The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT-2011)

Solving Incomplete Datasets in Soft Set Using Supported Sets and Aggregate Values

Ahmad Nazari Mohd Rose\textsuperscript{a}, Hasni Hassan\textsuperscript{a}, Mohd Isa Awang\textsuperscript{a}, Nor Aida Mahiddin\textsuperscript{a}, Hidayatulaminah Mohd Amin\textsuperscript{a}, Mustafa Mat Deris\textsuperscript{b}

\textsuperscript{a}FIT, Universiti Sultan Zainal Abidin, Terengganu, Malaysia, \\
\textsuperscript{b}FTMM, Universiti Hussein Onn, Johor, Malaysia

Abstract

The theory of soft set proposed by Molodtsovin 1999[1] is a new method for handling uncertain data and can be defined as a Boolean-valued information system. It has been applied to data analysis and decision support systems based on large datasets. In this paper, it is shown that calculated support value can be used to determine missing attribute value of an object. However, in cases when more than one value is missing, the aggregate values and calculated support values will be used in determining the missing values. By successfully recovering missing attribute values, the integrity of a dataset can still be maintained.

Keywords : Boolean-valued information systems; Soft Set Theory; supported set; incomplete datasets, missing values.

1. Introduction

Sometimes information retrieved from a dataset is incomplete due to problems such as data mishandling, virus attack or hardware degradation. Data are vital especially in the aspect of decision making. However, incomplete datasets will probably produce different results when used in the process of decision making. Therefore, we believe that it is imperative to do some pre-processing to a dataset so that extra attribute values could be stored and used in the process of recovering missing attributes in the dataset.

In 1999, Molodtsov proposed a new method for handling uncertain data, called the soft set theory. Soft sets are also called binary, basic, and elementary neighbourhood systems [1]. The soft set is a mapping from parameter to the crisp subset of universe. From such case, we may see that a soft set can be used to classify objects into two
classes (yes/1 or no/0). This means that the “standard” soft set deals with a Boolean-valued information system. The main strength of the soft set theory is its ability to liberate itself from any shortfall of the parameterization tools, unlike in the theories of fuzzy set, probability and interval mathematics [1]. The theory of soft set has been applied to data analysis and decision support systems based on large datasets.

Application on the concept of supported sets of objects in reducing Boolean-valued information based on soft set theory has been shown by Rose et. al [2]. Relying on the same technique of supported set as in [2], we then would like to propose algorithms for solving missing attribute values in datasets. The main purpose of the proposed technique is to ensure that any attribute values that are missing after the data has been processed can still be recovered. By successfully recovering missing attribute values, the integrity of the dataset can still be maintained hence maintaining consistencies in decision making.

At the moment, discussions pertaining to data analysis approach in incomplete set largely lies in the area of rough set [3], [4] and [5]. In the domain of soft set, Zhao et. al [6] highlighted that tolerance relation is also used to create tolerance matrices and discernibility matrices for dealing with incomplete data in rough sets; in addition to deletion and data filling. Using these methods, the equivalent relations are maintained. For standard soft sets, Zhao and Xiao proposed that the decision value of an object with incomplete information is calculated by weighted-average of all possible choice values of the object [7]. In addition, they introduced the idea of weighted-average of all the possible choices where the weight of each possible choice value is decided by the distribution of other objects. They had also used prediction based on the method of average-probability for solving incomplete data in fuzzy soft sets.

In this paper, we have proposed a method to recover missing attributes values of a stored dataset. We have shown that missing attribute values in a dataset could be retrieved using either supported set values or a combination of supported set values and the attribute aggregate values. The paper has presented a contribution in terms of recovering missing attributes in a pre-processed and stored dataset, hence maintaining the integrity of the dataset.

The rest of this paper is organized as follows. Section 2 describes the fundamentals of an information system and soft set theory. Section 3 presents the concepts of supported sets. Section 4 discusses the problem of missing attributes that are solved using supported sets and the aggregate values of the attributes. Finally, we conclude works in section 5.

2. Soft Set

In this section, the concepts and definition of soft sets are defined.

2.1 Information system

An information system is a 4-tuple $S = (U, A, V, f)$, where $U = \{u_1, u_2, \ldots, u_U\}$ is a non-empty finite set of objects, $A = \{a_1, a_2, \ldots, a_A\}$ is a non-empty finite set of attributes, $V = \bigcup_{a \in A} V_a$, $V_a$ is the domain (value set) of attribute $a$, $f : U \times A \rightarrow V$ is an information function such that $f : U \times A \rightarrow V$, $f(u, a) \in V_a$, for every $(u, a) \in U \times A$, called information (knowledge) function.

