

# IMPLEMENTATION OF C4.5 ALGORITHM TO EVALUATE THE CANCELLATION POSSIBILITY OF NEW STUDENT APPLICANTS AT STMIK AMIKOM YOGYAKARTA

Kusrini<sup>1</sup>, Sri Hartati<sup>2</sup>

<sup>1</sup> STMIK AMIKOM Yogyakarta, Jl. Ringroad Utara Condong Catur Sleman Yogyakarta Indonesia. Telp. +628157988801. Email: [kusrini@amikom.ac.id](mailto:kusrini@amikom.ac.id)

<sup>2</sup> Gadjah Mada University, Mathematic and Natural Science Faculty, Yogyakarta Indonesia. Email: [shartati@ugm.ac.id](mailto:shartati@ugm.ac.id)

Student applicant's cancellation often occurs in STMIK AMIKOM Yogyakarta. A student candidate, who has been succeeded in the admission test, cancels his/her application by disregarding the next phase of admission process (re-registration). This condition causes a detrimental effect for STMIK AMIKOM, this makes the number of new students always go under the desired capacity. If the possibility of the registration cancellation can be detected early, then the executive manager can make any attempts to keep the candidate go through the admission process and subsequently minimize the rate of admission cancellation. A research to detect the possibility of application withdrawal is carried out recalling a previous experience suitable for solving the current problem, the case search and matching process is made easier, an indexing method is conducted before forming a decision tree. The decision tree is developed using C4.5 algorithm, which is improvement from the predecessor ID3 algorithm. This application was designed to be flexible. It allows modifications of variables or training cases. As the trial medium, it used more than 1500 data records of new student applicants for 2006/2007 teaching season in STMIK AMIKOM Yogyakarta

## 1. Introduction

In the year of 2006 in STMIK AMIKOM Yogyakarta, there are 1956 student candidates who had been succeeded in admission test, but 499 of them cancelled their application by disregarding re-registration. 25.5 % potential student candidate could not be endured by STMIK AMIKOM.

The cancellation should be minimized by the STMIK AMIKOM management, since incoming students will become their new source for operational and development finances.

If the possibility of a student candidate withdrawal can be detected early, it is expected the STMIK AMIKOM management can make some action to make their student candidate stay.

A technique to analyze the possibility is by doing classification to a set of candidate application data. Whether a candidate is going to withdraw his/her application or not, it can be identified by search his/her classification. One of famous classification modeling is by using decision tree.

Decision tree is categorized as a case indexing technique with inductive approach in case based reasoning. Case indexing refers to assigning indexes to cases for future retrieval and comparison. Inductive approaches are used to determine the case-based structure, which determines the relative importance of features for discriminating among similar cases, the resulting hierarchical structure of the case base provides a reduced search space for the case retriever. This may, in turn, reduce the query search time [6]. Other approaches in case indexing technique are Nearest-neighbor retrieval, Knowledge-guided approaches and Validated retrieval.

Plenty of algorithms are developed to build decision tree like ID3, CART and C4.5 [5].

Our research is building an application to detect the possibility of application withdrawal is carried out recalling a previous experience suitable for solving the current problem, the case search and matching process is made easier, an indexing method is conducted before building a decision tree. The decision tree is developed using C4.5 algorithm, which is improvement from the predecessor ID3 algorithm. This application was designed to be flexible. It allows modifications of variables or training cases. As the trial medium, it used more than 1500 data records of new student applicants for 2006/2007 teaching season in STMIK AMIKOM Yogyakarta.

## 2. Theory Background

### 2.1 Case Based Reasoning

*Case-based reasoning* (CBR) is a problem solving technique based on previous experience knowledge [1].

The problem-solving life cycle in a CBR system consists essentially of the following four parts (see Fig. 1):

1. Retrieving similar previously experienced cases (e.g., problem-solution-outcome triples) which problem is judged to be similar
2. Reusing the cases by copying or integrating the solutions from the cases retrieved
3. Revising or adapting the solution(s) retrieved in an attempt to solve the new problem
4. Retaining the new solution once it has been confirmed or validated

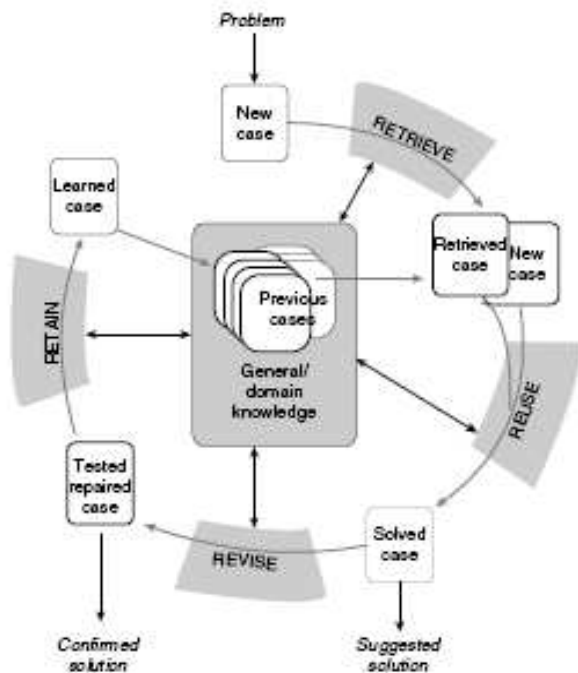


Fig. 1. Case Based Reasoning Life Cycle [6]

