Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Mining fuzzy association rules from questionnaire data

Yen-Liang Chen *, Cheng-Hsiung Weng

Department of Information Management, National Central University, Chung-Li, Taiwan 320, Taiwan, ROC

ARTICLE INFO

Article history:
Received 13 June 2007
Received in revised form 19 May 2008
Accepted 10 June 2008
Available online 18 June 2008

Keywords:
Data mining
Fuzzy sets
Association rules
Questionnaire data

ABSTRACT

Association rule mining is one of most popular data analysis methods that can discover associations within data. Association rule mining algorithms have been applied to various datasets, due to their practical usefulness. Little attention has been paid, however, on how to apply the association mining techniques to analyze questionnaire data. Therefore, this paper first identifies the various data types that may appear in a questionnaire. Then, we introduce the questionnaire data mining problem and define the rule patterns that can be mined from questionnaire data. A unified approach is developed based on fuzzy techniques so that all different data types can be handled in a uniform manner. After that, an algorithm is developed to discover fuzzy association rules from the questionnaire dataset. Finally, we evaluate the performance of the proposed algorithm, and the results indicate that our method is capable of finding interesting association rules that would have never been found by previous mining algorithms.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Association rule mining is an important data mining method that determines consumer purchasing patterns in transaction databases [7,15]. Many applications have used association rule mining techniques to discover useful information, including market basket analysis, product recommendation, web page pre-fetch, gene regulation pathways identification, medical record analysis, and so on. Agrawal et al. [1] first introduced the problem, and defined it as finding all rules from transaction data satisfying the minimum support and the minimum confidence constraints. Briefly, an association rule mining algorithm works in two steps: (1) generate all frequent itemsets that satisfy *minsup* and (2) generate all association rules that satisfy *minconf* using the large itemsets.

Due to its great success and widespread usage, many variants of association rule mining algorithms have been proposed. These algorithms can be roughly classified into three categories, according to the data types they can handle: nominal/Boolean data [1,7,11,19,26,29], ordinal data [10], and quantitative data [5,13,16–18,22,31,32]. The first category of algorithms views a transaction as a set of items with nominal/Boolean values, the second as a set of items with ordinal values, and the last as a set of items with quantitative values. Based on different assumptions about the underlying data, different methods have been developed to help discover association rules between items.

A questionnaire is a data collection method that a respondent completes in written format [23]. Questionnaire surveys are an important part of marketing [14] and customer relationship management [8]. The use of questionnaires is even popular in schools to collect students' opinions of teaching performance. According to Marshall [23], the types of questions in the questionnaires can be roughly classified into two categories, open-ended and closed-ended questions. Accordingly, the answers to these two types of questions constitute open questionnaire data and closed questionnaire data, respectively.

To collect open questionnaire data, structured questions must be supplemented with open-ended questions to get respondents to answer the problems in their own words. Researchers have proposed few methods for analyzing open questionnaire data, using multivariate analysis techniques such as cluster analysis [3] and correspondence analysis [6]. Furthermore, Yamanishi and Li [35] calculate associations between word pairs based on their co-occurrences in open answers and then visually present the words and associations on a two-dimensional map.

Text mining is the automated or partially automated processing of text. It involves imposing structure upon text so that relevant information can be extracted from it [25,27]. The applications of text mining include [24,33]: information extraction, topic tracking, summarization, categorization, clustering, concept linking, and question answering. Since the responses in open questionnaire data (i.e. open answers) are expressed in words, like text documents, text mining techniques could be also applied to analyzing open answers, such as text clustering techniques [9,21] or the self-organizing map technique [20].

Web mining refers to the use of data mining techniques to automatically retrieve, extract, and evaluate (generalize/analyze) information for knowledge discovery from web documents. Arotaritei and Mitra [4] consider that web mining can be broadly categorized as: (1) Web content mining of multimedia documents. (2) Web

* Corresponding author. Tel.: +886 3 4267266; fax: +886 3 4254604.
E-mail address: ylchen@mgt.ncu.edu.tw (Y.-L. Chen).

0950-7051/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2008.06.003

structure mining of inter-document links. (3) Web usage mining of the data generated by the users' interactions with the web. Among them, the web content mining techniques are the most appropriate for analyzing open questionnaire data (i.e. open answers). As the same as the text mining techniques, these web content mining techniques also focus on cluster analysis [28] and correspondence analysis [30].

