A Comparison of Decision Tree Algorithms For UCI Repository Classification

Kittipol Wisaeng

Mahasakham Business School (MBS), Mahasakham University
Kantharawichai, Khamriang , Mahasarakham, 44150, Thailand.

***Abstract*—The development of decision tree algorithms have been used for industrial, commercial and scientific purpose. However, the choice of the most suitable algorithm becomes increasingly difficult. In this paper, we present the comparison of decision tree algorithms using Waikato environment for knowledge analysis. The aim is to investigate the performance of data classification for a set of a large data. The algorithms tested are functional tree algorithm, logistic model trees algorithm, REP tree algorithm and best-first decision tree algorithm. The UCI repository will be used to test and justify the performance of decision tree algorithms. Subsequently, the classification algorithm that has the optimal potential will be suggested for use in large scale data.**

***Keywords*—Functional tree algorithm, logistic model trees algorithm, REP tree algorithm, best-first decision tree algorithm.**

I. Introduction

The decision tree algorithms represents a new trend in data mining techniques. Data mining for business intelligence (BI) has also been gathering momentum in recent years. The data mining platform, Waikato Environment for Knowledge Analysis (WEKA) [1] has been a popular for sharing algorithms amongst researchers. Many algorithms have been developed and applied for data classifying and classification.

Bhargavi et al. [2] used fuzzy c-mean (FCM) algorithm for classifying soil data. However, the main difficulty with FCM is how to determine the number of clusters. Storrie-Lombardi et al. [3] applied neural network (NN) for spectral classification of stars. However, the relative importance of potential input variables, long time training process, and interpretative difficulties have often been criticized. Zhang et al. [4] used support vector machines (SVM) to automatic classification, Qu et al. [5] applied SVM for object detection, Williams et al. [6] used SVM for identification of red variable, and Wadadekar [7] used SVM in redshift estimation. Although, the SVM algorithm has high performance in classification and identification problem but the rules obtained by SVM algorithm are hard to understand directly. Moreover, one possible drawback of SVM is its computational cost.

Many algorithms have been performed for data classification, but they limitations. A large scale data set affect the result of classification and algorithms require intensive computing power for training process and data classification. Furthermore, based on experimental work report in the previous work, most of algorithms mentioned above worked on small data set. In this paper, we propose the decision tree algorithms of data mining techniques to help retailers to classification for UCI repository [8]. The aim is to judge the accuracy of different decision tree algorithms namely, functional tree algorithm, logistic model trees algorithm, REP tree algorithm and best-first decision tree algorithm on various data sets. The performance of data classification depends on many factors encompassing test mode, size of data set and different nature of data sets.

II. Data Preparations

The data of UCI repository often presented in a database or spreadsheet and storage in attribute-relation file format (ARFF). Decision tree from WEKA can be easily converted from ARFF into a file in comma separated value (CVS) format as a list of records with commas between items. However, you don't actually have to go through these steps to create CVS format, yourself the explorer can read ARFF spreadsheet files directly.

A. Data Description

The data sets have different characteristics, such as nursery data set in UCI repository, it has 11,025 instances and nine attributes; the attribute are as follow:

- Parents
- Has_nurs
- Form
- Children
- Housing
- Finance
- Social
- Health
- Class

A history of nursery data set is described characteristics are real, therefore, we use the min-max normalization model to transform the attribute's values in a new range, 0 to 1. If numeric attribute is selected one can its maximum, minimum values, mean and standard deviation of that attribute in the dataset.

B. Loading and Filtering Files

Along, the top of the data preparation step, WEKA has file format converter for spreadsheets files with the extension such as, .CSV, .names for C.45 and .data.

The appropriate converter is used based on extension. If WEKA cannot load the extension data, it tries to interpret it as ARFF format.

III. Methods

In this paper, we choose four decision tree algorithms namely, functional tree algorithm, logistic model trees algorithm, REP tree algorithm and best-first decision tree algorithm are used for comparison. A comparison is based on sensitivity, specificity and accuracy by true positive and false positive in confusion matrix. To have a fair comparison between different algorithms, training time in seconds and tree size ratio for each technique on each data set obtained via 10-fold stratified cross validation. The overall methodology followed for fulfilling the goal of this paper is shown in Fig. 1.

