (X) is shown on equation 1 as follow:

$$P(H/X) = \frac{P(X/H) \cdot P(H)}{P(X)}$$  \hspace{1cm} (1)

By using the bayes theorem, the Naive Bayes Classifier can be formulated as follows on equation 2.

$$P(C_i/X) = \frac{P(X/C_i) \cdot P(C_i)}{P(X)}$$  \hspace{1cm} (2)

P (X) is the constant probability of the X dataset for all classes. P(C_i) is the number of training samples of the class Ci/m (m is the number of training data samples). In this case P(X/C_i) * P(C_i) is a unit that can be optimized so that P(C_i/X) becomes optimal, since P (X) and P(C_i) are constant. Based on the aforementioned assumptions, P(X/C_i) is formulated as follows on equation 3.

$$P(X/C_i) = \prod_{i=1}^{n} P(x_i/C_i)$$  \hspace{1cm} (3)

Where x_i is the value of the sample attribute X. The probability value P (x_i/C_i) can be estimated from the training dataset.

2.1. Naive Bayes Algorithm
The steps of Naive Bayes’s Algorithm in the classification of data are as follows:

- Find P(C_i), that is the class i probability by calculating total class i in total m training dataset.
- Calculate the probability P(x_i/C_i) for each attribute value of the new data sample X using the training dataset.
- Calculate P(X/C_i) Using equation 3 for each class.
- Calculate P(C_i/X) Using equation 2 for each class.
  - P(C_i/X) ≈ P(X/C_i) * P(C_i).
- Choose a maximum probability value P(C_i/X) Which is the result of class predictions of the new data.

3. Decision Tree
One of method that is quite reliable and efficient in generating a classifier model is the decision tree. The Decision Tree is called a predictive model that maps the observed data into the target value. In predicting the model, the Decision tree usually uses a statistical approach, data mining and machine learning. Decision tree is included into the inductive learning algorithm. Decision tree has a special feature in the learning process that is using a top-down strategy.

3.1. Decision Tree Algorithm
Assumed that all features have finite discrete domains, and there is a target feature called classification. Each element in the classification is called a class. Decision Tree learns from source data that has been divided into two parts, data training and data testing. The algorithm for forming a decision tree typically uses a top-down method [3].

There are many decision tree algorithms, such as ID3, C4.5, CART, CHAID, MARS, Conditional Inference Trees. In this case, the algorithm used is C4.5.

The steps to form a tree using the C4.5 algorithm are as follows:
- Select the feature (attribute) as the root
• Create a branch for each value
• Divide the cases into branches
• Repeat the process for each branch until all the cases on the branch have the same class.

To select the attribute as the root is used the highest gain information of each attribute. The gain information formula in equation 4.

\[
Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} \cdot Entropy(S_i)
\]  

(4)

where \( S \) is a set of cases. \( A \) is a set of feature (attribute), \( n \) represents number of attribute partitions, \( |S_i| \) is number of case on partition to-i, and \( |S| \) is number of cases in \( S \). While to calculate the value of entropy used equation 5.

\[
Entropy(S) = \sum_{i=1}^{n} - (p_i \cdot \log_2 p_i)
\]  

(5)

where \( S \) is a set of cases, \( n \) is number of partition, and \( p_i \) is a proportion of \( S_i \) of \( S \).

4. Genetic Algorithm (GA)
Genetic Algorithm is one of the heuristic search algorithms that mimics the process of natural evolution. GA is used for the problem of optimasi and search that is inspired by natural evolution such as inheritance, mutation, selection and marriage. GA was first proposed by Fraser [12], and then Bremerman [13].

4.1. GA for Feature Selection
In GA, mutations are done by turning on and off attributes. Crossover is defined as a feature change that is used.

