Making Use of Functional Dependencies Based on Data to Find Better Classification Trees

Hyontai Sug,
Dept. of Computer Eng. Dongseo University,
47 Jurye-ro, Sasang-gu, Busan, 47011
Republic of Korea
sht@dongseo.ac.kr

Received: April 20, 2021. Revised: September 6, 2021. Accepted: September 17, 2021. Published: September 20, 2021.

Abstract—For the classification task of machine learning algorithms independency between conditional attributes is a precondition for success of data mining. On the other hand, decision trees are one of the mostly used machine learning algorithms because of their good understandability. So, because dependency between conditional attributes can cause more complex trees, supplying conditional attributes independent each other is very important, the requirement of conditional attributes for decision trees as well as other machine learning algorithms is that they are independent each other and dependent on decisional attributes only. Statistical method to check independence between attributes is Chi-square test, but the test can be effective for categorical attributes only. So, the applicability of Chi-square test is limited, because most datasets for data mining have mixed attributes of categorical and numerical. In order to overcome the problem, and as a way to test dependency between conditional attributes, a novel method based on functional dependency based on data that can be applied to any datasets irrespective of data type of attributes is suggested. After removing highly dependent attributes between conditional attributes, we can generate better decision trees. Experiments were performed to show that the method is effective, and the experiments showed very good results.

Keywords—artificial intelligence, machine learning, classification, decision trees, knowledge modelling, preprocessing, information systems, functional dependency.

I. Introduction

Data mining has shown practicality and utility in various fields, showing several successful use cases [1, 2]. And explainable AI is a hot topic for research nowadays, because, even though their great success, it's hard to understand how neural networks inference logical decisions [3], likewise other high accuracy algorithms such as SVM [4, 5]. On the other hand, because of their structure decision trees are easy to understand. So, decision trees are considered one of the most important machine learning algorithms [6]. But, as the size of training data become large, which is very common in data mining tasks, the size of tree also tends to be large, as a result, understandability of the tree becomes worse. A simple method to generate smaller decision tree is to stop growing the tree in a certain depth. But, stopping at a certain depth of the tree may not be good enough with respect to the accuracy of the tree, because there would be many instances that are not classified fully [7].

The target datasets of decision trees consist of conditional attributes and decisional attributes. The requirement of conditional attributes is that they are independent each other and dependent on decisional attributes only, which is true for other machine learning algorithms also. If we have some dependency between conditional attributes, we may have more complex trees [8]. So, it is recommended to get rid of dependency between conditional attributes before we generate decision trees [9]. Statistical method to check dependency between attributes is Chi-square test. But, we can apply the test for nominal or categorical attributes only, and each cell in the contingency table needs at least five instances for more than 80% of the cells [10]. So, it may not be possible to do the independency test based on Chi-square test before we apply decision tree algorithms to the target datasets of which attributes usually consist of numerical and nominal attributes mixed.

On the other hand, functional dependency in relational databases is essential to ensure data integrity in relations, because we can check possible data inconsistency problems by confirming whether a relation schema is at least in the third normal form or Boyce-Codd normal form [11]. Designing right structure of relation schema requires the designer of the schema to have exact knowledge about the domain of target databases and functional dependency for each attribute in the relation schema. But, human designers might make some mistakes when they define relation schema, so that the related relations could have some duplicate data. Duplicate data in relations may cause data inconsistency problem if we miss updating any of them. Incorrect design of relation schema may raise some other anomalies, like not being able to timely updates, or loss of information for sole tuples after delete operations [12]. As a result, a lot of research has been done to discover functional dependencies based on stored data in relations or data sets in tabular form, and we may use the information of found functional dependencies to improve the structure of relation schema. Because we can have $2^m - 1$ combinations of attributes for a table having m attributes, the related algorithms try to find functional dependencies as efficiently as possible. There are several polynomial time algorithms suggested; top-down [13, 14], bottom-up [15], and hybrid algorithms [16]. The found functional dependencies can represent the relationship between attributes in the table.

Therefore, in order to find simpler and better decision trees we want to remove highly dependent attributes in conditional attributes first before we generate decision trees. As a way to do the task we try to do functional dependency test first to find closely related attributes, and use the information to get rid of redundant attributes to obtain smaller but still accurate decision trees.

II. RELATED WORK

Over-fitting and under-fitting in machine learning is a hard-to-solve problem [17]. In conventional decision tree algorithms tree size is determined by two factors – stopping criteria and pruning. If we apply stopping criteria early enough, we may have small and under–fitted decision trees. On the other hand, if we apply stopping criteria as late as possible, we may have large decision trees that are over–fitted to the training data set. After generating an over-fitted tree, we may apply pruning methods to achieve more generalization of the tree. The over-fitted tree is made into a smaller tree by removing sub–branches that are not contributing to the generalization. The pruning method was applied in CART [18], as well as in C4.5 [19], which are the two most well-known decision tree algorithms [20]. It has been shown that employing the pruning can improve the performance of generalization in many experiments.