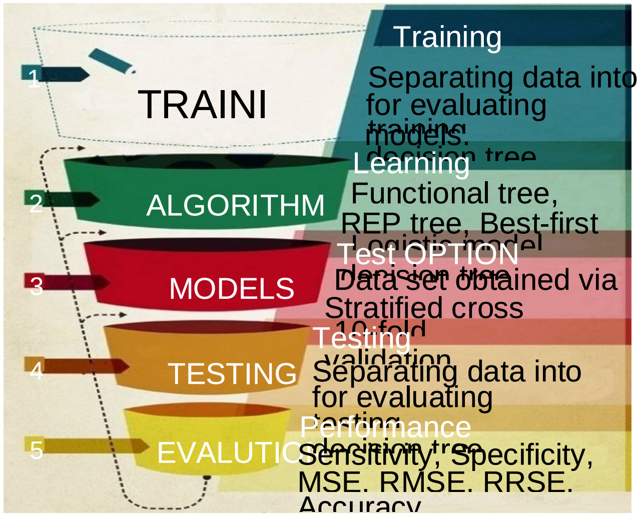

1
TRAINI
Training
Separating data into for evaluating training models.
decision tree
2
ALGORITHM
Learning
Functional tree, REP tree, Best-first Logistic model decision tree
3
MODELS
Test OPTION
Data set obtained via Stratified cross 10-fold validation
4
TESTING
Testing
Separating data into for evaluating testing decision tree
5
EVALUTIO
Performance
Sensitivity, Specificity, MSE. RMSE. RRSE. Accuracy

Fig. 1 The outline of methodology for UCI repository classification.

C. UCI Repository

The UCI repository used in our comparison of decision tree algorithm are shown in Table 1. The data sets shows the number of instance and attribute of each data sets for training and testing algorithms. In the remainder of data sets are referred to using the data number provided in the first column of Table 1.

TABLE I
UCI Repository Details

Num.	Name	Attribute	Instance
1	Nursery	9	11,025
2	Iris	5	150
3	Anneal	39	898
4	Shuttle_trn	10	43,500
5	Voting	17	435
6	Waveform	22	5000
7	Sick	30	2,800

D. Evaluation of Classification Algorithms

The performance of classification algorithms is usually examined by evaluating the sensitivity, specificity, and accuracy of the classification. Sensitivity is the fraction of retrieved instances that are relevant while specificity is the fraction of relevant instant that are retrieved. Accuracy is the overall success rate of the correctly. These values are defined as Eq. (1) – Eq. (3).

$$\text{Sensitivity} = \frac{TP}{TP + FN} \qquad (1)$$

$$\text{Specificity} = \frac{TP}{TP + FP} \qquad (2)$$

$$\text{Acurracy} = \frac{TP + TN}{TP + FP + FN + TN} \qquad (3)$$

All measures can be calculated based on four values, namely True Positive, False Positive, False Negative, and False Positive [9]. These values are described below.

- True Positive (TP) is a number of correctly classified that an instances positive.
- False Positive (FP) is a number of incorrectly classified that an instance is positive.
- False Negative (FN) is a number of incorrectly classified that an instance is negative.
- True Negative (TN) is a number of correctly classified that an instance is negative.

A confusion matrix is another important aspect to be considered, from this matrix classifications can be made. Example of confusion matrix is shown in Figure 2.

```
-------------------------- Confusion Matrix------------------------
       a     b   <-- classified as 4000 instance
     832    48 |    a = Yes
      26  3094 |    b = No
```

Fig. 2 The confusion matrix of 4,000 instance.

From the above result panel the classifications made are as follow, sine number values present in the class variable are two (confusion matrix is m×n matrix) i.e., “a” and “b”, therefore confusion matrix is represented in 2×2 matrix. Here, “a” represent Yes and “b” represent No. Diagonal element represent the correctly classified instances for the class value Yes and No, respectively. For above confusion matrix, TP for class a (Yes) is 832 while FP is 48 whereas, for class b (No), TP is 3094 and FP is 26 (diagonal element of matrix 832+3094 = 3926 represents the correct instances classified and other elements 480+26 = 506 represents the incorrect instances).