Selection feature stages are as follows:
• Generate initial population composed of individual \( p \)
• Each attribute is enabled with probability \( p_i \). The size of \( p \) and \( p_i \) is adjusted by population size and initial \( p \).
• All individuals in the population perform mutations, for example setting the attributes used to be unused attributes based on the probability of the mutation \( pm \) and vice versa. The probability of \( pm \) is obtained from the initial parameter of \( pm \).
• Choose two individuals from the population to cross-crossover based on the probability of crossover \( pc \). \( Pc \) is taken from the initialization of \( pc \) parameters. The type of crossover used can be selected from crossover type parameters.
• Mapping all individuals based on their fitness values for the selection process. Take individuals randomly according to their respective probabilities.
• As the fitness value increases, repeat step 2. Unless, the process is finished.

4.2. Crossover
In binary representation, crossover is done in the sexual way of two parents [12]. Some of the crossover operators used in GA, among others:
• One-point crossover : segments of the genes are exchanged between parents who form offspring, and not single genes [14].
• Two-point crossover : two bit positions are randomly selected and the bitstring subsequently exchanged.
• Uniform crossover : \( n \) - dimensional masks are created randomly [10]. If \( px = 0.5 \) (bit-swaping probability) it means the chance of bits to be exchanged equally.

In this paper, the crossover operator used using Uniform Crossover.
4.3. Mutation
For binary data, the usual mutation operator is as follows:

- **Uniform (Random) Mutation** [14]: the bits position is selected and the selected bits are reversed, from 0 to 1, and vice versa.
- **Inorder Mutation**: two points of mutation are randomly selected, bits between these two points behind the bits.
- **Gaussian Mutation**: for each chromosome, the random number of the Poisson is distributed to determine the gene that is mutated.

5. Design Experiment
The experiment design of this paper is shown in Figure 1.

![Figure 1. Methodology of Experiment](image)

6. Result
The dataset used to train and test the above experiments was obtained from UCI Machine Learning Repository. Table 1 shows the dataset with property A represents the attribute, I denotes the number of instances, C denotes the number of classes. Tables 2, 3, and 4 show the performance vector of the three Iris, Glass, and Credit datasets. Table 5 shows the results of data validation by training using Naive Bayes, and Table 6 shows the Decision Tree result.

| Attribute (A) | Class. (C) | Instance. (I) |
|---------------|------------|---------------|
| Iris          | 4          | 4             | 150           |
| Glass         | 9          | 7             | 214           |
| Credit        | 20         | 2             | 1000          |
### Table 2. Performance Vector Germany Credit GA-DT.

|               | True GOOD | True BAD | Class Precision |
|---------------|-----------|----------|-----------------|
| Pred. GOOD    | 178       | 52       | 77.39 %         |
| Pred. BAD     | 30        | 38       | 55.88 %         |
| Class Recall  | 85.58 %   | 42.22 %  |

### Table 3. Performance Vector Iris GA-DT.

|               | True Iris-Setosa | True Iris-VersiColor | True Iris-Virginica | Class Precision |
|---------------|------------------|-----------------------|---------------------|-----------------|
| Pred. Iris-Setosa | 25             | 0                     | 0                   | 100.00 %        |
| Pred. Iris-VersiColor | 0             | 25                    | 1                   | 96.15 %         |
| Pred. Iris-Virginica | 0              | 0                     | 24                  | 100.00%         |
| Class Recall   | 100.00 %        | 100 %                 | 96.00 %             |

### Table 4. Performance Vector Glass GA-DT.

|               | True 1 | True 2 | True 3 | True 5 | True 6 | True 7 | Class Precision |
|---------------|--------|--------|--------|--------|--------|--------|-----------------|
| Pred. 1       | 14     | 1      | 3      | 0      | 0      | 1      | 73.68 %         |
| Pred. 2       | 4      | 19     | 0      | 0      | 0      | 0      | 82.61 %         |
| Pred. 3       | 2      | 2      | 2      | 0      | 0      | 0      | 33.33 %         |
| Pred. 5       | 0      | 1      | 0      | 4      | 0      | 0      | 80.00 %         |
| Pred. 6       | 1      | 0      | 0      | 0      | 3      | 0      | 75.00 %         |
| Pred. 7       | 0      | 1      | 0      | 1      | 0      | 8      | 100.00 %        |
| Class Recall  | 66.67 % | 82.61 % | 40.00 % | 100.00 % | 100.00 % | 88.89 % |

