Assessment Of Component Structural Health By Rule Based Classification

Paulin Evans Pagalla
Mtech-IIyear,
Department of Computer Science and Engineering, Vignan University,
Guntur, India

R. Prathap Kumar
M.Tech CSE: Assistant Professor
Department of Computer Science and Engineering, Vignan University
Guntur, India

Abstract

Structural health is crucial for overall performance of the entire system. Technical failure of any component can effect its success. Such problems can be avoided well before by assessing the structural health and thus preventing unforeseen catastrophes. Rule Based Classification uses IF THEN rules for the assessing and classifying the data to different class labels. For example; using a flight database, a technician may want to assess the condition of a component in an aircraft with respect to few spatial dimensions (like route temperature, landing conditions, load etc). Depending on the structural health (of component) he might want the aircrafts to be scheduled for either the next flight or for maintenance. In this paper, we brief about the rule based classification, the spatial dimensions and propose appropriate algorithm for classifying the flight data into different class label. Synthetic data sets are used for implementing the system.

Keywords: *Rule based classification, spatial databases*

1. Introduction

Spatial database is a large collection of data like maps, satellite imagery, scientific and engineering application data etc. A System comprises of many components; components wear during operations, faulty components can decrease the system's life or even completely fail the system. Technical failures can be avoided before they actually occur by analyzing their health. An airliner database contains flight data and flight maintenance information. Flight data can be categorized as spatial data. It is a collection of both non spatial (ex. altitude, velocity, passenger load, temperature of the fuel lines etc.) and spatial data (commands from ground systems, images, cockpit voice recordings, route information etc.). Flight Maintenance information is a record of all the maintenance operations that are done on the aircraft during routine servicing. The data are analyzed by an analysis team for technical and structural health of the aircraft. Periodical Maintenance of the aircrafts is scheduled as per the manufacturer's advice, for instance, for every 1000 hours of flight. Components on an older aircraft experience frequent technical malfunctions than newer aircrafts hence the older components needs to be serviced more often than the manufacturer's advice. In this paper, we propose algorithm for assessing the health of the components and assign class labels to the data sets using rule based

classification of spatial data (flight data). The rest of the paper is structured as follows: Section 2 provides explanation on rule based classification and the Sequential Covering Algorithm. Section 3 describes the application and the results finally; Section 4 concludes the paper with future research directions.

2. Rule Based Classification of Spatial Data

Given a set *D* of data, rule based classification will assess the health of the component and assign the data sets to class labels. The health of a component is defined by structural and technical strength evaluated after the assessment. As a motivating example, a technician might want to assess the health of the engine of the aircraft with respect to the temperature exposure on fuel lines during the flight or for an instance he might want to assess the health of the landing system with respect to the vibration during its landing (whether a smooth landing or rough landing), and load on the aircraft. The basic structure of a rule is as follows:

IF *condition* **THEN** *conclusion.*

The c*ondition* is called the precondition of the rule which is also called as test condition, and the *conclusion* is the consequence if the condition is satisfied. The precondition usually consists of one or more test conditions that should be met. The rule consequence contains the class prediction or the class label. For ex., consider the rule R1given below, *temp* is the parameter and *threshold* is the limit where the parameter is not allowed to exceed i.e., if the temp parameter crosses the given threshold then an alarm set off. If the condition holds, we say that the rule covers the given data set.

R1: **IF** *temp>threshold* **THEN** *alarm.*

Usually a Rule Based Classifier extracts rules by using either by algorithms or by decision trees. As the size of the data sets increases the Decision tree extraction become complex over time and usually contain repetition of branches which have to be pruned later. Algorithms like AQ, CN2, and RIPPER are few examples. In our paper we have used a Sequential Covering Algorithms for rule extraction, which extracts rules from the data by one rule at a time.