Feature selection is also an important method to generate simpler trees [21]. Principle component analysis (PCA) is a well-known feature selection method [22]. But, because PCA is mainly designed for numerical attributes, it is not easy to apply PCA for datasets having categorical and numerical attributes together. Note that most datasets for data mining have the two kinds of attributes together. Filter methods [23] try to calculate dependency of attributes to class labels, for example, using Pearson's correlation coefficient and rank each attributes, so that they can remove most irrelevant attributes with respect to class labels. Wrapper method [24] contains a target machine learning algorithm in the wrapper, and try to find a best subset of attributes by running the algorithm, and supplies or removes attributes one by one until the best result is found. Supplying or removing attributes can be done exhaustively or heuristically. Because of computational nature, wrapper methods are not easy to apply for very large datasets or compute-intensive machine learning algorithms. Moreover, if the size of datasets is not large enough or used machine learning algorithm is deterministic, like decision tree algorithms, the method cannot avoid over-fitting problem [25]. Embed methods try select to subsets of attributes during building machine learning models [26]. Good point of embedded methods is less compute-intensive than the wrapper approach, and a week point is dependency on training data so that instability becomes bigger especially for relatively small training datasets [27, 28].

III. PROBLEM SOLUTION

The definition of functional dependency based on data can be defined as follows [12].

Definition 1. Let **r** be a relation over the set of attributes U, and X, Y be any subset of U. Then Y is functionally dependent on X, $X \rightarrow Y$, if and only if each X value in **r** is associated with precisely one Y value. □

If we try to find functional dependencies (FD) based on data, we may find two or more equivalent functional dependencies.

Definition 2. Let FD_1 and FD_2 are two sets of functional dependency for a relation **r**. First, if all functional dependencies of FD_1 can be derived from functional dependencies in FD_2, we can say that $FD_2 \supset FD_1$. Second, if all functional dependencies of FD_2 can be derived from functional dependencies in FD_1, we can say that $FD_1 \supset FD_2$. If the first and second fact are true, then FD_1 and FD_2 are equivalent. □

On the other hand, a relation may not be mature or large enough to contain all the possible values in the domain of attributes, so we need the concept of relation variable. A relation variable is a symbol that can have different values for its attributes at different time, and a relation or relation value is a particular state of the relation variable. If we expand definition 1 for relation variable, we have the following definition 3.

Definition 3. Let **R** be a relation variable over the set of attributes U, and X, Y be any subset of U. Then Y is functionally dependent on X, $X \rightarrow Y$, if and only if every possible X value in **R** is associated with precisely one possible Y value. □

If we have very large relations, it is highly possible that the relations approximate corresponding relation variables because they contain many data. So, the idea of trying to find functional dependencies efficiently from data has attracted many researchers' attention and most researches want to find functional dependencies from datasets in tabular form [29]. As a result, an open source software called FDtool is available [30]. FDTool is a Python based open source software to mine functional dependencies and candidate keys in tabular datasets. Note that the format of relations is in tabular form. The difference between relations and datasets in tabular form is that the latter can have many functional dependencies in them because they may not be normalized or may be less normalized.

A. Suggested Method

We want to check whether a given dataset which has conditional and decisional attributes for data mining has functional dependencies between conditional attributes. In order to check them, we use FDtool, then, if some functional dependencies are found, we try to eliminate the set of attributes that are dependent to other attributes before we supply the dataset to generate decision trees. The details of procedure is as follows:

PROCEDURE:
INPUT: a dataset D in tabular form.
OUTPUT: a decision tree
BEGIN
1. Find functional dependencies based on data in conditional attributes using FDtool;

/* Make a table having the information of {frequency in LHS, frequency in RHS, (frequency in RHS - Frequency in LHS), (frequency in RHS/Frequency in LHS)} */

2. **For** each conditional attribute **Do**
   Calculate {frequency in LHS, frequency in RHS, (frequency in RHS) – (Frequency in LHS), (frequency in RHS)/(frequency in LHS)};
   End For;
3. Select highly dependent attributes to other attributes and let the set of attributes be A;
4. **For** all subsets S of A except Ø **Do**
   Remove the columns of attributes in S from D;
   Generate decision trees;
   End For;
5. Select the best decision tree from the result of 4.

END.