### Table 5. Accuracy of Prediction Naive Bayes

|       | NB Naive Bayes | GA-NB | Acc. Improvement |
|-------|----------------|-------|------------------|
| Iris  | 96.00 %        | 100 % | 4.0 %            |
| Glass | 56.92 %        | 64.62 % | 7.7 %         |
| Credit| 73.67 %        | 80.00 % | 6.33 %         |
| Average|                |       | 6.01 %           |

### Table 6. Accuracy of Prediction Decision Tree

|       | DT Decision Tree | GA-DT | Acc. Improvement |
|-------|------------------|-------|------------------|
| Iris  | 94.67 %          | 98.67 % | 4.0 %             |
7. Summary
From the results obtained above, it is seen that the accuracy for each dataset is different. The highest accuracy is obtained by the Iris dataset, then Glass and last Data Germany Credit. However, when viewed from the level of increase in accuracy before and after done GA-Featured Selection, then the Glass dataset has the highest value. When compared between the Decision Tree and Naive Bayes models, for the three Iris, Glass, and Credit datasets the Decision tree performance is slightly better than Naive Bayes. Thus it can be concluded that Feature Selection using GA can improve the accuracy of both the Decision Tree and Naive Bayes models, and the Decision Tree shows slightly better accuracy than Naive Bayes.

References
[1] Kantardzic M 2003 Data Mining: Concept, Models, Methods, And Algorithms (IEEE Press 445 Hoes Lane, Piscataway, NJ 08854).
[2] Witten IH, Frank E, Hall MA 2011 Data Mining: Practical Machine Learning Tools and Techniques. (Burlington: Morgan Kaufmann Publisher).
[3] Rokach L and Maimon O 2005 Top-down induction of decision trees classifiers-a survey IEEE Trans. Syst. Man Cybernetics Part C (Appl. Rev.) 35(4) pp 476-87.
[4] Cardic C 1993 Proc. Int. Conf. Mach. Learn.
[5] Brownlee Jason 2011 Clever Algorithms Nature-Inspired programming Recipes (Australia: Creative Commons Attribution-Noncommercial-Share Alike 2.5 Australia License).
[6] Strobl, C.; Malley, G.; Tutz 2009. An Introduction to Recursive Partitioning: Rationale, Application and Char- acteristics of Classification and Regression Trees, Bag- ging and Random Forests. Psychological Methods 14 (4): 323–348. Strobl C, Malley J and Tutz G 2009 An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests Psy. Methods 14(4) pp 323.
[7] Breiman L, Friedman J, Stone CJ and Olshen RA 1984 Classification and regression trees (Monterey; CRC press).
[8] Rokach L, Maimon O 2008 Data mining with decision trees: theory and applications. (Singapore: World Scientific Pub Co Inc)
[9] Hebb D 1949 The Organization of Behavior (New York: Wiley).
[10] Farley BW, Clark W 1954 Simulation of self-organizing systems by digital computer Trans. IRE Prof. Group Inf. Theory 4(4) pp 76-84.
[11] Goldberg DE 1989 Genetic Algorithms in Search, Optimization, and Machine Learning. (Boston: Addison-Wesley).
[12] Fraser AS 1957 Simulation of Genetic Systems by Automatic Digital Computers I Introduction Aust. J. Biol. Sci. 10 pp 484–91.
[13] Reed J, Toombs R and Barricelli NA 1967 Simulation of Biological Evolution and Machine Learning J. Theor. Biol. 17 pp 319–42.
[14] Holland JH 1975 Adaptation in Natural and Artificial Systems (Michigan: University of Michigan).