Therefore, TP rate equals diagonal element divided by sum of relevant row, while FP rate equals non-diagonal element divided by sum of relevant row (TP rate for class a = 832/(832+48) = 0.94+, FP rate for class a = 26/(26+3094) = 0.008+, TP rate for class b = 3094/(26+3094) = 0.99+, and FN rate for class b = 48/(832+48) = 0.05+).

IV. Classifier and Experimental Results

We have performed classification using functional tree algorithm, logistic model trees algorithm, REP tree algorithm and best-first decision algorithm on UCI repository. The experimental results under the framework of WEKA (Version 3.6.10). All experiment were performed on Duo Core with 1.8GHz CPU and 2G RAM. The experimental results are partitioned into several sub item for easier analysis and evaluation. One the first part, sensitivity (SE), specificity (SP), accuracy (AC), kappa static (KS) and time taken to build model (TTBM) will be partitioned in first table while the second part, we also show the relative mean absolute error (RMAE), root mean squared error (RMSE), relative absolute error (RAE) and root relative squared error (RRSE) for reference and evaluation.

A. Functional Tree Algorithm

Classifier for building 'Functional trees', which are classification trees that could have logistic regression functions at the inner nodes and/or leaves. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values [10]. The example of functional tree algorithm is applied on UCI data set and the confusion matrix is generated for class gender having five possible values are shown in Fig 3.

```
-------------------------- Confusion Matrix ----------------------

    a    b   c    d     e   <-- classified as
 3675   0   0    0     0 |        a = not_recom
    0   0   2    0     0 |        b = recommend
    0   2  273  52     1 |        c = very_recom
    0   0  48  3898  174 |        d = priority
    0   0   0  159  2741 |        e = spec_prior
```

Fig. 3 The confusion matrix of functional tree algorithm.

The results of the functional tree algorithm are shown in Table 2 and Table 3. Table 2 mainly summarizes the result based on sensitivity, specificity, accuracy, kappa static and time taken to build model for classification. Meanwhile, Table 3 shows the results based on relative mean absolute error, root mean squared error, relative absolute error and root relative squared error during classification

TABLE II
Classification Result for Functional Tree Algorithm

Data Name	SE	SP	AC (%)	KS	TTBM (s)
Nursery	0.96	0.96	96.02+	0.94+	11.64
Iris	0.96	0.96	96.66+	0.95+	0.02
Anneal	0.92	0.92	92.87+	0.81+	4.32
Shuttle_trn	0.99	0.99	99.87+	0.99+	46.74
Voting	0.96	0.96	96.78+	0.93+	0.51
Waveform	0.84	0.84	84.66+	0.76+	4.07
Sick	0.97	0.97	97.64+	0.79+	1.68
Average	**0.94**	**0.94**	**94.92+**	**0.88+**	**9.85**

TABLE III
Error Results for Functional Tree Algorithm

Data Name	MAE	RMSE	RAE (%)	RRSE (%)
Nursery	0.01+	0.11+	6.29+	32.26+
Iris	0.03+	0.13+	7.11+	28.49+
Anneal	0.02+	0.14+	19.58+	57.58+
Shuttle_trn	0.00+	0.18+	0.39+	8.06+
Voting	0.03+	0.17+	8.32+	35.55+
Waveform	0.10+	0.30+	24.09+	63.71+
Sick	0.02+	0.14+	23.87+	58.58+
Average	**0.03+**	**0.16+**	**12.80+**	**40.60+**

B. Logistic Model Trees Algorithm

Classifier for building 'logistic model trees', which are classification trees with logistic regression functions at the leaves. The algorithm can deal with binary and multi-class target variables, numeric and nominal attributes and missing values [11]. The example of logistic model trees algorithm is applied on UCI repository and the confusion matrix is generated for class gender having two possible values are shown in Fig 4. The results of the functional tree algorithm are shown in Table 4 and Table 5.

```
-------------------------- Confusion Matrix ----------------------

              a          b   <-- classified as
            153         18 |      a = sick
             18       2611 |      b = negative
```

Fig. 4 The confusion matrix of logistic model trees algorithm.