In the procedure, LHS and RHS means the left hand side and the right hand side of the found functional dependencies respectively. Note that attributes in RHS of functional dependencies role as dependent attributes, and LHS as independent attributes. So, the meaning of the ratio, (frequency in RHS divided by frequency in LHS), is the ratio of the role as dependent attribute and independent attribute in the found functional dependencies. Moreover, the meaning of difference between frequency in RHS and LHS is the amount of positive or negative role of the attribute as independent attributes in the found functional dependencies. When we select highly dependent attributes, we use the calculated information on the whole. Please see the experiments for details.

IV. EXPERIMENTs

Two datasets, called adult and bank dataset in UCI machine learning repository were used for our experiments [31]. The datasets consist of several conditional attributes and one decisional attribute. We want to find functional dependencies between conditional attributes, and we expect many functional dependencies based on stored data in each dataset.

A. Adult Dataset

The task of adult dataset is to predict whether income exceeds $50K/year based on census income data. The data set has 48,842 records and has 14 conditional attributes and one decisional attribute, named 'class' having two different values, >50K or <=50K. There are 11,687 records having the class value of >50K, and 37,155 records having the class value of <=50K. The 14 conditional attributes consist of numerical and categorical attributes as in table 1. Categorical attributes have limited number of nominal values, while numerical attributes do not.

Table 1. Conditional attributes of Adult dataset

attribute	values
age	numeric
workclass	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked
fnlwgt	numeric
education	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool
education-num	numeric
marital-status	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
occupation	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces
relationship	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
race	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
sex	Female, Male
capital-gain	numeric
capital-loss	numeric
hours-per-week	numeric
native-country	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands

1) *Checking Functional Dependencies for Adult Dataset*

38 functional dependencies in the conditional attributes were found. Some examples of the found functional dependencies are as follows:

{ education} -> { educationNum}
{ educationNum} -> { education}
{age, relationship, education, fnlwgt} -> { sex}
{age, relationship, HoursPerWeek, fnlwgt} -> { sex}
. . . . . .

. . . . . .
{ sex, age, workclass, education, capitalLoss, fnlwgt, nativeCountry} -> { race}
{ relationship, age, workclass, capitalLoss, marital-status, fnlwgt, nativeCountry} -> { race}
{ relationship, workclass, education, HoursPerWeek, marital-status, fnlwgt, nativeCountry} -> { race}

There is an equivalency:
{ education} <-> { educationNum}

Based on the found functional dependencies between conditional attributes, we can calculate the frequency of each attributes in left and right hand side of the found functional dependencies and other values as in the table 2.

Table 2. Frequency of attributes in the found functional dependencies in adult dataset

attribute	frequency in LHS	frequency in RHS	f.RHS-f.LHS	f.RHS/ f.LHS
race	1	26	25	26
sex	4	10	6	2.5
educationNum	1	1	0	1
education	23	1	-22	0.04
fnlwgt	36	0	-36	0
age	28	0	-28	0
occupation	20	0	-20	0
workclass	18	0	-18	0
marital-status	18	0	-18	0
relationship	17	0	-17	0
hoursPerWeek	16	0	-16	0
nativeCountry	16	0	-16	0
capitalLoss	8	0	-8	0
capitalGain	2	0	-2	0

In the table f.RHS and f.LHS means frequency in RHS and frequency in LHS respectively. The attributes having zero frequency in the RHS of the found functional dependency means that the attributes have no effect in dependency in the conditional attributes, so that we don't need to consider as candidates to remove. Based on the value of (the frequency of RHS – the frequency of LHS), and the ratio between the frequency of RHS divided by the frequency of LHS (that is, f.RHS/f.LHS), we may choose three candidate attributes to remove, race, sex, and educationNum, whose corresponding values are {25, 26}, {6, 2.5} and {0, 1} respectively. Because two attributes educationNum and education make an equivalent functional dependency, we try to remove one of them alternately in order to see the effect of removing in the decision trees.

2) Generating Decision Trees

Table 3 shows the property of decision tree generated by J4.8 which is Java version of C4.5 in Weka machine learning package [32] from the original adult dataset. C4.5 is one of the most popular decision tree algorithms [20]. All experiments are performed in 10-fold cross-validation.

Table 3. Decision tree from adult dataset

Number of leaves	696		
Size of the tree	911		
Accuracy	86.0428%		
Confusion matrix		Predicted >50K	Predicted ≤50K
	Actual >50K	6975	4712
	Actual ≤50K	2105	35050

We try to generate decision trees for dataset having select attributes only. We'll try to remove attributes from the original dataset in the following order; educationNum, education, race, sex, {educationNum, race}, {education, race}, {educationNum, sex}, {education, sex}, {educationNum, race, sex}, {education, race, sex}.

Table 4 shows the property of decision tree generated from the adult dataset of which attribute educationNum is omitted.

