Predicting Missing Attribute Values based on Frequent Itemset and RSFit

Jiye Li
School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
j27li@uwaterloo.ca

Nick Cercone
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, Canada B3H 1W5
nick@cs.dal.ca

ABSTRACT

How to process missing attribute values is an important data preprocessing problem in data mining and knowledge discovery tasks. A commonly-used and naive solution to process data with missing attribute values is to ignore the instances which contain missing attribute values. This method may neglect important information within the data and a significant amount of data could be easily discarded. Some methods, such as assigning the most common values or assigning an average value to the missing attribute, make good use of all the available data. However the assigned value may not come from the information which the data originally derived from, thus noise is brought to the data. We introduce an integrated approach **ItemRSFit** to effectively predict missing attribute values by combining frequent itemset and RSFit approaches together. Frequent itemset is generated from the association rules algorithm and it displays the correlations between different items in a transaction data set. Using frequent itemset as a knowledge base to predict missing attribute values is shown to have a high prediction accuracy. However this approach alone cannot predict all the existing missing attributes. RSFit is a newly developed approach to predict missing attribute values based on the similarities of attribute-value pairs by only considering attributes contained in the core or the reduct of the data set. The RSFit approach provides a faster prediction and can be used for the cases that are not covered by the itemset approach. Empirical studies on UCI data sets and a real world data set demonstrate a significant increase of predicting accuracy obtained from this new integrated approach.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithms, Experimentation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

Keywords

Missing Attribute, Frequent Itemset, Rough Sets Theory

1. INTRODUCTION

How to process data containing missing attribute values is an important task in the data preprocessing stage for data mining applications. Missing attribute values commonly exist in real-world data sets. They may come from the collecting process, or redundant diagnoses tests, change of the experimental design, privacy concerns, unknown data and so on. Discarding all data containing the missing attribute values cannot fully preserve the characteristics of the original data. Understanding and usage of original context and background knowledge to assign the missing values seem to be an optimal approach for handling missing attribute values. But in reality, it is difficult to know the original meaning for this missing data. Various approaches on how to cope with the missing attribute values have been proposed in the past years.

In [6] nine approaches on filling in the missing attribute values were introduced, such as selecting the "most common attribute value", the "concept most common attribute value", "assigning all possible values of the attribute restricted to the given concept", "ignoring examples with unknown attribute values", "treating missing attribute values as special values", "event-covering method" and so on. Although these syntactic approaches provide a direct processing to the missing attribute values, noise is usually brought into the data set as well. Let us consider the approach of "assigning most common attribute values" [6] as an example of assigning missing attribute values. This approach selects the most frequently appeared value from the attribute to the missing value, as explained by the following example.

EXAMPLE 1. *Shown in Table 1 as an example, there are 4 data instances existing in a data set $T(C, D)$, where C is the condition attribute set, D is the decision attribute set, U is the set of data instances, $C = (c_1, c_2, c_3, c_4), D = (0, 1), U = \{u_1, \ldots, u_4\}$. There is a missing value for c_3 in u_2, represented with "?". According to this approach, the most common value for attribute c_3 is 2. However, if we assign the value, the data set becomes inconsistent. u_1 and u_2 will have the same condition attributes with different decision attributes.*

Another approach of "treating missing attribute values as special values" [6] may also bring noise to the original data. The missing value is considered as an individual "unknown"

Table 1: Sample Data Set with Missing Attribute Values

	Condition				Decision
U	c_1	c_2	c_3	c_4	D
1	1	2	2	1	1
2	1	2	?	1	0
3	1	1	3	1	0
4	1	0	2	0	1

value for the attribute. However, the attribute may not have another value in certain applications, as shown by Example 2.

EXAMPLE 2. *Suppose in a data set, the missing attribute is "gender of a patient" with values of either "male" or "female". In case of missing value for this attribute, we cannot assign a "unknown" to this attribute.*

More research efforts are concentrating on how to predict the missing attribute values by obtaining the most information out of the original data set. In [10], support and confidence for the association rules generated from data containing missing attribute values were considered not precise. Rough sets theory was used to estimate the support and confidence values for the generated association rules. For each large item set, based on which association rules would be further generated, the maximal set of tuples that matched, or may match, or certainly did not match, or may not match the item set were listed. The lowest and the highest possible support and confidence values were further defined and computed based on these sets. In [5] a "closest fit" approach was proposed to compare the vectors of all the attribute pairs from a preterm birth data set, and assign the value from the most similar pair to the missing value. In more recent research [4] four interpretations on the meanings of missing attribute values such as "lost" values and "do not care" values are discussed. Different approaches from rough sets theory are demonstrated on selecting values for the individual interpreted meanings.

In addition to the efforts from rough sets theory on processing missing attribute values, strategies from data mining are also widely applied in predicting the missing values. In [8] it is suggested that using regression or inference-based tools on the data set can make a more precise prediction for the missing attributes. A robust algorithm of generating optimal association rules to solve the missing attribute value problem in the testing data set has been discussed in [15]. In [22], the authors discussed a new approach on using association rules generation on completing missing values. Data associations are created based on association rule algorithm and are then used to find the associated values for the missing data. Formulas, based on support, confidence and lift, were applied to help choose better options when multiple matches existed. Recently Zhu and Wu introduced solutions on processing missing attribute values by considering the attribute cost [24]. They point out that the common problems on assigning missing values are that not all the missing values can be predicted by current data mining approaches, and the predictions usually do not bring higher prediction accuracy. They consider in the real world, it is expensive to predict all the missing attributes, therefore a technique is needed on balancing the prediction percentage, the prediction accuracy and the computational cost. They evaluate the importance of different missing data instances by information-gain ratio.

In this paper we concentrate on predicting missing attribute values in the data preprocessing stage. We discuss how to effectively predict missing attribute values combining both the data mining techniques and the rough sets theory. We show how to avoid bias and extract more information from the data itself to predict the missing values.