The above discussions indicate that there were some previous researches using "text mining" and "web mining" techniques to discover knowledge from open questionnaire data. However, in regard to closed questionnaire data no systematic mining method has been proposed so far to discover knowledge from them. Therefore, an unanswered question still exists, that is, how to extract knowledge from closed questionnaire data. This motivated us to study how to discover fuzzy association rules from closed questionnaire data.

Before we further discuss the potential problems that we may encounter if we want to discover knowledge from closed questionnaire data, we use Fig. 1 to show how our work can be placed in the context of previous work.

According to Marshall [23], closed questionnaire data (i.e. closed answers) includes the following five data types: (1) Category where there is a list of mutually exclusive categories; e.g., gender is either male or female. (2) List where the respondents can select more than one response from a list of categories; e.g., a user can have several favorite sports. (3) Quantity where the response is a number; e.g., how many times have you been examined during this pregnancy? (4) Ranking/scales, like the Likert scale, where the respondents choose from a list of values on an ordinal scale indicating the degree of agreement or disagreement with a statement; e.g., on a scale of 1–7, how would you rate your level of satisfaction with the class? (5) Linguistic ranking/scales where the respondents choose from a list of ranked linguistic terms; e.g., is he very tall, tall, short, or very short? If we allow users multiple-choices in types (4) and (5), we will have two additional data types: (6) Multiple-choice ranks and (7) multiple-choice linguistic ranks.

In short, users' response data in questionnaires can be classified into categories (Nominal), lists (Multiple-choice nominal), numbers (Quantitative), ranks (Ordinal), linguistic ranks (Fuzzy ordinal), multiple-ranks (Multiple-choice ordinal), and multiple-linguistic ranks (Multiple-choice fuzzy ordinal). The following discussion explains why we cannot apply traditional mining algorithms to discover association rules from questionnaire data.

First, the traditional approaches are designed for handling nominal/Boolean data, ordinal data, or quantitative data exclusively. Currently, no algorithms have been developed to handle these three data types simultaneously. Questionnaire data may have up to seven types of data, four of which are new and have not been considered by previous research.

Second, in previous approaches, a transaction is a set of items, each of which is associated with a value. For example, in nominal/Boolean data, the associated value is a choice from a set of exclusive categories. Unfortunately, in questionnaire data, it is possible to assign multiple values to a single item. For instance, a user's favorite fruit may include apples, bananas, and grapes. Since previous algorithms have not addressed the multiple-choice problem, we must consider this issue in our present research.

Third, although previous algorithms for handling quantitative data could discover association rules with linguistic terms (fuzzy rules), these algorithms made the assumption that the underlying data was purely quantitative. In other words, these algorithms find fuzzy rules from purely quantitative data, and they cannot deal with raw data involving linguistic terms. Since linguistic ranking/scales data are likely to appear in a questionnaire dataset, it is necessary to resolve this issue also.

Therefore, in order to discover rules from a questionnaire dataset, we need a brand new approach that can deal with different data types occurring simultaneously, including categories, lists, numbers, ranks, linguistic ranks, multiple-ranks, and multiple-linguistic ranks. Traditional approaches have only handled categories, numbers, and ranking/scales individually. Thus, we not only have to consider how our algorithm can simultaneously handle these three previous data types, but we must also consider how the other four new data types can be included.

The goal of this paper is to develop an algorithm that can handle all seven data types at the same time, allowing us to discover association rules from questionnaire data. Since our raw data may involve linguistic terms, which are expressed by fuzzy sets, we naturally adopted fuzzy techniques, so that all data types could be represented and operated from fuzzy points of view. Furthermore, since the linguistic terms in the underlying data may appear in the rules, we must extend the crisp association rules to fuzzy association rules. Therefore, the goal of this paper is to mine fuzzy association rules from questionnaire data.

The rest of this paper is organized as follows. First, we define the problem in Section 2. The proposed algorithm and an example are illustrated in Section 3. Section 4 uses questionnaire data regarding teaching/learning evaluations as a case study, demonstrating that the proposed algorithm can discover interesting patterns from questionnaire data. Conclusions and future works are discussed in Section 5.