## 2.2 Decision Tree

A decision tree model consists of a set of rules for dividing a large heterogeneous population into smaller, more homogeneous groups with respect to a particular target variable[5]. A decision tree may be painstakingly constructed by hand in the manner of Linnaeus and the generations of taxonomists that followed him, or it may be grown automatically by applying any one of several decision tree algorithms to a model set comprised of pre-classified data.

The target variable is usually categorical and the decision tree model is used either to calculate the probability that a given record belongs to each of the categories, or to classify the record by assigning it to the most likely class. Decision trees can also be used to estimate the value of a continuous variable, although there are other techniques more suitable to that task[5].

Since the decision tree combines between data exploration and modeling, it is very good for beginning step in modeling process even when it positioned as final model from some other techniques.

Badriyah, T.(2006) made a classification utility with decision tree for decision support system. The algorithm used was ID3. The utility built in the Badriyah’s research has been succeeded to build a decision tree and if-then rule to solved problem in decision support system[2].

The C4.5 *algorithm* is Quinlan’s extension of his own ID3 algorithm for generating decision trees [3]. Just as with CART, the C4.5 algorithm recursively visits each decision node, selecting the optimal split, until no further splits are possible. However, there are interesting differences between CART and C4.5[5]:

- Unlike CART, the C4.5 algorithm is not restricted to binary splits. Whereas CART always produces a

binary tree, C4.5 produces a tree of more variable shape.

- For categorical attributes, C4.5 by default produces a separate branch for each value of the categorical attribute. This may result in more “bushiness” than desired, since some values may have low frequency or may naturally be associated with other values.
- The C4.5 method for measuring node homogeneity is quite different from the CART method and is examined in detail below.

## 2.3 C4.5 Algorithm

In general, steps in C4.5 algorithm to build decision tree are[4]:

- Choose attribute for root node
- Create branch for each value of that attribute
- Split cases according to branches
- Repeat process for each branch until all cases in the branch have the same class

Choosing which attribute to be a root is based on highest gain of each attribute. To count the gain, we use formula 1, below [4]:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (1)$$

with:

{S1, ..., Si, ..., Sn} = partitions of S according to values of attribute A

n = number of attributes A

|Si| = number of cases in the partition Si

|S| = total number of cases in S

while entropy is gotten by formula 2 below[4]:

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i \quad \dots\dots\dots(2)$$

with:

S : Case Set

n : number of cases in the partition S

pi : Proportion of Si to S

## 3. Design

To implement C4.5 algorithm in this decision tree creating, we use some tables in relational databases. They are:

- Data: {Student\_id, Name, Religion, school\_grade, ...}. It is used for store candidate student data.
- Atribut\_List: {Atribut\_Name, Is\_Result, Is\_Active}. It is used to store list of atribut that use to make decision tree. Is\_Result is told about the atribut is a result variable or not, while is\_active is told about the atribut is used or not.
- Data\_Value: {Atribut\_Name, Atribut\_Value, Min\_Value, Max\_Value}. It is used to store value definition of each attribute. For example, student school grade will classify into some value: A for

grade between 8 and 10, B for grade between 7 to 8 and C for grade under 7.

- Cases: {Case\_Id, Atribut\_Name[0], Atribut\_Name[1], ...Atribut\_Name[n]}. It is used to store cases data. Cases are taken from student\_data appropriate with selected atribut\_name in Atribut\_List table.

Besides tables that we have explained before, we dynamically create 2 kinds tables. They are Work: {Atribut\_Name, Gain} and Detail\_Work: {Atribut\_Name, Atribut\_Value, Case\_Count, Case\_Count\_Result[0], Case\_Count\_Result[1], ..., Case\_Count\_Result[n], Entropy}. Table Work is used to store data attribute that will be chosen as a selected node, whereas table Detail\_work is used to store atribut\_values of each atribut\_name so that we can generate gain value that store in table work. Table work and detail\_work are created dynamically from root until leaf of decision tree.