We are interested in integrating two techniques into our research. One of the techniques is the association rule algorithm, which is well known in data mining for discovering item relationships from large transaction data sets. Priori to the association rule generation, frequent itemsets are generated based on the item-item relations from the large data set according to a certain *support*. Thus the frequent itemsets of a data set represent strong correlations between different items, and the itemsets represent probabilities for one or more items existing together in the current transaction. When considering a certain data set as a transaction data set, the implications from frequent itemsets can be used to find to which attribute value the missing attribute is strongly connected. Thus the frequent itemset can be used for predicting the missing values. We call this approach "itemset-approach" for prediction. Apparently, the larger the frequent itemsets used for the prediction, the more information from the data set itself will be available for prediction, hence the higher the accuracy will be obtained. However, generating frequent itemset for large data set is time-consuming. Although itemsets with higher support need less computation time, they restrict item-item relationships, therefore not all the missing values can be predicted. In order to balance the tradeoff between computation time and the percentage of the applicable prediction, another approach must be taken into consideration.

Rough sets theory, proposed in the 1980's by Pawlak [19], has been used for attribute selection, rule discovery and many knowledge discovery applications in the areas such as data mining, machine learning and medical diagnoses. Core and reduct are among the most important concepts in this theory. A reduct contains a subset of condition attributes that are sufficient enough to represent the whole data set. The intersection of all the possible reduct is the core. Therefore the attributes contained in the reduct or core are more important and representative than the rest of the attributes. Thus by examining only attributes within the same core or reduct to find the similar attribute value pairs for the data instance containing the missing attribute values, we can assign the most relevant value for the missing attribute. Since this method only considers a subset of the data set, which is either the core or the reduct, the prediction is quite fast. This approach "RSFit" is recently proposed in [13], and it is an alternative approach designed for fast prediction. It can be used to predict missing attributes that cannot be predicted by the frequent itemset.

We integrate the prediction based on frequent itemset and RSFit approach into a new approach **ItemRSFit** to predict missing attribute values. This approach can predict missing values from the data itself, therefore less noise is brought into the original data. Experiments on UCI data sets and a real world data set demonstrate our proposed approach on assigning missing attribute values obtains a high accuracy.

The rest of the paper is organized as follows. We introduce frequent itemset generation and rough sets theory in Section 2 and 3. In Section 4, the details of the ItemRSFit and RSFit approach, and the evaluation method are elaborated. We will show the experimental results in Section 5, and Section 6 gives the conclusion remarks.

2. FREQUENT ITEMSET

The association rule algorithm was first introduced in [1], and it can be used to discover rules from transaction datasets. Many contributions on how to efficiently generate frequent itemsets and association rules have been reported [23], [7], [20], [3]. Association rule algorithms can be used to find associations among items from transactions. For example, in *market basket analysis*, by analyzing transaction records from the market, we could use association rule algorithms to discover different shopping behaviors such as, when customers buy bread, they will probably buy milk. This type of behavior can be used in the market analysis to increase the amount of milk sold in the market.

Frequent itemset generation is the first step of the two for association rule generation. Itemsets that frequently occur together in the transactions are generated. Rules based on these itemsets are further discovered to represent the associated relations.

Here we consider the transaction data in the form of decision table for generating frequent itemset. For a rough set approach we define the following concepts.

Definition 1. **Transaction**. The transaction data to the frequent itemset generation is in a form of a decision table $T = (C, D)$, where $C = \{c_1, c_2, \ldots, c_m\}$ is the condition attribute set where m is the number of condition attributes, and $D = \{d_1, d_2, \ldots, d_l\}$ is the decision attribute set where l is the number of decision attributes. $U = \{u_1, u_2, \ldots, u_n\}$ represent the itemsets in T, where n is the number of itemsets in T. Each itemset contains $(m + l)$ items.

Therefore each attribute value is considered to be an item in the transaction.

An association rule [1] is a rule of the form $\alpha \rightarrow \beta$, where α and β represent itemsets which do not share common items.

Definition 2. **Support**. A support of an itemset is the percentage of the number of transactions containing the union of all the items in the itemset to the total number of transactions.

Support can be represented as

$$support = \frac{|\alpha \cup \beta|}{|T|}. \quad (1)$$

Definition 3. **Frequent Itemset**. Frequent itemsets are itemsets that satisfy the minimum support.

Frequent itemset that contains l items is a l-itemset.

3. ROUGH SETS AND RSFIT APPROACH

We briefly introduce rough sets theory [19] as follows. U is the set of objects we are interested in, where $U \neq \phi$.

Definition 4. **Equivalence Relation**. Let R be an equivalence relation over U, then the family of all equivalence classes of R is represented by U/R. $[x]_R$ means a category in R containing an element $x \in U$. Suppose $P \subseteq R$, and $P \neq \phi$, $IND(P)$ is an equivalence relation over U. For any $x \in U$, the equivalence class of x of the relation $IND(P)$ is denoted as $[x]_P$.

Definition 5. **Lower Approximation and Upper Approximation**. X is a subset of U, R is an equivalence relation, the lower approximation of X and the upper approximation of X is defined as:

$$\underline{R}X = \cup\{x \in U | [x]_R \subseteq X\} \quad (2)$$

$$\overline{R}X = \cup\{x \in U | [x]_R \cap X \neq \phi\} \quad (3)$$

respectively.

Reduct and core are further defined as follows [19]. R is an equivalence relation and let $S \in R$. We say, S is dispensable in R, if $IND(R) = IND(R - \{S\})$; S is indispensable in R if $IND(R) \neq IND(R - \{S\})$. We say R is independent if each $S \in R$ is indispensable in R.

Definition 6. **Reduct**. Q is a reduct of P if Q is independent, $Q \subseteq P$, and $IND(Q) = IND(P)$.

An equivalence relation over a knowledge base can have many reducts.

Definition 7. **Core**. The intersection of all the reducts of an equivalence relation P is defined to be the *Core*, where

$$Core(P) = \cap \text{All Reducts of P}.$$

The reduct and the core are important concepts in rough sets theory. Reduct sets contain all the representative attributes from the original data set. The reducts can be used in attribute selection process. There may exist more than one reduct for each decision table. Finding all the reduct sets for a data set is NP-hard [11]. Approximation algorithms are used to obtain reduct sets [2]. The intersection of all the possible reducts is called the *core*. The core is contained in all the reduct sets, and it is the essential of the whole data. Any reduct generated from the original data set cannot exclude the core attributes.