2. Problem definition

In this section, we define the problem of mining fuzzy association rules from questionnaire data. First, several kinds of items

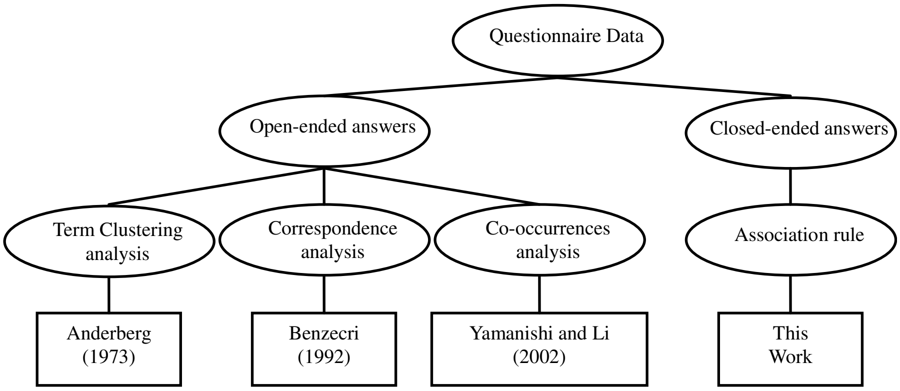

Fig. 1. The taxonomy of questionnaire data mining.

used in the proposed algorithm are introduced. Second, we define the membership degrees for each kind of item and use them to calculate itemsets' supports.

Definition 1. Let $IT = \{it_1, it_2, \ldots, it_m\}$ be a set of all items. A q-item is denoted as (it_i, q), where $it_i \in IT$ is the item name and q is the value of it_i. Semantically, (it_i, q) means that q is the respondent's answer to the ith question in questionnaire.

Seven data types may appear in a questionnaire: *categories* (nominal), *lists* (multiple-choice nominal), *numbers* (quantitative), *ranks* (ordinal), *multiple-ranks* (multiple-choice ordinal), *linguistic ranks*(fuzzy ordinal), and *multiple-linguistic ranks* (multiple-choice fuzzy ordinal). Therefore, a q-item could be a *category* q-item, *list* q-item, *number* q-item, *rank* q-item, *multiple-rank* q-item,*linguistic rank* q-item, or *multiple-linguistic rank* q-item.

Example 1. Assume we have a dataset containing 10 transactions, as shown in Table 1. The values "blackboard instruction", "computer-aided instruction", and "audio-visual instruction" are abbreviated as BBI, CAI, and AVI, respectively. In this example, there are five items i_1, i_2, i_3, i_4, i_5. The q-item could be *category* q-item (i_1, male), *list* q-item (i_2, (BBI, CAI)), *number* q-item (i_3, 6), *rank* q-item (i_4, 4), *multiple-rank* q-item (i_4, (1, 2)), *linguistic rank* q-item (i_5, good), and *multiple-linguistic rank* q-item (i_5, (good, very good)).

Before we detail the various definitions of q-item, q-itemset, *rule* q-item, *rule* q-itemset, and the supports of *rule* q-itemsets, we have illustrated the key ideas in Fig. 2. The low level shows the q-items that may appear in q-itemsets of the questionnaire dataset, while the middle shows the q-items that may appear in a *rule* q-itemset. The arrows between them illustrate how the q-items in data can be mapped to the q-items in *rule* q-itemsets. The upper level indicates that the fuzzy rules are generated from *rule* q-itemsets. Since only three types of q-items appear in the *rule* q-itemsets, our rules also only contain these three kinds of q-items.

Definition 2. Assume that we have a *category* q-item $a_i = (it_i, q_i)$ and a *category* q-item $b_j = (ic_j, f_j)$. Let $sup(a_i, b_j)$ denote the degree to which a_i matches b_j. Then, $sup(a_i, b_j)$ is given as follows:

$$sup(a_i, b_j) = \begin{cases} 1, & \text{if } it_i = ic_j \quad \text{and} \quad q_i = f_j \\ 0, & \text{otherwise} \end{cases}$$

Example 2. Given a *category* q-item $a_1 = (it_1, male)$ and a *category* q-item $b_1 = (ic_1, male)$, the degree $sup(a_1, b_1) = sup((it_1, male), (ic_1, male)) = 1.0$ if $it_1 = ic_1$.