Table 4. Decision tree from adult dataset – educationNum attribute

Number of leaves	396		
Size of the tree	547		
Accuracy	85.967%		
Confusion matrix		Predicted >50K	Predicted ≤50K
	Actual >50K	6853	4834
	Actual ≤50K	2020	35135

Table 5 shows the property of decision tree generated from the adult dataset of which attribute education is omitted.

Table 5. Decision tree from adult dataset – education attribute

Number of leaves	571		
Size of the tree	838		
Accuracy	86.0735%		
Confusion matrix		Predicted >50K	Predicted ≤50K
	Actual >50K	7023	4664
	Actual ≤50K	2138	35017

Omitting attribute education has less effect in reducing the size of the tree. Note that the attribute occurs more often than attribute educationNum in LHS of the found functional dependencies as we see in Table 2. Table 6 shows the property of decision tree generated from the adult dataset of which attribute race is omitted.

Table 6. Decision tree from adult dataset – race attribute

Number of leaves	622
Size of the tree	810
Accuracy	86.0796%

Confusion matrix		Predicted >50K	Predicted ≤50K
	Actual >50K	6936	4751
	Actual ≤50K	2048	35107

Table 7 shows the property of decision tree generated from the adult dataset of which attribute sex is omitted.

Table 7. Decision tree from adult dataset – sex attribute

Number of leaves	662		
Size of the tree	845		
Accuracy	86.9%		
Confusion matrix		Predicted >50K	Predicted ≤50K
	Actual >50K	6958	4729
	Actual ≤50K	2104	35051

Table 8 shows the property of decision tree generated from the adult dataset of which attribute educationNum and race are omitted.

Table 8. Decision tree from adult dataset – educationNum and race attribute

Number of leaves	369		
Size of the tree	506		
Accuracy	86.0018%		
Confusion matrix		Predicted >50K	Predicted ≤50K
	Actual >50K	6875	4812
	Actual ≤50K	2025	35130

Table 9 shows the property of decision tree generated from the adult dataset of which attribute education and race are omitted.

Table 9. Decision tree from adult dataset – education and race attribute

Number of leaves	517		
Size of the tree	775		
Accuracy	86.1533%		
Confusion matrix		Predicted >50K	Predicted ≤50K
	Actual >50K	6932	4755
	Actual ≤50K	2008	35147

Table 10 shows the property of decision tree generated from the adult dataset of which attribute educationNum and sex are omitted.

Table 10. Decision tree from adult dataset – educationNum and sex attribute

Number of leaves	405		
Size of the tree	553		
Accuracy	85.926%		
Confusion matrix		Predicted >50K	Predicted ≤50K
	Actual >50K	6849	4838
	Actual ≤50K	2036	35119

Table 11 shows the property of decision tree generated from the adult dataset of which attribute education and sex are omitted.

Table 11. Decision tree from adult dataset – education and sex attribute

Number of leaves	482		
Size of the tree	699		
Accuracy	86.0591%		
Confusion matrix		Predicted >50K	Predicted ≤50K
	Actual >50K	7002	4685
	Actual ≤50K	2124	35031

Table 12 shows the property of decision tree generated from the adult dataset of which attribute educationNum, race and sex are omitted.

Table 12. Decision tree from adult dataset – educationNum, race, and sex attribute

Number of leaves	381		
Size of the tree	516		
Accuracy	85.9506%		
Confusion matrix		Predicted >50K	Predicted ≤50K
	Actual >50K	6836	4851
	Actual ≤50K	2011	35144

Table 13 shows the property of decision tree generated from the adult dataset of which attribute education, race and sex are omitted.

Table 13. Decision tree from adult dataset – education, race, and sex attribute

Number of leaves	407		
Size of the tree	597		
Accuracy	86.1328%		
Confusion matrix		Predicted >50K	Predicted ≤50K
	Actual >50K	6915	4772

	Actual ≤50K	2001	35154

The following table 14 summarizes the experiments with respect to accuracy and size of trees as attributes are dropped from the dataset before generating decision trees.

Table 14. The summary of experiments of decision tree from adult dataset

Dropped attributes	Accuracy (%)	Tree size
none	86.0428	911
educationNum	85.967	547
education	86.0735	838
race	86.0796	810
sex	86.9	845
educationNum, race	**86.0018**	**506**
education, race	86.1533	775
educationNum, sex	85.926	553
education, sex	86.0591	699
educationNum, race, sex	85.9506	516
education, race, sex	86.1328	597

As we see in the table, dropping two attributes, educationNum and race, generates the tree of 506/911=56% size with similar accuracy compared to the tree from the original dataset, which enhances comprehensibility a lot without losing accuracy. Comparing the confusion matrix of the two trees in table 8 (from dropping the two attributes, educationNum and race) and table 3 (from original dataset), we have the loss of -100 cases of correct prediction of '>50K' while the gain of +80 cases of correct prediction of '≤50K', which causes a slight prediction rate change in the trees.