Since it is infeasible to obtain the core attributes by intersecting all the possible reducts, other approaches are proposed to generate the core attributes. Hu et al. [9] introduced a core generation algorithm based on rough sets theory and efficient database operations, without generating reducts. The algorithm is shown in Algorithm 1, where C is the set of condition attributes, and D is the set of decision attributes. $Card$ denotes the count operation in databases, and Π denotes the projection operation in databases.

This algorithm is developed to consider the effect of each condition attribute on the decision attribute. The intuition is that, if the core attribute is removed from the decision table, the rest of the attributes will bring different information to the decision making. Theoretical proof of this algorithm is provided in [9]. The algorithm takes advantage of efficient database operations such as count and projection. This algorithm requires no inconsistency in the data set.

3.1 RSFit Approach

The reduct of a data set is a set of condition attributes whose values are sufficient to correctly predict the decision attribute. The core is the intersection of all possible reducts.

Algorithm 1: Hu's Core Generating Algorithm

input : Decision table $T(C, D)$, C is the condition attributes set; D is the decision attribute set.
output: $Core$, Core attributes set.

$Core \leftarrow \phi$;
for *each condition attribute* $A \in C$ **do**
  if $Card(\Pi(C - A + D)) \neq Card(\Pi(C - A))$ **then**
    $Core = Core \cup A$;
  end
end
return $Core$;

RSFit approach [13] consider attribute-value pairs contained in the core or a reduct set to find the best match for the missing values. This approach is inspired by the "closest fit" approach by Grzymala-Busse [5], however it is different from it. Instead of searching the whole data set for closest matched attribute-value pairs, RSFit searches only the attribute-value pairs within the core or the reduct. The attribute-value pair is defined as following.

Definition 8. **Attribute-Value Pair**. In decision table $T = (C, D)$ as defined in Definition 1, for each u_i in $U = \{u_1, u_2, \ldots, u_n\}$, $(1 \leq i \leq n)$, an attribute-value pair for this data instance is defined to be $u_i = (v_{1i}, v_{2i}, \ldots, v_{mi}, d_i)$, where v_{1i} is the attribute value for condition attribute c_1, v_{2i} is the attribute value for condition attribute c_2, ..., and v_{mi} is the attribute value for condition attribute c_m.

We describe the RSFit approach as follows.

For each missing attribute value, we let the attribute be the "target attribute"(represented as c_k in the following). We consider the situation when the missing attribute values are only existing in the condition attributes, not in the decision attributes. We explain the RSFit approach in detail on how to find the matched value for this target attribute.

Firstly, we obtain the core of the data set $T = (C, D)$ based on the core algorithm in Algorithm 1. If the target attribute c_k does not belong to the core, we include c_k into the core. In case the core set is empty for decision table T, we consider a reduct of T. We use the ROSETTA software [18] for reduct generation. ROSETTA software supports complete data mining process, and many tasks are integrated such as data preprocessing, incomplete data processing, data discretization, reduct sets generation, rule generations and so on. There are a few reduct generation algorithms provided by ROSETTA, such as Genetic reducer, Johnson reducer, Holte1R reducer, Manual reducer, Dynamic reducer, RSES Exhaustive reducer and so on. We use Johnson reducer with the option of full discernibility[1] from ROSETTA GUI version 1.4.41 for a single reduct generation with minimum number of attributes in the reducts. In case of no reducts containing the target attribute c_k, we include the target c_k into the reduct.

[1]For reduct generation, there are two options on discernibility provided by ROSETTA software, which are full discernibility and object related discernibility. With the option of full discernibility, the software will produce a set of minimal attribute subsets that can discern all the objects from each other. With object related discernibility, the software produces reducts that can discern a certain object from all the other objects [17].

Secondly, a new decision table $T' = (C', D)$ is created based on the previous step where $C' = \{c'_1, c'_2, \ldots, c_k, \ldots, c'_{m'}\}$, $1 \leq k \leq m' \leq m$, and $C' \subseteq C$, C' is either the core or the reduct of C, $U' = \{u_1, u_2, \ldots, u_{n'}\}$, $1 \leq n' \leq n$. There are two possibilities for selecting the data instances. One possibility is to include other data instances with missing values to predict the current target attribute value; the other option is to exclude all the other data instances containing missing attribute values. We allow the other missing attribute values existing by designing the proper distance function.

Thirdly, in T', when considering the match cases, there are two possibilities existing. One possibility is that we consider all the data instances having various values of the decision attributes; the other is to consider data instances having the same decision attribute values as the target data instance while finding a matched attribute-value pair. Here we call the first possibility *global match*, and the second *concept match*. We use *global match* in the experiments in our experimental studies[2].

Fourthly, we define the distance function to compute the similarities between different attribute-value pairs. The details of the distance function is elaborated in the following. Let $u_i = (v_{1i}, v_{2i}, \ldots, v_{ki}, \ldots, v_{m'i}, d_i)$ $(1 \leq i \leq n')$ be the attribute-value pair containing the missing attribute value v_{ki} (represented as $v_{ki} =?$) for c_k $(1 \leq k \leq m')$. Distance functions, such as Euclidean distance and Manhattan distance, are used in the instance-based learning to compare the similarity between a test instance and the training instances [21]. We use Manhattan distance [3]to evaluate the distance between an attribute-value pair containing missing attribute values with other attribute-value pairs. This formula is also used in the "closest fit" approach [5]. Let u_j be a data instance from U. The distance between u_j to the target data instance u_i is defined as[4]

$$distance(u_i, u_j) = \frac{|v_{i1} - v_{j1}|}{\max v_1 - \min v_1} + \ldots + \frac{|v_{im} - v_{jm}|}{\max v_m - \min v_m}. \quad (4)$$

For attributes which are the missing attribute values, the distance is set to be 1, which specifies the maximum difference between unknown values. The best match has the smallest difference from the target attribute-value pair. After the best matched attribute-value pair is returned by the algorithm, the corresponding value will be assigned to the target attribute. We consider all the attributes as numerical attributes. In case of symbolic attributes, we convert them to numerical ones during the preprocessing stage.