Definition 3. Assume that we have a *list* q-item $a_i = (it_i, q_i)$ and a *list* q-item $b_j = (ic_j, f_j)$. Let $|f_j|$ and $|q_i \cap f_j|$ denote the numbers of values in f_j and $q_i \cap f_j$, respectively. Then, the degree to which a_i matches b_j, $sup(a_i, b_j)$, can be given as follows:

$$sup(a_i, b_j) = \begin{cases} \frac{|q_i \cap f_j|}{|f_j|}, & \text{if } it_i = ic_j \\ 0, & \text{otherwise} \end{cases}$$

Table 1
A questionnaire dataset

TID	Itemsets
1	(i_1, male), (i_2, (BBI, CAI)), (i_3, 3), (i_4, 3), (i_5, average)
2	(i_1, male), (i_2, (BBI, CAI)), (i_3, 5), (i_4, 4), (i_5, (good, very good))
3	(i_1, female), (i_2, (BBI, CAI)), (i_3, 2), (i_4, 2), (i_5, poor)
4	(i_1, female), (i_2, (CAI)), (i_3, 2), (i_4, (1, 2)), (i_5, poor)
5	(i_1, male), (i_2, (BBI, CAI)), (i_3, 5), (i_4, (4, 5)), (i_5, (good, very good))
6	(i_1, male), (i_2, (BBI, CAI)), (i_3, 5), (i_4, 4), (i_5, good)
7	(i_1, female), (i_2, (CAI)), (i_3, 2), (i_4, 2), (i_5, poor)
8	(i_1, male), (i_2, (BBI)), (i_3, 4), (i_4, (4, 5)), (i_5, (average, good))
9	(i_1, female), (i_2, (BBI, CAI, AVI)), (i_3, 5), (i_4, 4), (i_5, (good, very good))
10	(i_1, male), (i_2, (AVI)), (i_3, 5), (i_4, 4), (i_5, good)

Example 3. Suppose we have a *list* q-item $a_2 = (it_2, \text{(BBI)})$ and a *list* q-item $b_2 = (ic_2, \text{(BBI, CAI)})$. If $it_2 = ic_2$, the degree $sup(a_2, b_2) = 1/2 = 0.5$.

Definition 4 (*Fuzzification*). Suppose we have a universe of discourse X in a *quantitative domain*, where each element x belongs to X. Then, a fuzzy set F is characterized by a membership function $m_F\,(x)$, which maps x to a membership degree in interval [0, 1].

Example 4. Assume that we have six membership functions, R_{short}, R_{middle}, R_{long}, S_{low}, S_{middle}, and S_{high}. The first three are for the review time and the others are for the scores. From these six membership functions, we know that $R_{short}(2) = 0.5$, $R_{middle}(3) = 1.0$, $R_{long}(5) = 1.0$, $S_{low}(73) = 0.7$, $S_{middle}(81) = 0.9$, and $S_{high}\,(92) = 1.0$.

$$R_{short}(q) = \begin{cases} 1, & \text{if } q \leqslant 1 \\ \frac{3-q}{3-1}, & \text{if} 1 \leqslant q \leqslant 3 \end{cases} \tag{1}$$

$$R_{middle}(q) = \begin{cases} \frac{q-1}{3-1}, & \text{if } 1 \leqslant q \leqslant 3 \\ 1, & \text{if } q = 3 \\ \frac{5-q}{5-3}, & \text{if } 3 \leqslant q \leqslant 5 \end{cases} \tag{2}$$

$$R_{long}(q) = \begin{cases} \frac{q-3}{5-3}, & \text{if } 3 \leqslant q \leqslant 5 \\ 1, & \text{if } 5 \leqslant q \end{cases} \tag{3}$$

$$S_{low}(q) = \begin{cases} 1, & \text{if } q \leqslant 70 \\ \frac{80-q}{80-70}, & \text{if } 70 \leqslant q \leqslant 80 \end{cases} \tag{4}$$

$$S_{middle}(q) = \begin{cases} \frac{q-70}{80-70}, & \text{if } 70 \leqslant q \leqslant 80 \\ 1, & \text{if } q = 80 \\ \frac{90-q}{90-80}, & \text{if } 80 \leqslant q \leqslant 90 \end{cases} \tag{5}$$

$$S_{high}(q) = \begin{cases} \frac{q-80}{90-80}, & \text{if } 80 \leqslant q \leqslant 90 \\ 1, & \text{if } 90 \leqslant q \end{cases} \tag{6}$$