B. Bank Dataset

The purpose of bank dataset is to predict if the client will subscribe a term deposit for direct marketing campaigns of a Portuguese banking institution. The data set has 4,521 records and has 16 conditional attributes and one decisional attribute, named 'y', having two different values, yes or no, which means the client subscribed a term deposit or not. There are 521 records having class value of yes, and 4,000 records having class value of no. The 16 conditional attributes have variety of values as in table 15.

Table 15. Conditional attributes of bank dataset

attribute	values
age	numeric
job	admin., unknown, unemployed, management, housemaid, entrepreneur, student, blue-collar, self-employed, retired, technician, services
marital	divorced, married, single, unknown
education	basic.4y, basic.6y, basic.9y, high.school, illiterate, professional.course, university.degree, unknown
default	no, yes, unknown
housing (housing loan)	no, yes, unknown
loan (personal loan)	no, yes, unknown
contact	cellular, telephone
month(last contact month of year)	jan, feb, mar, ..., nov, dec
day_of_week	mon, tue, wed, thu, fri
Duration(last contact duration, in seconds)	numeric
Campaign(number of contacts performed during this campaign)	numeric
Pdays(number of days that passed by after the client was last contacted)	numeric
Previous(number of contacts performed before this campaign)	numeric
Poutcome(outcome of the previous marketing campaign)	failure, nonexistent, success
emp.var.rate(employment variation rate)	numeric
cons.price.idx(consumer price index)	numeric
cons.conf.idx(consumer confidence index)	numeric
euribor3m(euribor 3 month rate)	numeric
nr.employed(number of employees)	numeric

1) Checking Functional Dependencies for Bank Dataset

1,040 functional dependencies were found in the conditional attributes based on the dataset. That is, we found 21, 132, 331, 261, 224, 42, 20, 9 functional dependencies of length 4, 5, 6, 7, 8, 9, 10, 11 respectively. Some examples of found functional dependencies are as follows:

{balance, poutcome, duration} -> {previous}
{balance, age, duration} -> {default}
. . . . . .
. . . . . .
{balance, poutcome, campaign, default, duration, job, contact} -> {education}
{poutcome, campaign, age, housing, marital, duration, month} -> {pdays}
. . . . . .
. . . . . .
{education, campaign, age, marital, job, contact, duration, loan, housing, previous} -> {day}
{education, default, marital, job, contact, duration, loan, housing, day, previous} -> {month}

Based on the found functional dependencies between conditional attributes, we can calculate the frequency of each attribute in the left and right hand side of the found functional dependencies and other values as in the table 16.

Table 16. Frequency of attributes in the found functional dependencies in bank dataset

attribute	frequency in LHS	frequency in RHS	f.RHS-f.LHS	f.RHS/ f.LHS
pdays	174	203	29	1.167
poutcome	206	122	-84	0.592
previous	262	103	-159	0.393
default	48	101	53	2.104
contact	274	90	-184	0.328
housing	331	71	-260	0.215
loan	271	68	-203	0.251
month	400	63	-337	0.158
marital	323	50	-273	0.155
education	366	49	-317	0.134
job	421	30	-391	0.071
day	519	26	-493	0.050
campaign	517	16	-501	0.301
balance	154	12	-142	0.078
age	521	11	-510	0.021
duration	723	11	-712	0.015

Based on the value of (the frequency of RHS – the frequency of LHS), and the ratio between the frequency of RHS and the frequency of LHS (that is, f.RHS/f.LHS), we may choose two candidate attributes to remove, default and pdays, whose corresponding values are {53, 2.104} and {29, 1.167} respectively. Additionally, we may try to remove other two attributes, poutcome and previous, whose corresponding values are {-84, 0.592} and {-159, 0.393} respectively, because even though their values of (the frequency of RHS – the frequency of LHS) are negative, which means that there are more attributes that role as independent attributes rather than dependent attributes, but the values of the ratio, (the frequency of RHS)/(the frequency of LHS), are the third and fourth respectively, and frequency of RHS is relatively high. Note that attributes in RHS of functional dependencies role as dependent attributes.

2) Generating Decision Trees

Table 17 shows the property of decision tree from the original bank dataset. All experiments are performed with 10-fold cross-validation.

Table 17. Decision tree from bank dataset

Number of leaves	104		
Size of the tree	146		
Accuracy	88.8963%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	187	334
	Actual no	168	3832

We'll try to remove attributes from the original dataset in the following order; pdays, default, poutcome, previous, {pdays, default}, {pdays, poutcome}, {pdays, previous}, {default, poutcome}, {default, previous}, {poutcome, previous}, {pdays, default, poutcome}, {pdays, default, previous}, {pdays, poutcome, previous}, {default, poutcome, previous}, {pdays, default, poutcome, previous}. Table 18 shows the property of decision tree generated from the bank dataset of which attribute pdays is omitted.