TABLE IV
Classification Result for Logistic Model Trees Algorithm

Data Name	SE	SP	AC (%)	KS	TTBM (s)
Nursery	0.98	0.98	98.78+	0.98+	533.83+
Iris	0.94	0.94	94.00	0.91	0.54
Anneal	0.95	0.95	95.65	0.89+	65.38
Shuttle_trn	0.97	0.97	97.64+	0.94+	231.36
Voting	0.96	0.96	96.78+	0.93+	4.39
Waveform	0.87	0.87	87.02+	0.80+	102.24
Sick	0.98	0.98	98.71+	0.88+	42.04
Average	**0.95**	**0.95**	**95.51+**	**0.90+**	**139.96+**

TABLE V
ERROR RESULTS FOR LOGISTIC MODEL TREES ALGORITHM

Data Name	MAE	RMSE	RAE (%)	RRSE (%)
Nursery	0.00+	0.69+	1.80+	18.91+
Iris	0.04+	0.15+	9.86+	32.71+
Anneal	0.01+	0.11+	13.11+	43.19+
Shuttle_trn	0.56+	0.18+	12.49+	35.89+
Voting	0.55+	0.16+	11.72+	34.88+
Waveform	0.13+	0.25+	29.85+	53.59+
Sick	0.01+	0.11+	12.86+	46.64+
Average	**0.18+**	**0.23+**	**13.09+**	**37.97%**

C. REP Tree algorithm

REP Tree algorithm is a fast decision tree learner which builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning (with backfitting). Only sorts values for numeric attributes once. Missing values are dealt with by splitting the corresponding instances into pieces (i.e. as in C4.5) [12]. The example of REP Tree algorithm is applied on UCI repository and the confusion matrix is generated for class gender having six possible values are shown in Fig 5. The results of the REP tree algorithm are shown in Table 6 and Table 7.

```
--------------------------- Confusion Matrix -------------------------

          a  b   c     d  e    f  <-- classified as
          1  2  5     0  0   0 |      a = 1
          0  71  28  0  0   0 |      b = 2
          0  10 652  0  20   2 |     c = 3
          0  0   0     0  0   0 |    d = 4
          0  0  12    0  55  0 |     e = 5
          0  0  6     0  0  34 |     f = U
```

Fig. 5 The confusion matrix of REP Tree algorithm.

TABLE VI
CLASSIFICATION RESULT FOR REP TREES ALGORITHM

Data Name	SE	SP	AC (%)	KS	TTBM (s)
Nursery	0.95	0.95	95.92+	0.91+	0.00
Iris	0.94	0.94	94.00	0.91	0.54
Anneal	0.90	0.90	90.53	0.75+	0.06+
Shuttle_trn	0.99	0.99	99.86+	0.99+	0.16+
Voting	0.95	0.94	94.94+	0.89+	0.00
Waveform	0.76	0.76	76.42+	0.64+	0.34
Sick	0.98	0.98	98.42+	0.85+	0.11
Average	**0.92**	**0.92**	**92.87+**	**0.84+**	**0.17+**

TABLE VII
ERROR RESULTS FOR REP TREES ALGORITHM

Data Name	MAE	RMSE	RAE (%)	RRSE (%)
Nursery	0.01+	0.10+	7.33+	29.22+
Iris	0.05+	0.19+	12.67+	41.05+
Anneal	0.04+	0.14+	31.70+	55.11+
Shuttle_trn	0.00+	0.61+	0.16+	8.62+
Voting	0.89+	0.08+	16.94+	43.16+
Waveform	0.19+	0.34+	43.38+	72.77+
Sick	0.02+	0.11+	22.67+	48.90+
Average	**0.17+**	**0.22+**	**13.09+**	**19.26%**

D. Best-First Decision Tree algorithm

Class for building a best-first decision tree algorithm. This class uses binary split for both nominal and numeric attributes. For missing values, the method of ‘fractional’ instances is used [13]. The example of best-first decision tree algorithm is applied on UCI repository and the confusion matrix is generated for class gender having three possible values are shown in Fig 6. The results of the best-first decision tree algorithm are shown in Table 8 and Table 9.

```
--------------------------- Confusion Matrix -------------------------

                 a        b         c      <-- classified as
                   1192   253     212      |      a = 0
                  192  1312   143    |b = 1
                  170   151   1375    |c = 2
```

Fig. 6 The confusion matrix of best-first decision tree algorithm.