In case there are multiple matched attribute-value pairs for the missing attribute, one of the values is randomly selected to be assigned to the missing value.

4. ITEMRSFIT APPROACH

The RSFit approach, like other syntactic approaches, cannot provide high prediction precision, although it provides a faster prediction than the "closest fit" approach [13].

[2]RSFit-global and RSFit-concept approaches provide very close prediction accuracies [14].

[3]In our experiments, the prediction results by Manhattan distance and Euclidean distance returned the same accuracy. Because the computation for Manhattan distance is faster, we use Manhattan distance as the distance function.

[4]In the algorithm, $|x|$ returns the absolute value of x.

The unsatisfactory prediction accuracy of RSFit approach can be explained by the fact that this approach does not fully consider the item-item relationship inside the data set. The RSFit uses the subset of a transaction as a whole object to find the similar object. This approach compares the similarity between subsets of transactions and assign the value from the most similar transaction to the missing item. This kind of similarity does not consider the item-item relationship. The frequency of a certain item existing in the transaction in fact indicates how frequently the other item(s) exist(s) in the transaction. The indictions from the strong associations between different items can be discovered by the association rule algorithm.

4.1 Frequent Itemset on Prediction

The frequent itemset generation in association rule algorithm first counts the frequencies of each individual item among the whole transaction. Then based on the 1-itemsets whose support are no less than the predefined minimum support, frequent 2-itemsets are generated. Those itemsets with the occurrence no less than the minimum support are selected for frequent 3-itemsets generation. Frequent l-itemsets are generated based on the frequent $(l-1)$-itemset. The process continues until no new frequent itemsets are found. The l value can also be specified in the itemset generation algorithm to achieve limited itemsets within a preferred time period.

We explain in the following how to use itemset to predict missing attribute values.

Let $T = (C, D)$ be the decision table that contains missing attribute values, where $C = \{c_1, c_2, \ldots, c_k, \ldots, c_m\}$, $1 \leq k \leq m$, and $U = \{u_1, u_2, \ldots, u_n\}$, $1 \leq n$.

Firstly, the data input to the association rule algorithm is prepared. Data instances with missing attribute values are all removed from T, and we call the new decision table T''. T'' does not contain any missing values. Let R a set of data instances containing the missing attribute values, and $T = T'' \cup R$.

Secondly, frequent l-itemsets are generated based on T'' with a given minimum support. Let $Itemsets = \{S_1, S_2, \ldots, S_g\}$, where S_i $(1 \leq i \leq g)$ is a frequent l-itemset generated based on $T = (C, D)$ according to a minimum support, $S_i = \{v_{p_1}, v_{p_2}, \ldots, v_{p_l}\}$, l is the number of items contained in S_i, and v_{p_j} $(1 \leq j \leq l)$ is an attribute value in T.

Thirdly, we use the frequent itemsets generated in the previous step as our knowledge base to find a match for the missing value.

Let $u_i = (v_{1i}, v_{2i}, \ldots, v_{ki}, \ldots, v_{mi}, d_i)$ $(1 \leq i \leq n)$ be the data instance in T containing the missing attribute value v_{ki} (represented as $v_{ki} = ?$) for attribute c_k $(1 \leq k \leq m)$. Search from $Itemsets$ for all the itemsets containing the missing attribute v_k, check which itemset among the itemsets can be **applied** to u_i. We say a frequent itemset can be applied to this data instance if all the items in this itemset, except the missing attribute, have exactly the same attribute values as contained by the data instance that has the missing attribute value. If this itemset can be applied, we assign the attribute value contained in this itemset to the missing attribute. In case there are multiple matched attribute-value pairs for the missing attribute, one of the values is randomly selected to be assigned to the missing value. The following example shows how to use frequent itemset for prediction.

EXAMPLE 3. *Suppose u_i is one of the data instances in T that contain missing attribute values, $u_i = (v_{1i} = 1, v_{2i} = 2, v_{3i} = 4, v_{4i} = ?, v_{5i} = 8)$. An itemset generated from T is $S = \{v_2 = 2, v_3 = 4, v_4 = 6, v_5 = 8\}$. Since all the items in S can be applied to u_i, we assign $v_{4i} = 6$.*

Algorithm 2 shows the pseudo code for this prediction of missing attributes using frequent itemset, which provides linear time prediction.

Algorithm 2: Algorithm for Predicting Missing Values by Frequent Itemset.

input : A sorted data instance $r \in R$ with p missing attributes and a sorted itemset $I \in Itemset$.
output: A set A of attributes values that are the possible missing values of r; or an empty set if the itemset cannot be used to predict the missing attributes of r.

```
A ← ∅
i ← 0 //iterator on I
j ← 0 //iterator on r
while i < |I| and j < |r| and |A| < p do
    if I[i] = r[j] then
        i ← i + 1
        j ← j + 1
    else if I[i] > r[j] then
        j ← j + 1
    else if I[i] < r[j] then
        A ← A ∪ {I[i]}
        i ← i + 1
A ← A ∪ {r[k]|j ≤ k < |r|}
if |A| > p then
    A ← ∅
return A
```

4.2 ItemRSFit Approach

The frequent itemset is generated from the original data set without missing values. We use itemset as our knowledge base to predict missing attribute. Since the knowledge base is generated with a certain support value, when support is high, the item-item relations are stronger, the available knowledge for prediction is less. Missing attribute values from some data instances can be predicted by frequent itemset. We call these data instances *Compatible Records*. There also exist data instances that no possible match can be found to predict the missing values.

Definition 9. **Compatible Record**. A compatible record (CR) is a record whose missing attributes can be predicted by an itemset. More formally, a record r with p missing attributes is a CR if there exists an itemset I such that $|I \cap r| \leq p$.

The missing attributes of a CR are predicted using the technique described in Section 4.1. If a record is not CR, the RSFit method is applied to predict the rest of the missing attribute values. We call this integrated approach **ItemRSFit**. The details on the integrated approach is shown in the following figure.

The procedure of the ItemRSFit approach is described in Figure 1.

