Computers & Industrial Engineering 56 (2009) 1319–1327

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A confirmation technique for predictive maintenance using the Rough Set Theory

Micaela Caserza Magro, Paolo Pinceti *

University of Genoa, Department of Electrical Engineering, via all'Opera Pia, 11a, 16145 Genova, Italy

ARTICLE INFO

Article history:
Received 18 December 2006
Received in revised form 28 July 2008
Accepted 30 July 2008
Available online 9 August 2008

Keywords:
Predictive maintenance
Rough Set Theory
Intelligent field device
Data mining
Knowledge discovery in database

ABSTRACT

This paper presents a technique to improve the accuracy of the predictions obtained using the Rough Set Theory (RST) in non-deterministic cases (rough cases). The RST is here applied to the data collected by the Intelligent Field Devices for identifying predictive diagnostic algorithms for machinery, plants, subsystems, or components. The data analysis starts from a historical data set recorded from the field instruments, and its final result is a set of "if–then" rules identifying predictive maintenance functions. These functions may be used to predict if a component is going to fail or not in the next future. The prediction is obtained by applying the rules extracted with the RST algorithm on the real-time values transmitted by the field device. It may happen that some diagnoses are uncertain, in the sense that it is not possible to take a certain decision (device sound or close to fail) with a given set of data. In this paper, a new algorithm for increasing the confidence in these uncertain cases is presented. To show an example, the proposed confirmation algorithm is applied to the predictive algorithms obtained for an intelligent pressure transmitter.

1. Introduction

The Rough Set Theory (RST) has great potentialities in the area of predictive maintenance. RST is one of the mathematical techniques for extracting automatically information hidden in a database (KDD = knowledge discovery in database). KDD is commonly defined (Fayyad, Piatetsky-Shapiro, & Smyth, 1996; Piatetsky-Shapiro & Frawley, 1991) as "the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data".

Thus the KDD process starts from the atomic pieces of information (data) and processes them to obtain information at a higher-level (knowledge) and to present them in an organize form (pattern).

KDD is based on multiple successive actions, such as: selection, pre-processing, transformation, data-mining, and interpretation. The very core of KDD is the *data-mining* (Hand, Mannila, & Smyth, 2001) that represents a mathematical tool to find unsuspected relationships between data, and to summarize the data in novel ways that are both understandable and useful to the data owner. Several techniques for data-mining exist, and the paper focuses on the RST that seems to be promising for defining predictive maintenance algorithms.

The RST technique is quite robust and results obtained in the medical field, where the theory was first applied, are quite good. The idea behind RST is to cross-correlate non-homogenous data to find-out the data that are most significant for predicting the system health. It seems to be possible to use the RST for obtaining diagnostic algorithms for industrial plants or subsystems, machinery, and Intelligent Field Devices (IFD) as well. IFD are devices (transmitters, actuators, inverters, and so on) with a digital core, and the capacity of transmitting large quantities of additional data together with their basic information. These data can be used to evaluate the system operational status and to predict future failures or collapses. Applying the RST to a historical database containing records of symptoms (internal parameters) and operational status (sound, abnormal, failed), it is possible to create a predictive tool to be used with the real-time data flow. Rules extracted from the RST are in the form "If–Then", therefore very simple to implement in a control system even with limited computational power. The obtained rules can be written in a script file, that has a very small size and that can be executed using very small CPU and memory resources. This means that even the CPU of an IFD can run such scripts without installing specific software or mathematical tools. Thus the diagnostic algorithms can be directly executed on board without using the control system resources. The "If–then" rules correlate directly the data existing in the IFD, without the need of adding external parameters or variables (i.e., the weights in a neural network).

An important feature of RST is its capacity of shrinking the size of the database, since it considers only the data really useful for finding out the diagnosis. In fact, some records of the database may be redundant, and some of the features may not be useful for determining the value of the decision attribute. RST is very effective in eliminating these redundant data.

* Corresponding author. Tel.: +39 010 353 2205.
E-mail address: pinceti@die.unige.it (P. Pinceti).

0360-8352/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2008.07.024

This paper presents in the first part the fundaments of the RST technique, with a particular focus on the management of uncertain cases, called *rough concepts*. In the framework of RST, *uncertainty* means that two or more database items having the same symptoms present different diagnosis.

The second part of the paper defines a new algorithm based on the cluster analysis for increasing the confidence about the diagnosis of uncertain cases. The aim of this algorithm is to compute the probability associated with each possible diagnosis for every rough concept. We call it a *confirmation algorithm*.

At the end of the paper a practical case is studied. The RST and the proposed confirmation algorithm are applied to an intelligent pressure transmitter. A model for the predictive diagnostic is presented and discussed.

2. Rough Set Theory fundaments

Rough Set Theory deals with the information hidden in the data collected in a database (Komorowski, Pawlak, Polkowski, & Skowron, 1999; Pawlak, 1998; Polkowski & Skowron, 1998; Walczak & Massart, 1999). Each item x_i of the *Universe* is known by means of a given set of features (called *F*). It may happen that two or more items are *indiscernible* because the values of all their features (a_i) are identical.