TABLE VIII
CLASSIFICATION RESULT FOR BEST-FIRST DECISION TREES ALGORITHM

Data Name	SE	SP	AC (%)	KS	TTBM (s)
Nursery	0.99	0.99	99.20+	0.98+	6.11
Iris	0.94	0.94	94.66+	0.92	0.03
Anneal	0.92	0.92	92.20	0.80+	0.63+
Shuttle_trn	0.99	0.99	99.94+	0.99+	8.15
Voting	0.95	0.95	95.63+	0.90+	0.51
Waveform	0.77	0.77	77.58+	0.66+	5.63
Sick	0.98	0.98	98.78+	0.89+	1.96
Average	**0.93**	**0.93**	**93.99+**	**0.87+**	**3.28+**

TABLE IX
ERROR RESULTS FOR BEST-FIRST DECISION TREES ALGORITHM

Data Name	MAE	RMSE	RAE (%)	RRSE (%)
Nursery	0.00+	0.04+	1.12+	12.43+
Iris	0.04+	0.17+	9.23+	37.20+
Anneal	0.04+	0.14+	31.70+	55.11+
Shuttle_trn	0.00+	0.01+	0.19+	5.29+
Voting	0.06 +	0. 20+	13.89+	41.81+
Waveform	0. 18+	0.36 +	41.28+	76.91+
Sick	0. 01+	0. 10+	13.40+	44.42+
Average	**0.04+**	**0.14+**	**15.83+**	**39.02%**