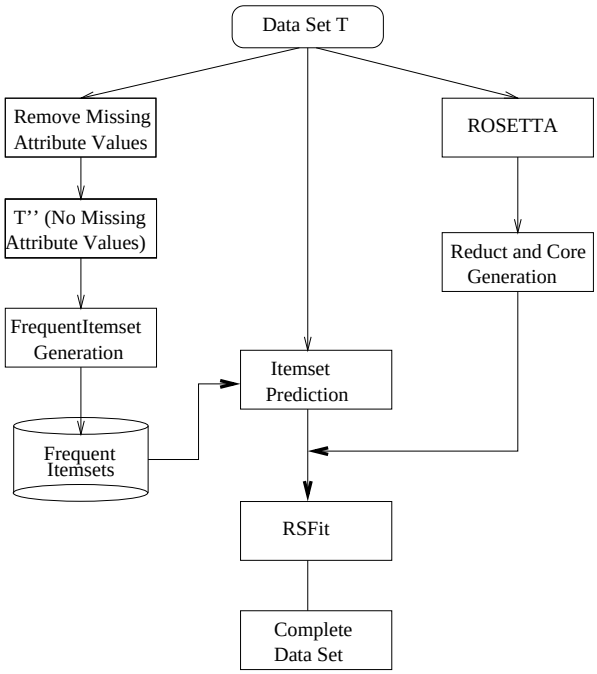

Figure 1: ItemRSFit Approach

4.3 Evaluation Method

We use the following approach to perform the evaluation process. We consider complete data sets as the transaction data set T. For each data set, we randomly select a certain number of missing values among the whole data set to produce n missing attribute values per data set. We then apply both RSFit approach and ItemRSFit approach on predicting missing values, and compare the accuracy of the prediction.

5. EXPERIMENT

The ItemRSFit approach is implemented by Perl and the experiments are conducted on Sun Fire V880, four 900Mhz UltraSPARC III processors. We use apriori frequent itemset generation [3] to generate frequent 5-itemset. The core generation in RSFit approach is implemented with Perl combining the SQL queries accessing MySQL (version 4.0.12). ROSETTA software [18] is used for reduct generation.

5.1 Experiments on Geriatric Care Data

We first perform experiments on a geriatric care data set as shown in Table 2. This data set is an actual data set from Dalhousie University Faculty of Medicine to determine the survival status of a patient giving all the symptoms he or she shows. The data set contains $8,547$ patient records with 44 symptoms and their survival status. We use *survival status* as the decision attribute, and the 44 symptoms of a patient as condition attributes, which includes *education level, the eyesight, hearing, be able to walk, be able to manage his/her own meals, live alone, cough, high blood pressure, heart problem, cough, gender, the age of the patient at investigation* and so on.[5] There is no missing value in this data set. There are 12 inconsistent data entries in the medical data set. After removing these instances, the data contains $8,535$ records. [6] The core attributes for this data set are *eartrouble, livealone, heart, highbloodpressure, eyetrouble, hearing, sex, health, educationlevel, chest, housework, diabetes, dental, studyage.* Table 3 lists the prediction

Table 2: Geriatric Care Data Set

edulevel	eyesight	...	livealone	cough	hbp	heart	...	studyage	sex	livedead
0.6364	0.25	...	0.00	0.00	0.00	0.00	...	73.00	1.00	0
0.7273	0.50	...	0.00	0.00	0.00	0.00	...	70.00	2.00	0
0.9091	0.25	...	0.00	0.00	1.00	1.00	...	76.00	1.00	0
0.5455	0.25	...	1.00	1.00	0.00	0.00	...	81.00	2.00	0
0.4545	0.25	...	1.00	0.00	1.00	0.00	...	86.00	2.00	0
0.2727	0.00	...	1.00	0.00	1.00	0.00	...	76.00	2.00	0
0.0000	0.25	...	0.00	0.00	0.00	1.00	...	76.00	1.00	0
0.8182	0.00	...	0.00	0.00	1.00	0.00	...	76.00	2.00	0
...	...	...	...	...	...	...	...	...	...	...

accuracy comparisons for the RSFit and the ItemRSFit approaches. RSFit is used to predict missing attribute values based on the attribute-value pairs from the core or the reduct. ItemRSFit approach is the new integrated approach introduced in this paper. Table 3 lists the prediction accuracy for both RSFit and ItemRSFit according to different number of missing attribute values and different support values. We also list the number and the percentage of compatible records by only using frequent itemset as knowledge for prediction. In this research, we experiment on geriatric care with 50 to 200 missing attribute values.

From Table 3 we can see, the smaller the support becomes, the more itemsets are generated, the number of compatible records from frequent itemset becomes larger. ItemRSFit approach always obtains higher or the same prediction accuracy as the RSFit approach.

Figure 2 shows the comparison for the number of compatible records by Itemsets prediction according to different support for different number of missing values. Frequent itemsets with lower support value can provide a bigger knowledge base to find predictions, and this is not related to the number of missing values existing in the data set. We can also see from Figure 2 that using itemsets alone cannot predict all the missing values. For instance, when there are 50 missing values existing in the data set, given $support = 10\%$, there are still 8% missing instances that cannot be predicted by the itemsets.

In order to show that ItemRSFit approach obtains better prediction accuracy than RSFit, we show the prediction accuracy comparisons on geriatric care data set with 150 missing attribute values, as shown in Figure 3. We can see from Figure 3 when support value is lower, the prediction accuracy of ItemRSFit is significantly higher than RSFit prediction. This result demonstrates that frequent itemsets as knowledge base can effectively be applied for predicting missing attribute values.

Figure 4 demonstrates the prediction accuracy comparisons for different number of missing attribute values with different support for the geriatric care data set using ItemRSFit. We can see from the comparisons that ItemRSFit approach obtains higher accuracy when support value is lower. The number of missing attribute values existing in the data set does not affect this fact.

Observations. From the experimental results on geri-

[5]Refer to [12] for details about this data set.

[6]Notice from our previous experiments that core generation algorithm cannot return correct core attributes when the data set contains inconsistent data entries.