Sequential Covering Algorithms:

The general strategy of the algorithm is to learn one rule at time for a class label. Every time a new rule is learned (created) the tuples that are covered by the rule are removed. The process continues until all tuples are covered or the termination condition is met. Basically we would like the rule to cover all or many tuples for the given class label and none from other classes since there could be many rules for a single class. The outline of the sequential covering algorithm is given below.

Algorithm: Sequential Covering

Input: D, a data set class labeled tuples

Att_vals:the set of all attributes and their possible values.

Output: A set of IF-THEN rules

Procedure:

1. Rule-*Set*= {}; //initial set of rules learned is empty

2. **For each** class *c* **do**

3. **repeat**

4. *Rule*=**learn-one-rule (***D,Att_vals,c***)**

5. remove tuples covered by *Rule* from *D*;

6. **until** terminating condition;

7. Rule_Set=Rule_Set+Rule;

8**. endfor**

9**. return** *Rule_Set*;

Learn-one-rule procedure finds the best rule for the current class for the given data set. It basically

adopts Greedy depth -first strategy. Initially the rule is created with an empty test condition for a current class, then each time the procedure encounters new data tuple (from the training data) it picks only those test conditions that improves the rule quality. Rule Quality is a measure of how accurate the rule classifies the given tuple to the current class. A rule is assessed by its coverage and accuracy

Rule Coverage: Coverage is the number of data tuples a rule covers It can be defined as below:

$$coverage(R) = \frac{n_{covers}}{|D|}$$

where, R is the current rule, n_{covers} is the number of data tuples that are covered by the rule and $|D|$ is the total number of tuples in the database.

Rule Accuracy: Accuracy is a measure of the correctness of the predicted class label. It can be defined as below:

$$accuracy(R) = \frac{n_{correct}}{n_{covers}}$$

where, R is the current rule, $n_{correct}$ is the number of tuples that are correctly classified to their class label and n_{covers} is the number of tuples that are covered by the rule R.

The idea is to aiming for high accuracy rather than high coverage.We initially start with an empty test condition for the class label, then we logically conjunct the test conditions to it. A general to specific rule search is done to establish the rules.

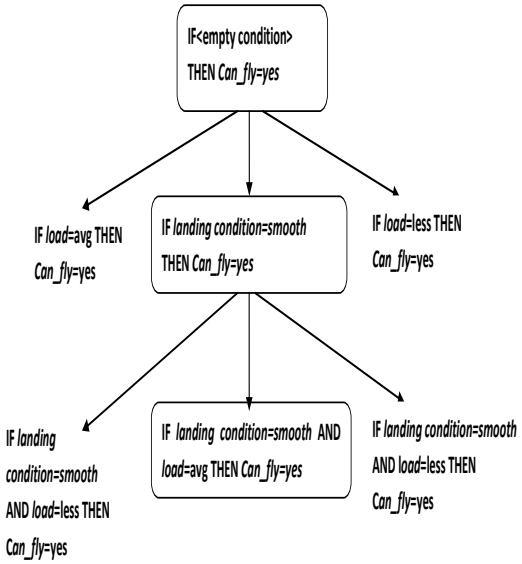

Figure 1. A general to specific rule search

In our example, the test condition *landing condition= smooth* best improves the quality of the current (empty) rule R2. We add it as our test condition and the rule which becomes:

R2: IF *landing condition=smooth* **THEN** *can_fly=yes*

In the next iteration another test condition *load*=avg best improves the quality of the rule. After logically appending the test condition, the rule becomes:

R2: IF *landing condition*=smooth **AND** *load*=med **THEN** *can_fly=yes*

The rule R2 classify the data to the class label "can_fly=yes" only when the test conditions *load* and *landing condition* are satisfied.

3. Application

The application has been developed on a Java platform with MySQL as Database Management System. The intended users of the system are administrator, project manager and technician. The Functional Requirements and respective users of the

application are identified as shown in the table 1. User Management deals with creation of user credentials and registration process. Component Management deals with addition of new component details and updating the status of obsolete components. Parameter Mangement deals with adition, removal parameters into the application database. All the functional requirements are implemented. The Menu is formatted as shown in the Figure 2 based on the requirements outlined in Table 1.