Table 18. Decision tree from bank dataset – pdays attribute

Number of leaves	102		
Size of the tree	142		
Accuracy	88.852%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	185	336
	Actual no	166	3832

Table 19 shows the property of decision tree generated from the bank dataset of which attribute default is omitted.

Table 19. Decision tree from bank dataset – default attribute

Number of leaves	102		
Size of the tree	140		
Accuracy	88.9847%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	186	335
	Actual no	163	3837

Table 20 shows the property of decision tree generated from the bank dataset of which attribute poutcome is omitted.

Table 20. Decision tree from bank dataset – poutcome attribute

Number of leaves	92		
Size of the tree	136		
Accuracy	88.4318%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	175	346
	Actual no	177	3823

Table 21 shows the property of decision tree generated from the bank dataset of which attribute previous is omitted.

Table 21. Decision tree from bank dataset – previous attribute

Number of leaves	104
Size of the tree	146
Accuracy	88.8741%

Confusion matrix		Predicted yes	Predicted no
	Actual yes	188	333
	Actual no	170	3830

Table 22 shows the property of decision tree generated from the bank dataset of which attribute pdays and default are omitted.

Table 22. Decision tree from bank dataset – pdays and default attribute

Number of leaves	100		
Size of the tree	136		
Accuracy	88.9405%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	184	337
	Actual no	163	3837

Table 23 shows the property of decision tree generated from the bank dataset of which attribute pdays and poutcome are omitted.

Table 23. Decision tree from bank dataset – pdays and poutcome attribute

Number of leaves	102		
Size of the tree	159		
Accuracy	88.3433%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	172	349
	Actual no	178	3822

Table 24 shows the property of decision tree generated from the bank dataset of which attribute pdays and previous are omitted.

Table 24. Decision tree from bank dataset – pdays and previous attribute

Number of leaves	104		
Size of the tree	144		
Accuracy	88.8299%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	186	335
	Actual no	170	3830

Table 25 shows the property of decision tree generated from the bank dataset of which attribute default and poutcome are omitted.

Table 25. Decision tree from bank dataset – default and poutcome attribute

Number of leaves	87		
Size of the tree	126		
Accuracy	88.4318%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	175	346
	Actual no	177	3823

Table 26 shows the property of decision tree generated from the bank dataset of which attribute default and previous are omitted.

Table 26. Decision tree from bank dataset – default and previous attribute

Number of leaves	102		
Size of the tree	140		
Accuracy	88.9405%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	186	335
	Actual no	165	3835

Table 27 shows the property of decision tree generated from the bank dataset of which attribute poutcome and previous are omitted.

Table 27. Decision tree from bank dataset – poutcome and previous attribute

Number of leaves	101		
Size of the tree	144		
Accuracy	88.476%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	172	349
	Actual no	172	3828

Table 28 shows the property of decision tree generated from the bank dataset of which attribute pdays, default and poutcome are omitted.

Table 28. Decision tree from bank dataset – pdays, default, and poutcome attribute

Number of leaves	104
Size of the tree	149
Accuracy	88.3433%

		Predicted yes	Predicted no
Confusion matrix	Actual yes	173	348
	Actual no	179	3821

Table 29 shows the property of decision tree generated from the bank dataset of which attribute pdays, default and previous are omitted.

Table 29. Decision tree from bank dataset – pdays, default, and previous attribute

Number of leaves	102		
Size of the tree	138		
Accuracy	88.8963%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	184	337
	Actual no	165	3835

Table 30 shows the property of decision tree generated from the bank dataset of which attribute pdays, poutcome and previous are omitted.

Table 30. Decision tree from bank dataset – pdays, poutcome, and previous attribute

Number of leaves	91		
Size of the tree	133		
Accuracy	89.1376%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	192	329
	Actual no	162	3838

Table 31 shows the property of decision tree generated from the bank dataset of which attribute default, poutcome and previous are omitted.

Table 31. Decision tree from bank dataset – default, poutcome, previous attribute

Number of leaves	96		
Size of the tree	134		
Accuracy	88.4539%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	171	350
	Actual no	172	3828

Table 32 shows the property of decision tree generated from the bank dataset of which attribute pdays, default, poutcome and previous are omitted.

Table 32. Decision tree from bank dataset – pdays, default, poutcome, and previous attribute

Number of leaves	86		
Size of the tree	123		
Accuracy	89.228%		
Confusion matrix		Predicted yes	Predicted no
	Actual yes	192	329
	Actual no	158	3842

The following table 33 summarizes the experiments with respect to accuracy and size of trees as attributes are dropped before generating decision trees.