Table 3: Comparisons on Geriatric Data on Prediction Accuracy (CR: the compatible records)

Data Sets	Average Accuracy(Percentage%)				
Missing Values	RSFit	Support	# CR	% CR	Integrated ItemRSFit
50	64.00%	90%	11	22%	64.00%
		80%	22	44%	68.00%
		70%	26	52%	68.00%
		60%	38	76%	72.00%
		50%	41	82%	70.00%
		40%	43	86%	72.00%
		30%	43	86%	78.00%
		20%	46	92%	90.00%
		10%	46	92%	96.00%
100	69.00%	90%	26	26%	69.00%
		80%	53	53%	74.00%
		70%	58	58%	74.00%
		60%	69	69%	77.00%
		50%	80	80%	75.00%
		40%	87	87%	76.00%
		30%	87	87%	81.00%
		20%	95	95%	87.00%
		10%	95	95%	96.00%
150	73.33%	90%	43	29%	75.33%
		80%	85	57%	79.33%
		70%	94	63%	79.33%
		60%	120	80%	80.00%
		50%	133	89%	81.33%
		40%	137	91%	82.00%
		30%	137	91%	83.33%
		20%	142	95%	89.33%
		10%	142	95%	96.67%
200	73.50%	90%	39	20%	73.50%
		80%	103	52%	77.00%
		70%	118	59%	76.50%
		60%	146	73%	75.50%
		50%	169	84%	73.50%
		40%	182	91%	79.00%
		30%	182	91%	79.50%
		20%	192	96%	88.50%
		10%	194	96%	96.00%

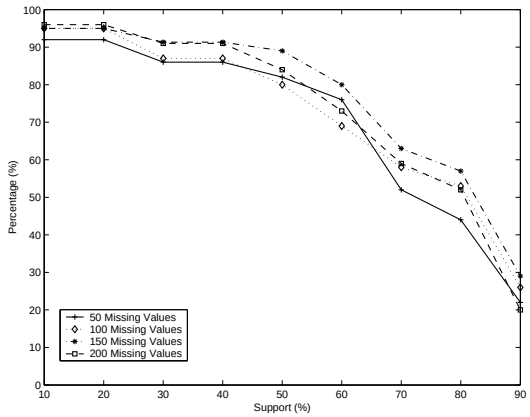

Figure 2: Comparisons on the Percentage of Compatible Records for Geriatric Care Data using Frequent Itemsets to Predict

atric care data set shown in Figures 2, 3, 4, we observe that

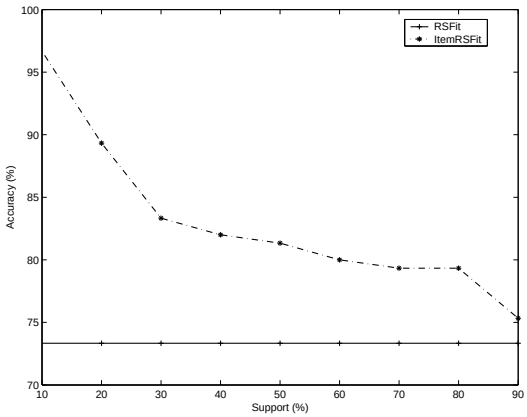

Figure 3: Accuracy Comparisons for Geriatric Care Data with 150 Missing Attribute Values

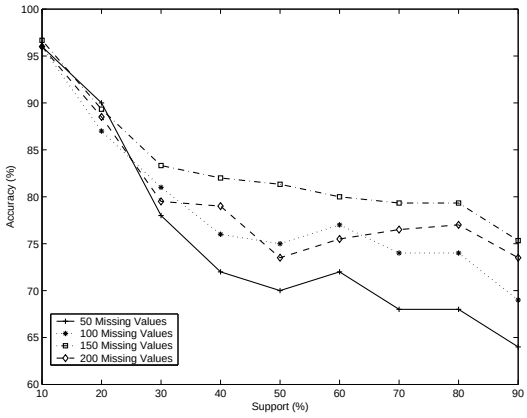

Figure 4: Accuracy Comparisons for Geriatric Care Data with Different Number of Missing Attribute Values

- The prediction accuracy for ItemRSFit approach increases while the support value decreases.
- Frequent Itemset approach can provide a higher prediction by itself. But this approach cannot predict all the missing values in the geriatric care data set.
- For the ItemRSFit approach on geriatric care data, the highest accuracy is obtained when $support = 10\%$; the lowest accuracy is obtained when $support = 90\%$. This can be explain as the following. "Support" is a measure to evaluate the occurrence of both the antecedents and the consequents of an association rule in the data set. The higher the support is, the more frequent this occurrence becomes, the less knowledge for prediction is obtained. When the support value is increased, less matched cases are found from the itemset approach, therefore more missing values have to be predicted by RSFit approach.
- The lowest accuracy of the ItemRSFit approach is equal to the accuracy from RSFit approach. RSFit approach gives the baseline prediction accuracy for the ItemRSFit approach.

- For different number of missing attribute values, frequent itemset with the lowest support brings the highest prediction accuracy. The frequent itemset alone as the knowledge base to predict the missing values cannot fully find all the matches for the missing value for geriatric care data.

5.2 Experiments on UCI Data Sets

In the experiments on the UCI data sets [16] we study how the ItemRSFit approach can be applied for predictions on different types of data sets. We experiments on data sets with no missing attribute values. For each data set, we randomly select 5% of the total possible missing values (total number of condition attributes × total number of data instances) as missing attribute values, and list the prediction accuracy comparisons for the ItemRSFit and RSFit approaches according to different support values.

Abalone Data This data set is used to predict the age of abalone from physical measurements. There are $4,177$ instances and 8 condition attributes in this data set. There are no missing attribute values or inconsistent data instances in the data set. For this data set, we randomly select 0.5% missing attribute values, which is 167 missing values. The prediction comparisons between RSFit and ItemRSFit approaches are shown in Figure 5.