2.2 Definitions

Throughout this section $U$ refers to an initial universe, $E$ is a set of parameters, $P(U)$ is the power set of $U$ and $A \subseteq E$.

**Definition 1.** (See[1].) A pair $(F, E)$ is called a soft set over $U$, where $F$ is a mapping given by $F : E \rightarrow P(U)$

$$F : A \rightarrow P(U).$$
In other words, a soft set over $U$ is a parameterized family of subsets of the universe $U$. For $e \in A$, $F(e)$ may be considered as the set of $e$-elements of the soft set $(F,A)$ or as the set of $e$-approximate elements of the soft set. Clearly, a soft set is not a (crisp) set. To illustrate this idea, let us consider the following example.

**Example 1.** (See[8].) Let’s consider a soft set $(F,E)$ which describes the “attractiveness of houses” that Mr. X is considering to purchase. Suppose that there are six houses in the universe $U$ under consideration,

$$U = \{h_1, h_2, h_3, h_4, h_5, h_6\} \text{ and } E = \{e_1, e_2, e_3, e_4, e_5\}$$

is a set of decision parameters, where $e_1$ stands for the parameters “expensive”, $e_2$ stands for the parameters “beautiful”, $e_3$ stands for the parameters “wooden”, $e_4$ stands for the parameters “cheap” and $e_5$ stands for the parameters “in the green surrounding”.

Consider the mapping $F : E \to P(U)$, given by “houses ( )”, where ( ) is to be filled in by one of parameters $e \in E$.

Suppose that

$$F(e_1) = \{h_2, h_4\}, \quad F(e_2) = \{h_1, h_3\}, \quad F(e_3) = \{h_1, h_4, h_5\}, \quad F(e_4) = \{h_1, h_3\}, \quad F(e_5) = \{h_1\}$$

Therefore, $F(e_1)$ means “houses (expensive)”, whose functional value is the set $\{h_2, h_4\}$. While $F(e_2)$ means “houses (beautiful)”, whose functional value is the set $\{h_1, h_3\}$ and so forth.

Each approximation has two parts, a predicate $p$ and an approximate value set $v$. For example, for the approximation “expensive houses $F(e_1)$”, we have the predicate name of expensive houses and the approximate value set or value set of $\{h_2, h_4\}$. Thus, a soft set $(F,E)$ can be viewed as a collection of approximations, has been defined in the following:

$$F = \{p_1 = v_1, p_2 = v_2, \ldots, p_e = v_e\}.$$  \hspace{1cm} (1)

**Definition 2.** (See[2]) If $(F,E)$ is a soft set over the universe $U$, then $(F,E)$ is a binary-valued information system $S = (U,A, \{V_{e,i}\}, f)$.  

**Proof.** Let $(F,E)$ be a soft set over the universe $U$, the mapping is defined as follows:

$$F = \{f_1, f_2, \ldots, f_e\},$$

Where

$$f_i : U \to V_i \text{ and } f_i(x) = \begin{cases} 1, & x \in F(e_i) \\ 0, & x \notin F(e_i) \end{cases}, \quad \text{for } 1 \leq i \leq |A|. $$

Hence, if $A = E$, $V = \bigcup_{e \in A} V_e$, where $V_e = \{0,1\}$, then a soft set $(F,E)$ can be considered as a binary-valued information system $S = (U,A, \{V_{e,i}\}, f)$. From Definition 2, it is therefore permissible to represent a binary-valued information system as a soft set. Thus, we can construct a one-to-one correspondence between $(F,E)$ over $U$ and $S = (U,A, \{V_{e,i}\}, f)$.

Table 1. Tabular representation of a soft set in the above example.
As observed from Table 1, the usage of “1” and “0” is to denote whether the attribute can be used for the description of the house when the table is represented as a Boolean-valued information system. In the table, a “1” is defined by \( f(x) = \{1, x \in F \} \), denotes the presence of the described attribute for the house, and a “0” denotes that the attribute is not part of the description of the house.

**Definition 3.** (See [1].) \( S = (U, A, V_{\emptyset, \ast}, f) \). If \( (F, E) \) is a soft set over the universe \( U \), then \( (F, E) \) is an incomplete binary-valued information system.

Each attribute value which contains null or interpreted to have a missing value, can be filled with * as an indicator of a missing value.

Hence, if \( A = E, V = \bigcup_{e \in A} V_e \), where \( V_e = \{0, *, \ast\} \), then a soft set \( (F, E) \) can be considered as an incomplete binary-valued information system \( S = (U, A, V_{\emptyset, \ast}, f) \).

### 3. Interpreting missing information

This section presents techniques used to identify the missing bits. When the number of missing values is small, using the supported set values and attribute aggregate values would suffice. On the other hand, there are cases when supported set and attribute aggregate values are not able to make out the missing values. This is when the diagonal aggregate value must be used to interpret the missing values.