Table 1 Functional Requirements of the application

Users	Functional Responsibilities
System Administrator	• User Management • Component Management • Parameter Management
Project Manager	• Allocation of Component s' Parameters • Setting the bounds for the Parameters
Analyst	• Assessment of the Component Health • Preparing the Structural Health reports and MIS reports

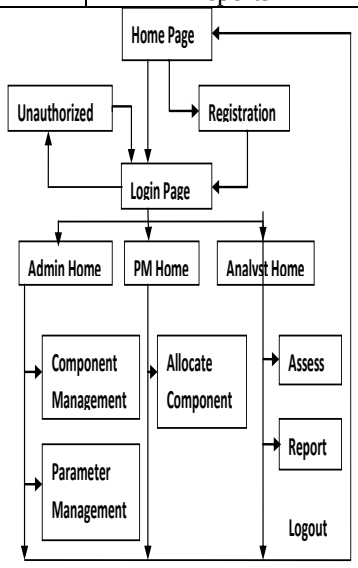

Figure 2. Menu Format

The GUI of the project manger and the technician are given in the figure3 and figure 4 respectively.

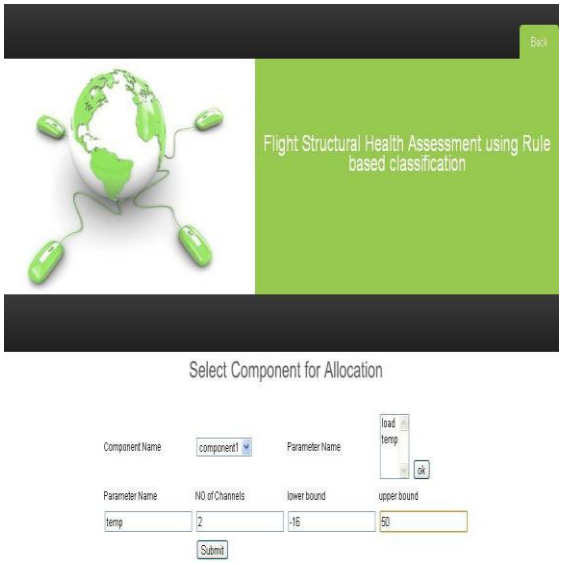

Figure 3 GUI for allocation of component

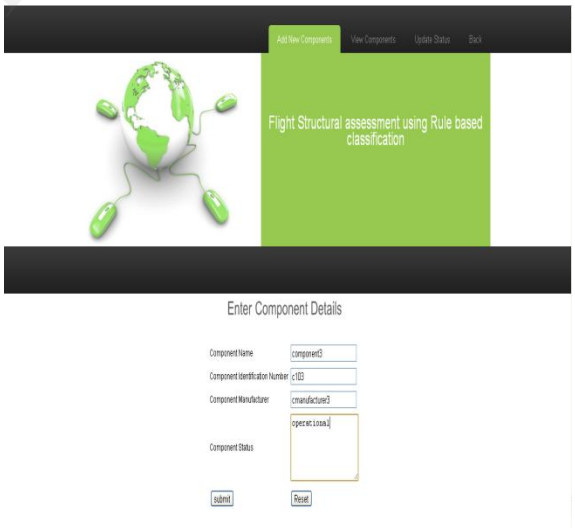

Figure 4 GUI for Component Management

Data acquisition will be done by sensors placed on the aircraft's components. The number and type of sensors to be deployed depends on the component being assessed. For ex, for assessing the health of a rudder, the parameters of interest would be wind velocity, stress on the wing, etc. Load on the