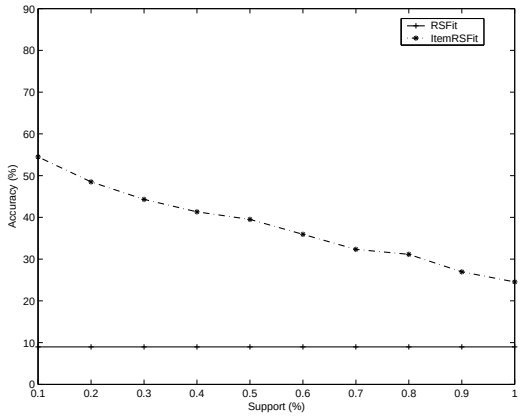

Figure 5: Accuracy Comparisons for Abalone Data with 0.5% Missing Attribute Values

Observation. As we can see, when support value decreases, the prediction accuracy increases.

Lymphography Data The data set contains 148 instances and 18 condition attributes. There are no missing attribute values in this data. We check that there is no inconsistent data. The core is empty for this data set. We randomly select 133 missing attribute values from this data set, which is around 5% of the data set. The prediction comparisons between RSFit and ItemRSFit approaches are shown in Figure 6.

Observation. As we can see, when support value decreases, the prediction accuracy increases. We further explore the prediction accuracy on smaller number of missing values with this data set, as shown in Figure 7. For 10 missing values, when support reaches less than or equal to 20%, the accuracy is 100%. This observation implies that less number of frequent itemsets can also be used to provide high predictions for missing attributes.

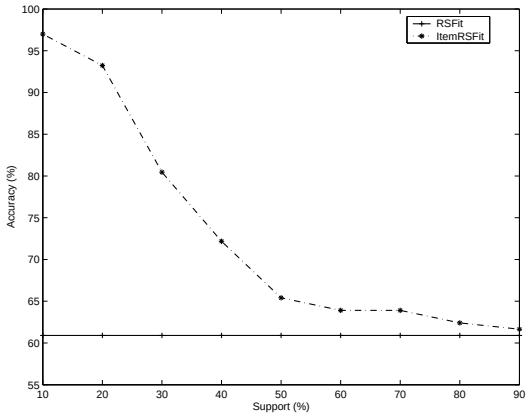

Figure 6: Accuracy Comparisons for Lymphography Data with 5% Missing Attribute Values

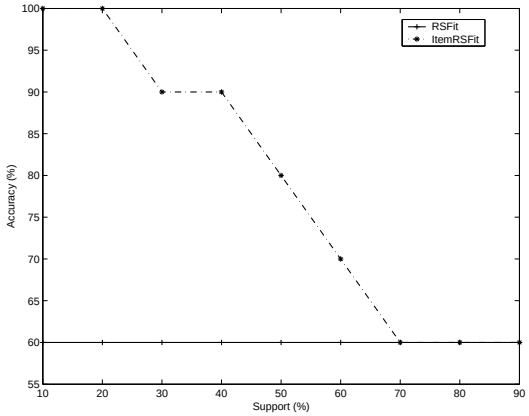

Figure 7: Accuracy Comparisons for Lymphography Data with 10 Missing Attribute Values

Glass Data This data set is used for the study of classification of types of glass by criminological investigation. At the scene of the crime, the glass left can be used as evidence. There are 214 instances and 9 condition attributes. There are no missing attribute values or inconsistent data instances. We randomly select 96 missing attribute values from this data set, which is around 5% of the data set. The prediction comparisons between RSFit and ItemRSFit approaches are shown in Figure 8.

Observation. For glass data set, the support values rank from 1% to 10% for frequent itemset generation. We can see as support decreases, the prediction accuracy increases. The highest prediction accuracy obtained when $support = 1\%$. The ItemRSFit always achieves higher prediction than RSFit.

Iris Data For Iris data set, there are 4 condition attributes, 150 instances. There is no inconsistent data existing in the data. We first use core algorithm to generate core attributes, but the result is empty. This means none of the attributes is indispensable. We randomly select 30 missing attribute values from this data set, which is around 5% of the data set. The prediction comparisons between RSFit and ItemRSFit approaches are shown in Figure 9.

Observation. For iris data set, the support values rank

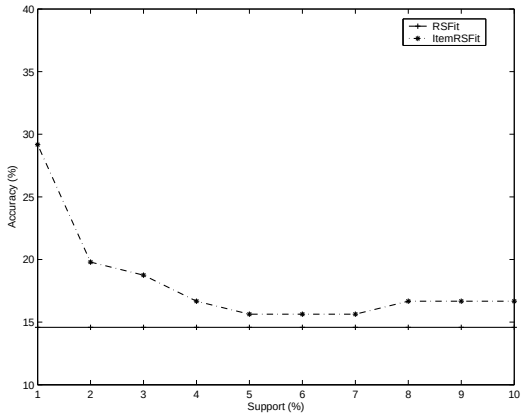

Figure 8: Accuracy Comparisons for Glass Data with 5% Missing Attribute Values

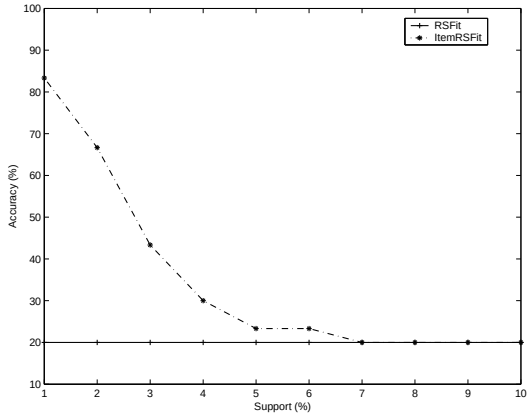

Figure 9: Accuracy Comparisons for Iris Data with 5% Missing Attribute Values

from 1% to 10% for frequent itemset generation. We can see as support decreases, the prediction accuracy increases. The highest prediction accuracy of 83.33% is obtained when $support = 1\%$. The ItemRSFit always achieves higher prediction than RSFit. It is also interesting to notice how drastically the prediction accuracy increases from 20% to 83.33% within a small range of support values decreases from 7% to 1%.

5.3 Discussions and Related Work

Experimental results from both the real-world geriatric care data set and UCI data sets have demonstrated the high prediction characteristics of the proposed ItemRSFit approach on processing data with missing attribute values. Frequent itemset can be used as knowledge base to predict missing attribute values.