Table 33. The summary of experiments of decision tree from bank dataset

Dropped attributes	Accuracy (%)	Tree size
none	88.8963	146
pdays	88.852	142
default	88.9847	140
poutcome	88.4318	136
previous	88.8741	146
pdays, default	88.9405	136
pdays, poutcome	88.3433	159
pdays, previous	88.8299	144
default, poutcome	88.4318	126
default, previous	88.9405	140
poutcome, previous	88.476	144
pdays, default, poutcome	88.3433	149
pdays, default, previous	88.8963	138
pdays, poutcome, previous	89.1376	133
default, poutcome, previous	88.4539	134
pdays, default, poutcome, previous	**89.228**	**123**

As we see in the table, dropping the four attributes, {pdays, default, poutcome, and previous}, generates the best result with respect to tree size as well as accuracy. Comparing the confusion matrix of the two trees in table 32 (from dropping four attributes, pdays, default, poutcome, and previous) and table 17 (from original dataset), we have the gain of +5 cases of correct prediction of 'yes' while the gain of +10 cases of correct prediction of 'no', that cause better accuracy of the trees.

V. Conclusion

When we do data mining task of classification using machine learning algorithms, independence between conditional attributes in datasets is a precondition for the success of the task. And, because their understandability is very good, decision trees that are used for classification tasks are considered one of the most important machine learning algorithms. But, as the size of training data becomes large, which occurs often for data mining task, the size of tree also tends to be large, as a result, the understandability of the tree becomes worse. Moreover, if we have some dependency or lack of independency between conditional attributes, we can have more complex trees. So, it is recommended to get rid of dependency between conditional attributes before we generate decision trees. Chi-square test is well-known statistical method to check dependency between attributes. But, its applicability is limited to categorical attributes only, while target datasets of data mining usually consist of categorical and numeric attributes mixed. So, in order to overcome the problem, and as a way to do independence test between conditional attributes irrespective of the type of attributes, a novel functional dependency-based method is suggested. Experiments based on two real world datasets were done using open source software called FDtool to find functional dependencies based on data, and showed very good results. Future works may be the improvement of FDtool. Because the tool is based on polynomial time algorithm, it may take some long time to find all functional dependencies especially if the size of datasets is very large consisting of millions of records. Note that because our functional dependencies are based on data, the more data the better. So, parallelization of it is desirable.

Acknowledgment

This work was supported by Dongseo University, "Dongseo Frontier Project" Research Fund of 2020.

References

[1] Z.A. Al-sai, R. Abdullah, M.H. Husin, "Critical Success Factors for Big Data: A Systematic Literature Review," IEEE Access, vol. 8, pp. 118940-118956, 2020.
[2] M. Sharma, S. Mahapatra, A. Shankar, X. Wang, "Predicting the Utilization of Mental Health Treatment with Various Machine Learning Algorithms," WSEAS Transactions on Computers, vol. 19, pp. 285-291, 2020.
[3] P. Linardatos, V. Papastefanopoulos, S. Kotsiantis, "Explainable AI: A Review of Machine Learning Interpretability Methods," Entropy, vol. 23, no. 18, https://dx.doi.org/10.3390/e23010018, 2021.
[4] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," Neurocomputing, vol. 408, pp. 189-215, 2020.
[5] A. Abed, J. Yuan, L. Li, "Based SVM Distinct Stages Framework Data Mining Technique Approach for Text Extraction," WSEAS Transactions on Information Science and Applications, vol. 16, pp. 100-110, 2019.
[6] Y. Song, Y. Lu, "Decision tree methods: applications for classification and prediction," Shanhai Archives of Psychiatry, vol. 27, no. 2, pp. 130-135, 2015.
[7] L. Rokach, O. Maimon, "Decision Trees," In Data Mining and Knowledge Discovery Handbook, O. Maimon, L. Rokach Ed., 2005, pp. 165-192.
[8] J. Lee, A new approach of top-down induction of decision trees for knowledge discovery, PhD thesis, Iowa State University, 2008.
[9] H. Lou, L. Wang, D. Duan, C. Yang, M. Mammadov, "RDE: A novel approach to improve the classification performance and expressivity of KDB", PLoS ONE, vol. 13, no. 7, https://doi.org/10.1371/journal.pone.0199822, 2018.
[10] SPSS Tutorial: Chi-square test of independence,https://libguides.library.kent.edu/spss/chisquare, 2021.
[11] R. Elmasri and S.B. Navathe, Fundamentals of Database Systems, 7th ed., Pearson, 2017.
[12] C.J. Date, Introduction to Database Systems, 8th ed., Pearson, 2003.
[13] T. Papenbrock and F. Naumann, "A Hybrid Approach to Functional Dependency Discovery," Proceedings of the 2016 International Conference on Management Data, pp. 821-833, 2016.
[14] Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen, "TANE: An efficient algorithm for discovering functional and approximate dependencies," The Computer Journal, vol. 42, no. 2, pp. 100-111, 1999.
[15] L. Caruccio, S. Cirillo, V. Deufemia, and G. Polese, "Incremental Discovery of Functional Dependencies with a Bit-vector Algorithm," Proceedings of the 27th Italian Symposium on Advanced database Systems, pp. 146-157, 2019.
[16] J. Liu, J. Li, C. Liu, and Y. Chen, "Discover dependencies from data – a review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 2, pp. 251-264, 2012.
[17] D. Bashir, G.D. Montañez, S. Sehra, P.S. Segura, J. Lauw, "An Information-Theoretic Perspective on Overfitting and Underfitting," In: AI 2020: Advances in Artificial Intelligence, M. Gallagher, N. Moustafa, E. Lakshika Ed.,Lecture Notes in Computer Science, vol. 12576, Springer, 2020, https://doi.org/10.1007/978-3-030-64984-5_27
[18] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, Chapman and Hall/CRC, 1984.
[19] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Inc., 1993.
[20]] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z. Zhou, M. Steinbach, D.J. Hand, D. Steinberg, "Top 10 algorithms in data mining," Knowledge and Information System, vol. 14, pp. 1-37, 2008.
[21] G. Chandrashekar, F. Sahin, "A survey on feature selection methods," Computers and Electrical Engineering, vol. 40, pp.16-28, 2014.
[22] I.T. Jolliffe, J. Cadima, "Principle component analysis: a review and recent developments," Philosophical Transactions of the Royal Society A, https://doi.org/10.1098/rsta.2015.0202, 2016.
[23] A. Bommert, X. Sun, B. Bischl, J. Rahnenfürer, M. Lang, "Benchmark for filter methods for feature selection in high-dimensional classification data," Computational