#### 3.1 Supported sets

We will be assuming that data that has been analysed are presented as in Table 1. Using the calculated values, the incomplete dataset is then processed to identify the missing values. Throughout this sub-section, the pair \( (F, E) \) refers to the soft set over the universe \( U \) representing a Boolean-valued information system \( S = (U, A, V_{\emptyset, \ast}, f) \).

**Definition 4.** Let \( (F, E) \) be a soft set over the universe \( U \) and \( u \in U \). Support of an object \( u \) is defined by \( \text{supp}(u) = \text{card}(\{e \in E : f(u, e) = 1\}) \) \hspace{1cm} (2)

#### 3.2 Attribute aggregate values

In this section, we will introduce a a concept called attribute aggregate values. An attribute aggregate value is actually the cumulative value for any particular attribute.

**Definition 5.** Let \( (F, E) \) be a soft set over the universe \( U \) and \( e \in A \). The aggregate value for an attribute \( e \) is defined as \( C_{\text{agg}} = \sum_{u \in U} f(u, e) \). \hspace{1cm} (3)
3.3 Diagonal aggregate values

As mentioned previously, an information system is \( S = (U, A, V, f) \), where \( A = \{a_1, a_2, \ldots, a_n\} \) is a non-empty finite set of attributes, \( U = \{u_1, u_2, \ldots, u_m\} \) is a non-empty finite set of objects, and \( V = \bigcup_{a \in A} V_a \). \( V_a \) is the domain (value set) of attribute \( a_i \). In computing the diagonal aggregate values for an information system \( S = (U, A, V, f) \) based on values of \( f(u_i, a_j) \), where \( i = 1, 2, 3, \ldots, |U| \) and \( j = 1, 2, 3, \ldots, |A| \), [12] has defined tuple, \( t \) as 

\[
\mathbf{t} = (f(u_1, a_1), f(u_1, a_2), f(u_1, a_3), \ldots, f(u_1, a_n)),
\]

where \( i = 1, 2, 3, \ldots, |U| \).

There are two implementations to calculate the diagonal aggregate values. One is diagonal Left-to-Right aggregate values defined as diagonal LR aggregate values that start its diagonal stretch from the sides of \( f(u_i, a_j) \). Another implementation is Right-to-Left aggregate values defined as RL aggregate values with its diagonal stretch from the sides of \( f(u_i, a_j) \).

**Definition 6.** Let \((F, E)\) be a soft set over the universe \( U \) and \( e \in A \) and tuple 

\[
\mathbf{t}_i = \left( f(u_1, a_i), f(u_2, a_i), f(u_3, a_i), \ldots, f(u_n, a_i) \right),
\]

where \( i = 1, 2, 3, \ldots, |U| \) and \( D = |U| + |A| - 1 \) as the number of diagonals

For \( 1 \leq k \leq |A| \)

\[
\text{Diag}_{LR}(k) = \sum_{j=i}^{k} f(u_j, a_i) j = k - i + 1;
\]

For \( |A| < k \leq D \)

\[
\text{Diag}_{RL}(k) = \sum_{i=k}^{n} f(u_i, a_j) j = |A| - k + i;
\]

where \( i = 1, 2, 3, \ldots, |U| \).

3.4 Algorithm for solving missing attributes in a dataset

The pseudo-code for searching and solving missing values is as follows:

1. Input the supported values \( \text{supp}(a) \), the aggregate of column values and the diagonal aggregate values.
2. Solve each single missing value either horizontally, vertically or diagonally.
3. Repeat Step 2 until there are no more single missing values.
4. Solve other missing values using supported values or/and column aggregate values or/and diagonal aggregate values.
5. Repeat Step 2.

Figure 1. The pseudo-code of the proposed technique
4 Discussion

4.1 Calculating supported set and attribute aggregate values.

Based on the Definition 4, we can calculate the supported set values of Table 1. We then proceed by calculating the attribute aggregate values. Table 2 presents both the attribute aggregate values and supported values based on Table 1.

Table 2. Integrating supported value into tabular representation of soft set in Table 1.

<table>
<thead>
<tr>
<th>U</th>
<th>e₁</th>
<th>e₂</th>
<th>e₃</th>
<th>e₄</th>
<th>e₅</th>
<th>supp(h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>h₁</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>h₂</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>h₃</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>h₄</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>h₅</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>h₆</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>∑ col</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

We then calculate the diagonal LR aggregate values as shown in Table 3. Note that the column representing the supported values and the row implying the attribute aggregate values are not shown.