aircraft may not be a crucial parameter for such scenario. Data thus recorded are used for assessment post flight. For convenience, we have assumed data are clean and free from inconsistencies. Upper and Lower bounds for the parameters are obtained as per the standards given by the components' manufacturers. The data are randomly picked up from the database. Each of the picked data is compared with the bounds and any values that are outside of the range are reported. All the Data that falls between the lower and upper bounds are considered safe where as those that fall outside of the range are considered as unsafe. The farther the deviation from the safe zone the more abnormal the data are. The notify zone, alarm zone and auto zone are the unsafe zones.Data are assessed by plotting and counting their occurances. Each occurances in the unsafe zones are individually counted and based on their count the component is given a health status. The landing component of the aircraft has been assessed by considering three parameters, namely vibration experienced during the landing, climatic temperature and the load on the aircraft in the application. Half of the data from the database is used as training data sets and the rest as the test data. Rule Quality is determined by the rule accuracy and Rule Coverage as explained in the section 2.

Consider the sample training data sets from the flight database. The first column in the table uniquely identifies each row in the table. The second, third and fourth columns stores the summarized data for the parameters temp, vibration and load on a landing component. The last column Class:can_fly stores the prediction for the class label "can_fly". A total of 18 data sets are represented in the table 2. Consider the rule R2 from the section 2. It covers two tuples (RID 11 and 17) of the18 tuples. It can correctly classify both tuples. Therefore, Rule coverage and accuracy for the rule R2 are calculated.

coverage(R2) = 2/18=11.11%

where $n_{covers\ =\ 2}$ and $|D|$=18

accuracy(R2) = 2/2=100%

where $n_{correct}$ =2 and n_{covers} =2

Table 2 Sample training data sets.

RID	Route Temperature	Landing Condition	Load	Class: can_fly
1	high	rough	heavy	no
2	high	rough	average	no
3	high	rough	light	no
4	high	smooth	heavy	no
5	high	smooth	average	no
6	high	smooth	light	yes
7	normal	rough	heavy	no
8	normal	rough	average	no
9	normal	rough	light	yes
10	normal	smooth	heavy	yes
11	normal	smooth	average	yes
12	normal	smooth	light	yes
13	low	rough	heavy	no
14	low	rough	average	no
15	low	rough	light	no
16	low	smooth	heavy	no
17	low	smooth	average	yes
18	low	smooth	light	yes

There is no known solution (application) that assesses the post flight structural health of a component using Rule based Classification. The application developed has achieved all functional requirements listed in Table 1. Rule Quality is determined by Rule Accuracy rather than Rule Coverage the application has achieved 100% Rule Accuracy as explained above.

4. Conclusion

In this paper, we have studied Rule based classification of Spatial Data. Rule based Classification uses decision trees and sequential covering algorithms for rule extraction. We have used sequential covering algorithms to extract rules from the training data sets and classified the test data sets. Rule Extraction using decision trees becomes complex when the database is large. Branch repetition and pruning has to be performed. We can reduce the time and space complexities by using sequential covering algorithm. The application was developed by considering the landing system in the aircraft. The various parameters that are considered to assess the health of the landing system are three, namely vibration experienced by the landing gear

while landing, load on the aircraft, and the temperature exposure. In future we will extend this study by considering multiple dimensions, keeping in view the overall performance of the application.

REFERENCES

[1] Data Mining Concepts and Techniques, Second Edition, Jiawei Han and Micheline Kamber.
[2].Software Requirements, Karl E Weigers
[3] UML for Java programmers, Martin, Low Price Edition
[4] RAPID JAVA application development using JBUILDER#, Y.Daniel Liang, Prentice Hall
[5] JSP2.0 The Complete Reference, Hanna, TMH Edition
[6] Software Requirements & Specifications a lexicon of practice, principles & prejudices, Micheal Jackson, Addison-Wesley
[7] The Web Programming Desktop Reference 6 in 1, Micheal Afergan, Rick Darnell, Brian Farear, Russ Jacobs, David Medinets, Robert Muller, Micheal O Foghtu, PHI