Statistics and Data Analysis, vol. 143, 2020, https://doi.org/10.1016/j.csda.2019.106839.

[24] N. El Aboudi and L. Benhlima, "Review on wrapper feature selection approaches," 2016 International Conference on Engineering & MIS (ICEMIS), pp. 1-5, 2016, DOI: 10.1109/ICEMIS.2016.7745366.

[25] J. Loughrey, "Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets," The Twenty-fourth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Queens' College, Cambridge, UK, 13-15 December 2004, DOI:10.1007/1-84628-102-4_3

[26] D.A.A.G. Singh, S.A.A. Balamurugan, E.J. Leavline, "Literature Review on Feature Selection Methods for High-Dimensional Data," International Journal of Computer Applications, vol. 136, no. 1, pp. 9-17, 2016.

[27] U.M. Khaire, R. Dhanalakshmi, "Stability of feature selection algorithm: A review," Journal of King Saud University – Computer and Information Sciences, 2019, https://doi.org/10.1016/j.jksusi.2019.06.012.

[28] P. Yang, B.B. Zhou, J.Y. Yang, A.Y. Zomaya, "Stability of Feature Selection Algorithms and Ensemble feature Selection Methods in Bioinformatics," In: Biological Knowledge Discovery Handbook, M. Elloumi, A.Y. Zomaya Ed., Wiley Online Library, 2013, https://doi.org/10.1002/9781118617151.ch14.

[29] J. Liu, J. Li, C. Liu, and Y. Chen, "Discover dependencies from data – a review," IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 2, pp. 251-264, 2012.

[30] M. Buranosky, E. Stellnberger, E. Pfaff, D. Diaz-Sanchez, C. Ward-Caviness, FDTool: a Python application to mine for functional dependencies and candidate keys in tabular form [version 2; peer review: 2 approved], F1000Research 2019, 7:1667, https://doi.org/10.12688/f1000research.16483.2.

[31] D. Dua and C. Graff, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml] Irvine, CA, University of California, School of Information and Computer Science, 2019.

[32] E. Frank, M.A. Hall, I.H. Witten, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, Fourth Edition, 2016.

Hyontai Sug received the B.S. degree in Computer Science and Statistics from Busan National University, Busan, Korea, in 1983, the M.S. degree in Computer Science from Hankuk University of Foreign Studies, Seoul, Korea, in 1986, and the Ph.D. degree in Computer and Information Science & Engineering from University of Florida, Gainesville, FL, USA in 1998. Currently, he is a Professor of the Department of Computer Engineering of Dongseo University, Busan, Korea from 2001. From 1999 to 2001, he was a full time Lecturer of Pusan University of Foreign Studies, Busan, Korea. He was a Researcher of Agency for Defense Development, Korea from 1986 to 1992. His areas of research include data mining and database applications.