We find the approach introduced in [22] close to our work. An approach of using association rules generations on completing missing values is discussed. However, our proposed ItemRSFit approach is quite different from the approach introduced in [22]. First, only frequent 1-itemset and 2-itemset are used in [22] to find the possible values for the missing data, and data associations with missing attribute on the consequent part are used for prediction. It is not discussed how much percentage of the missing data can be predicted with the data association. We use frequent 5-itemset as knowledge base for prediction. We explore the relations between different support and the percentage of the *compatible records* using frequent itemsets as shown in Figure 2. Second, in case there is no match from the data association, the missing value is assigned by the most common value of the missing attribute in [22]. We use frequent itemsets as knowledge base for prediction, and RSFit approach for the *non-compatible records* that the itemset cannot be applied, which guarantee that more important attributes are taking into considerations while predicting attributes. The proposed ItemRSFit approach provides predictions based on the data domain itself, which better preserves the originality of the data sets and avoids noises. Third, our approach is also more efficient, because we do not need to generate data association based on both support and confidence for prediction. Only support is used for frequent itemsets generations in ItemRSFit approach.

6. CONCLUDING REMARKS AND FUTURE WORK

An efficient approach on predicting missing attribute values is proposed in this paper. The experimental results show this new approach always obtains higher prediction accuracy than RSFit. The prediction relies on the data itself as knowledge bases and therefore the predicted values are not biased. The ItemRSFit approach can be applied for data preprocessing in data mining and knowledge discovery tasks.

In this paper, the ItemRSFit approach uses the RSFit approach on predicting non-compatible records. We would like to experiment on other techniques on predicting missing values for the non-compatible records. In our research, we also adopt the strategies used by [24] on balancing the computational cost and the prediction accuracy. Lower support value can bring a higher prediction accuracy, however frequent itemset with lower support requires more time for computation than frequent itemset with higher support. In the future, we are interested to explore a satisfactory balance between the support value and the prediction accuracy. Given the available computational cost and the affordable computation time, it is interesting to explore to what percentage the missing attributes can be effectively predicted, and what are the most effective attributes to be predicted. In case of a higher prediction cost, the idea of giving more important attributes higher priorities for predictions can be applied as heuristics.

Acknowledgements

We gratefully acknowledge the financial support of the Natural Science and Engineering Research Council of Canada. Thank Dr. Arnold Mitnitski from Department of Medicine, Dalhousie University for sharing the Geriatric Care data set. Thank Claude-Guy Quimper from University of Waterloo for helpful discussions on the algorithm issues.

7. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proc. 20th Int. Conf. Very Large*

Data Bases, VLDB, pages 487–499. Morgan Kaufmann, 12–15 1994.

[2] J. G. Bazan, H. S. Nguyen, S. H. Nguyen, P. Synak, and J. Wróblewski. Rough set algorithms in classification problem. pages 49–88, 2000.

[3] C. Borgelt. Efficient implementations of apriori and eclat. In *Proceedings of the FIMI'03 Workshop on Frequent Itemset Mining Implementations, Melbourne, Florida, USA, November, CEUR Workshop Proceedings*, 2003.

[4] J. W. Grzymala-Busse. Incomplete data and generalization of indiscernibility relation, definability, and approximations. In *RSFDGrC (1)*, pages 244–253, 2005.

[5] J. W. Grzymala-Busse, W. J. Grzymala-Busse, and L. K. Goodwin. Coping with missing attribute values based on closest fit in preterm birth data: A rough set approach. *Computational Intelligence*, 17(3):425–434, 2001.

[6] J. W. Grzymala-Busse and M. Hu. A comparison of several approaches to missing attribute values in data mining. In *Rough Sets and Current Trends in Computing*, pages 378–385, 2000.

[7] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. Naughton, and P. A. Bernstein, editors, *2000 ACM SIGMOD Intl. Conference on Management of Data*, pages 1–12. ACM Press, 05 2000.

[8] K.-M. Han, J. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2000.

[9] X. Hu, T. Y. Lin, and J. Han. A new rough sets model based on database systems. *Fundamenta Informaticae*, 59(2-3):135–152, 2004.

[10] M. Kryszkiewicz. Association rules in incomplete databases. In *PAKDD*, volume 1574 of *Lecture Notes in Computer Science*, pages 84–93. Springer, 1999.

[11] M. Kryszkiewicz and H. Rybinski. Finding reducts in composed information systems. In *RSKD*, pages 261–273, 1993.

[12] J. Li and N. Cercone. Empirical analysis on the geriatric care data set using rough sets theory. Technical Report CS-2005-05, School of Computer Science, University of Waterloo, 2005.

[13] J. Li and N. Cercone. Assigning missing attribute values based on rough sets theory. In *IEEE Granular Computing*, Atlanta, USA, 2006.

[14] J. Li and N. Cercone. Comparisons on different approaches to assign missing attribute values. Technical Report CS-2006-04, School of Computer Science, University of Waterloo, 2006.

[15] J. Li, R. Topor, and H. Shen. Construct robust rule sets for classification. In *Proceedings of the eighth ACMKDD international conference on knowledge discovery and data mining*, pages 564 – 569, Edmonton, Canada, 2002. ACM press.

[16] H.-S. B. C. Newman, D.J. and C. Merz. UCI repository of machine learning databases, 1998.

[17] A. Øhrn. Rosetta technical reference manual. Department ofComputer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway. May 25, 2001.

[18] A. Øhrn. *Discernibility and Rough Sets in Medicine: Tools and Applications.* PhD thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim Norway, 1999.

[19] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[20] J. Pei, J. Han, and R. Mao. CLOSET: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[21] F.-E. Witten, I.H. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, 2000.

[22] C.-H. Wu, C.-H. Wun, and H.-J. Chou. Using association rules for completing missing data. In *HIS*, pages 236–241. IEEE Computer Society, 2004.

[23] M. Zaki and C. Hsian. Charm: an efficient algorithm for closed association rule mining. *Technical Report, RPI, Troy NY*, 1999.

[24] X. Zhu and X. Wu. Cost-constrained data acquisition for intelligent data preparation. *IEEE Transactions on Knowledge And Data Engineering*, 17(11):1542–1556, 2005.