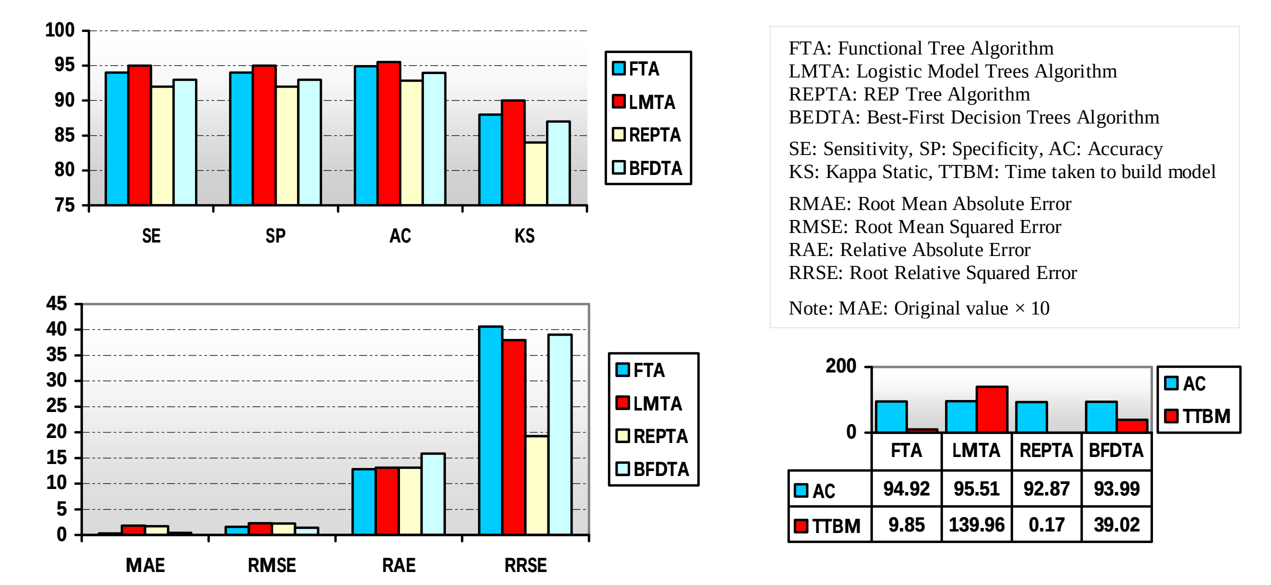
V. CONCLUSIONS

In this paper, we have compared the effectives of the decision tree algorithm namely, functional tree algorithm, logistic model trees algorithm, REP tree algorithm and best-first decision tree algorithm. The achieved performance are compared on the collected UCI repository. Based on the above classifier and experimental results, we can clearly see that highest accuracy belong to the logistic model trees algorithm followed by functional tree algorithm, best-first decision tree algorithm and REP trees algorithm, respectively. The total time to build the model is also a crucial parameter in computing the classification algorithm. In this experimental result, we can see that a REP trees algorithm requires the shortest time which is 0.17 second while logistic model trees algorithm requires the longest time which is 139.96 seconds.

As a conclusion, the best algorithm based on the UCI data is logistic model trees algorithm which an accuracy are 95.51% and the total time to build the model is at 139.96 seconds while the REP tree algorithm has the lowest accuracy are 92.87% and the total time to build the model is at 0.17 seconds. The graphical representations of all the classification result is shown in Fig 7.

ACKNOWLEDGMENT

This paper was supported by the Mahasakham Business School (MBS), Mahasahakham University, Thailand. We also would like to thank UCI repository for the data set used in this experiment.



FTA: Functional Tree Algorithm
LMTA: Logistic Model Trees Algorithm
REPTA: REP Tree Algorithm
BEDTA: Best-First Decision Trees Algorithm

SE: Sensitivity, SP: Specificity, AC: Accuracy
KS: Kappa Static, TTBM: Time taken to build model

RMAE: Root Mean Absolute Error
RMSE: Root Mean Squared Error
RAE: Relative Absolute Error
RRSE: Root Relative Squared Error

Note: MAE: Original value × 10

	FTA	LMTA	REPTA	BFDTA
AC	94.92	95.51	92.87	93.99
TTBM	9.85	139.96	0.17	39.02

Fig. 7 The graphical of all the classification results.

REFERENCES

[1] University of Waikato, New Zealand: WEKA version 3.6.10 (2013) Website: http://www.cs. Waikato.ac.nz/ml/weka.html

[2] P. Bhargavi, and S. Jyoth, “Soil Classification using data mining techniques: A comparative study,” International Journal of Engineering Trends and Technology, pp. 55–59, July. 2011.

[3] M.C. Storrie-Lombardi, A.A. Suchkov and E.L. Winter, “Morphological classification of galaxies by artificial neural network” MNRAS, pp. 8-12, 1992.

[4] Y. Zhang, and Y. Zhao, “Automated clustering algorithm for classification of astronomical objects,” pp. 1113-1121, 2004.

[5] M. Qu, Y. Frank, and J. Jing, “Automated solar flare detection using MLP, RBF, and SVM,” Solar Physics, pp. 157-172, 2003.

[6] S.J. Williams, P.R. Wozniak, and W.T. Vestrand, “Identifying Red Variable in the Northern Sky Variability Survey” pp. 2965-2976, 2004.

[7] S.J. Williams, P.R. Wozniak, and W.T. Vestrand, “Identifying Red Variable in the Northern Sky Variability Survey” pp. 2965-2976, 2004.

[8] Y. Wadadekar, “Estimating photometric redshifts using support vector machine,” pp.79-85, 2009.

[9] D.J. Newman, S. Hettich, C.L. Blake and C.J. Merz, UCI repository of machine leaning databases, University of California, Department of Computer Science, Website: http://www.ics.usi.edu

[10] R. Tina, and S.S. Sherekar, “Performance Analysis of Naïve Bayes and J48 Classification Algorithm for Data Classification,” pp. 256-261, April. 2013.

[11] Joao Gama, “Functional Tree,” Machine Learning, pp. 219–250, 2004.

[12] N. Landwehr, M. Hall, and E. Frank, “Logistic Model Trees,” Machine Learning, pp. 161-205, 2005.

[13] J. Park, T. Hsiao-Rong and C.-C.J. Kuo, “GA-Based Internet Traffic Classificaton Technique for QoS Provisioning,” pp. 251-254, 2006.

[14] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: A statistical view of boosting,” Annals of statistics, pp. 337-407, 2000.