The general steps of using our application are shown if fig.2 below:

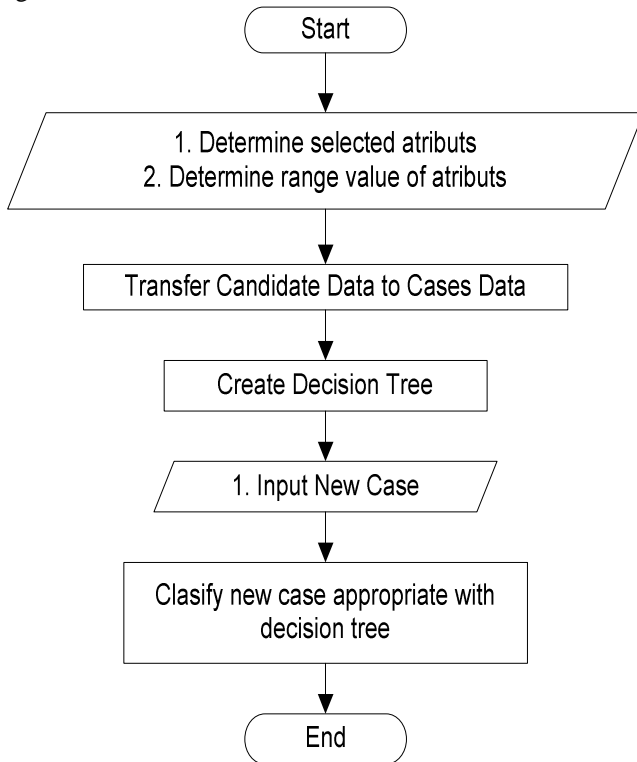


Fig.2. General steps of application

#### 4. Result

The interface of our application created can be shown in fig.3. below:

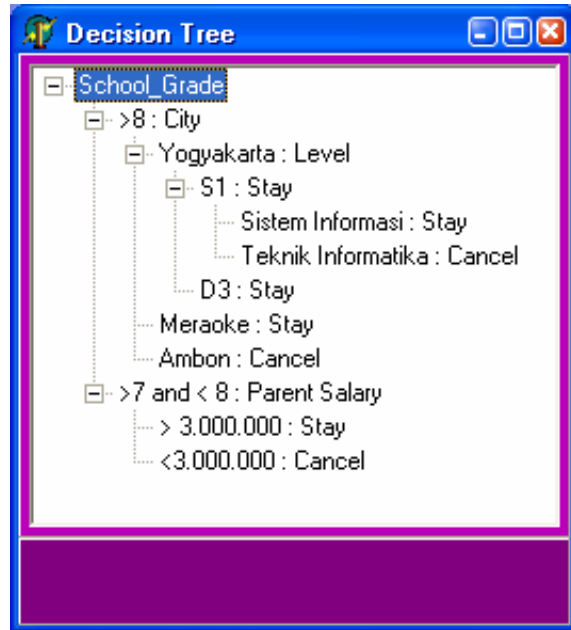


Fig. 4. Decision Tree interface

Possibility decision of a candidate is going to withdraw his/her application can be seen by matching the candidate data with the decision tree route from root to leaf. The leaf obtained describes the possibility of the candidate is going to leave or stay.

The decision tree produced conforms to the case's data input. In this application, the user allowed to add, replace or delete case. In addition, the variable used to build the decision tree can also be modified or managed from the application.

From 1956 data records of new student applicants in 2006's admission time of STMIK AMIKOM Yogyakarta, we used 1500 records for training the application. The remained data was left as the new data input and we used the result to test the application's accuracy.

#### 5. Conclusion

The application we have built can produce decision tree that conforms to variables and case's data given by user.

Accuracy level of the prediction data of this application is very depended to chosen variable that will be the basis to make the decision tree.

For the next improvement research, we can explore for variable(s) that can produce highest data accuracy level.

#### References

- (1) Armengol, E., Onta, S., dan Plaza, E., *Explaining similarity in CBR* Eva Armengol, Artificial Intelligence Research Institute (IIIA-CSIC). Campus UAB, 08193 Bellaterra, Catalonia
- (2) Badriyah, T., Rahmawati, R., *Alat Bantu Klasifikasi dengan Pohon Keputusan untuk Sistem Pendukung Keputusan*, Proceedings: Seminar Nasional Aplikasi Teknologi Informasi 2006, Jurusan Teknik Informatika, Universitas Islam Indonesia Yogyakarta (2006)
- (3) Berry, Michael J.A., Linoff, Gordon S., *Data Mining Techniques For Marketing, Sales, and Customer Relationship Management* Second Edition, Wiley Publishing, Inc., Indianapolis, Indiana (2004)
- (4) Craw, S., *Case Based Reasoning : Lecture 3: CBR Case-Base Indexing*, [www.comp.rgu.ac.uk/staff/smc/teaching/cm3016/Lecture-3-cbr-indexing.ppt](http://www.comp.rgu.ac.uk/staff/smc/teaching/cm3016/Lecture-3-cbr-indexing.ppt) (---)

- (5) Larose, Daniel T., Discovering Knowledge in Data: an Introduction to Data Mining, John Wiley and Sons, USA (2005)
- (6) Pall, Sankar K., Shiu, Simon C.K., Foundation of Soft Case Based Reasoning, John Wiley and Sons, USA (2004)