In formal terms, RST operates on a *decision system*, which is defined formally as:

$$\Gamma = (U, F \cup \{d\}) \tag{1}$$

where:

- *U* is a non-empty finite set of items, referred as the Universe;
- *F* is a non-empty finite set of features;
- *d* is a decision attribute which represents the classifying feature ($d \notin F$).

Extracting the knowledge from a decision system means to find-out the rules that can be used to determine the decision attribute of a new item with a given feature set. It may happen that not all the items contained in a decision system are useful in determining the rules. Either some items are indiscernible, or some features do not influence the decision attribute. The RST is quite effective in simplifying the database under both these aspects.

To identify redundant items the *indiscernibility relation* is defined, that is an equivalence relation that defines sets of items having the same features.

Formally, let $\Gamma = (U, F)$ be an information system, then an equivalence relation $\mathrm{IND}_\Gamma(B)$ is associated to any subset $B \subseteq F$:

$$\mathrm{IND}_\Gamma(B) = \{(x, x') \in U^2 | \forall f \in B, f(x) = f(x')\} \tag{2}$$

$\mathrm{IND}_\Gamma(B)$ is called the indiscernibility relation. If $(x, x') \in \mathrm{IND}_\Gamma(B)$, then x and x' are indiscernible by means of the features in the subset *B*. The indiscernibility relation divides the Universe into partitions (or equivalence classes), where each partition contains indiscernible items.

The RST operates with discrete variables. If a feature is collected as a continuous variable, then it is necessary to split it into a discrete number of fields. Several methods exist to define the optimal splitting of a variable (i.e., equal width and equal frequency interval binning, Holte 1R discretizer, etc) (Beynon, 2004; Beynon & Peel, 2001; Nguyen & Nguyen, 1998). Experience and the knowledge of the physical process generating the items can be effectively used for increasing the efficiency of the discretization process. In fact, an expert can define ranges of the features with a physical sense related to the process (low, normal, high, etc.), and this can help the understanding and use of the extracted rules.

For a better understanding of the above concepts a simple example is given, where the decision system is based on eight items and three features. The items are cars and the three features are: "*Consumption*", "*Age*", and "*Previous accidents*". In addition, there is a decision feature about the reliability of the car (1 means that the car with that set of features had a failure within one week, 0 means that the car was sound after one week).

Starting from the four features of the decision system in Table 1, we can define different partitions of the universe, like for example:

- $\mathrm{IND}(\{Consumption\}) = (\{x_1, x_3, x_7, x_9\}, \{x_2, x_5, x_6\}, \{x_4, x_8\})$
- $\mathrm{IND}(\{Consumption, \ Age\}) = (\{x_1, x_7\}, \{x_3, x_9\}, \{x_2, x_6\}, \{x_5\}, \{x_4, x_8\})$
- $\mathrm{IND}(\{Consumption, \ Age, \ Prev. \ Accident\}) = (\{x_1, x_7\}, \{x_3\}, \{x_2, x_6\}, \{x_5\}, \{x_4, x_8\}, \{x_9\})$

The partition of the universe based on the decision feature, that is the class of the items with the same decision value, is referred as a *concept*. In our example there is a concept associated with "*Decision = 1*" and one associated with "*Decision = 0*". Concepts can be *crisp* or *rough*. Concept "A" is rough if at least one item of the universe with the same values of all the features of an item belonging to the concept "A" belongs to another concept. A concept is defined crisp in all the other cases. In the above example, both the concepts are rough. In fact, item x_2 and x_6 have the same conditional features, but different decisions. To deal with this kind of uncertainty RST first identifies the items surely belonging to the class Decision = 0 and the items that surely belong to the class Decision = 1. The set of indiscernible items belonging only to concept "A" is called *lower approximation of concept "A"*. The upper approximation of concept "A" is the union of its lower approximation and all the items that cannot be surely assigned to concept "A". A boundary region exists between these two sets that contain the items that can not be classified with certainty that is with equal features and different decision.

In a formal way the lower approximation ($\underline{B}X$) and the upper approximation ($\overline{B}X$) can be expressed as in (3) and (4), respectively. Let $B \subseteq F$ and $X \subseteq U$:

$$\underline{B}X = \{x \,|\, [x]_B \subseteq X\} \tag{3}$$

$$\overline{B}X = \{x \,|\, [x]_B \cap X \neq \emptyset\} \tag{4}$$

A graphical representation of the upper and lower approximation for the example in Table 1 is shown in Fig. 1.

The lower approximation of concept "Decision = 1" contains the items that can surely be classified as belonging to this concept (x_1, x_3, x_7, x_9), while its upper approximation contains also the items belonging both to D = 1 and to the concept "Decision = 0" (x_2, x_6).

RST defines the *rough membership function* (μ_X^B) to quantify the degree of relative overlapping between the set *X* and the equivalence class $[x]$ to which the item x belongs:

Table 1
The considered decision system

Item	Consumption	Age	Prev. Accidents	Decision
X_1	5–11	6–10	0	1
X_2	12–20	6–10	0	0
X_3	5–11	0–5	1	1
X_4	21–33	0–5	0	0
X_5	12–20	0–5	1	0
X_6	12–20	6–10	0	1
X_7	5–11	6–10	0	1
X_8	21–33	0–5	0	0
X_9	5–11	0–5	0	1