Table 3. Samples of diagonal LR-aggregate values

<table>
<thead>
<tr>
<th>U</th>
<th>e₁</th>
<th>e₂</th>
<th>e₃</th>
<th>e₄</th>
<th>e₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>h₁</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>h₂</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>h₃</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>h₄</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>h₅</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>h₆</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

4.2 Missing Cases

In Table 4, incomplete attribute value will be denoted by symbol “*”.

Table 4. An example of missing attributes values.

<table>
<thead>
<tr>
<th>U</th>
<th>e₁</th>
<th>e₂</th>
<th>e₃</th>
<th>e₄</th>
<th>e₅</th>
<th>supp(h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>h₁</td>
<td>*</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>h₂</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>h₃</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>h₄</td>
<td>1</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>h₅</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>*</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>h₆</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>∑ col</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Using Table 4 as an example to determine the missing attribute values, we hereby present the following cases:

i) **Single missing attribute value:**

\(f(h_i, e_j)\) is an example of the case of single missing value from the both vertical and horizontal perspectives. Looking from the horizontal perspective, the missing attribute value is 0. The interpretation is based on the supported value for the object \(h_i\) which is already 3, and if \(f(h_i, e_j)\) is 1 it would cause the supported value of \(h_i\) to be 4. Next, \(f(h_i, e_j), f(h_i, e_j)\) and \(f(h_i, e_j)\) are examples of single missing values from the vertical perspective. Consequently, after solving \(f(h_i, e_j), f(h_i, e_j), f(h_i, e_j)\) and \(f(h_i, e_j)\); we can represent the results as in Table 5.

Table 5. Representation of Table 5 after solving \(f(h_i, e_j), f(h_i, e_j), f(h_i, e_j)\) and \(f(h_i, e_j)\)

<table>
<thead>
<tr>
<th>(U)</th>
<th>(e_1)</th>
<th>(e_2)</th>
<th>(e_3)</th>
<th>(e_4)</th>
<th>(e_5)</th>
<th>(\text{supp}(h))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(h_i)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>(h_i)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(h_i)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>(h_i)</td>
<td>1</td>
<td>0</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>(h_i)</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>(h_i)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(\sum_{\text{col}})</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Now, looking from the horizontal perspective; we have two single missing values for \(f(h_i, e_j)\) and \(f(h_i, e_j)\). The missing values are both 1 since the supported values of \(h_i\) and \(h_j\) are both 2.

ii) **Three missing attribute values:**

Table 6. An example of three missing attributes values.

<table>
<thead>
<tr>
<th>(U)</th>
<th>(e_1)</th>
<th>(e_2)</th>
<th>(e_3)</th>
<th>(e_4)</th>
<th>(e_5)</th>
<th>(\text{aggregate})</th>
<th>(\text{supp}(h))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(h_i)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>(h_i)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(h_i)</td>
<td>0</td>
<td>1</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>(h_i)</td>
<td>1</td>
<td>0</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>(h_i)</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>(h_i)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(\sum_{\text{col}})</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6 shows an example of three missing attribute values. We first solve \(f(h_i, e_j)\), by referring to diagonal RL aggregate as shown by the dotted oval. Since the sum of the diagonal is 2, therefore \(f(h_i, e_j)\) is definitely 1. As for \(f(h_i, e_j)\), it can be deduced that the missing value is 0 since its diagonal RL aggregate value is 0. Next, by referring to the supported value of object \(h_j\), it can be deduced that \(f(h_j, e_j)\) and \(f(h_j, e_j)\) are both 1’s. By solving the four missing values, we then obtain Table 7.
Now we have two single missing values that are \( f(h_1, e_1) \) and \( f(h_2, e_4) \) from the diagonal perspective. After solving \( f(h_1, e_1) \) and \( f(h_2, e_4) \), we are left with \( f(h_3, e_1), f(h_4, e_3), f(h_5, e_3) \) as single missing values, which can then solved using their corresponding attribute aggregate value.

### 5 Conclusion

In this paper, we have defined supported sets in accordance to Boolean-valued information systems. While in [2] supported sets have proven feasible in reducing a dataset; we have used the supported set to recover missing attribute values in an incomplete information system. In addition to supported values set, we have also proven that attribute aggregate values and diagonal aggregate values can also be used as guidance in solving the missing attribute values in a dataset. The importance of this solution is that it helps to maintain the integrity of the dataset by successfully recovering the missing values. We intend to elaborate more on other solutions for solving incomplete information systems in a different kind of situation.

### Acknowledgement

This work was supported by the FRGS of Ministry of Higher Education, Malaysia. Ref. No: FRGS/2/2010/SG/UDM/02/2,

### References

3. Thiesson, B., Accelerated Quantification of Bayesian Networks with Incomplete Data. The First International Conference on Knowledge Discovery and Data Mining, Montreal, Canada, 1995: p. 306-311.