Definition 5. Assume that we have a *number* q-item $a_i = (it_i, q_i)$, a *linguistic* q-item $b_j = (ic_j, f_j)$, and a membership function (FS_{f_j}), where $FS_{f_j}(q_i)$ denotes the membership degree to which q_i belongs to f_j. Then, the degree to which a_i matches b_j, $sup(a_i, b_j)$ can be given as follows:

$$sup(a_i, b_j) = \begin{cases} FS_{f_j}(q_i), & \text{if } it_i = ic_j \\ 0, & \text{otherwise} \end{cases}$$

Example 5. Suppose we have a *number* q-item $a_3 = (it_3, 4)$, a *linguistic* q-item $b_3 = (ic_3, long)$, and a membership function (R_{long}), as shown in Example 4. Then, the degree $sup(a_3, b_3) = sup\,((it_3,4), (ic_3, long)) = R_{long}(4) = 0.5$, if $it_3 = ic_3$.

Definition 6. For *rank* i and *linguistic rank* f_j, let $sim^{RL}(i, f_j)$ denote the similarity between *rank* i and *linguistic rank* f_j. In this study, the *RL* similarity matrix Sim_{RL} stores all the similarities between *rank* q-items and *linguistic rank* q-items.

Example 6. Table 2 is an example of *RL* similarity matrix Sim_{RL}. From this matrix, we know that $sim^{RL}(1, \textit{very poor}) = 0.86$, $sim^{RL}(1, \textit{poor}) = 0.43$, $sim^{RL}(2, \textit{very poor}) = 0.43$, $sim^{RL}(2, \textit{poor}) = 0.86$, $sim^{RL}(2, \textit{average}) = 0.43, \ldots, sim^{RL}(5, \textit{very good}) = 0.76$.

Definition 7. Assume that we have a *rank* q-item $a_i = (it_i, q_i)$, a *linguistic* q-item $b_j = (ic_j, f_j)$, and an *RL* similarity matrix Sim_{RL}, where $sim^{RL}(q_i, f_j)$ denotes the similarity between q_i and f_j. Then, the degree to which a_i matches b_j, $sup(a_i, b_j)$ can be given as follows:

$$sup(a_i, b_j) = \begin{cases} sim^{RL}(q_i, f_j), & \text{if } it_i = ic_j \\ 0, & \text{otherwise} \end{cases}$$

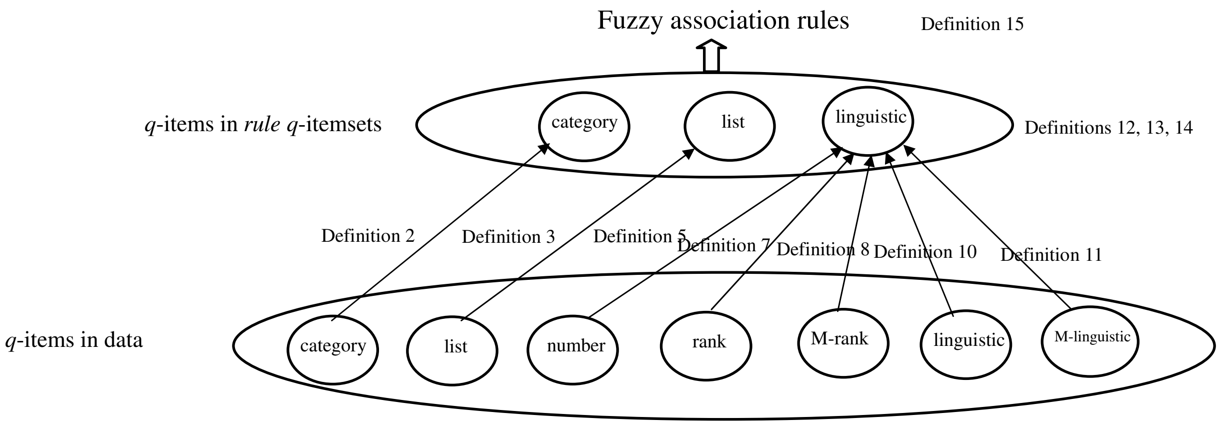

Fig. 2. The relationships between *q*-items and *rule q*-itemsets.

Table 2
An *RL* similarity matrix Sim_{RL}

	Very poor	*Poor*	*Average*	*Good*	*Very good*
1	0.86	0.43	0	0	0
2	0.43	0.86	0.43	0	0
3	0	0.5	1	0.5	0
4	0	0	0.43	0.86	0.43
5	0	0	0	0.38	0.76

Example 7. Suppose we have a *rank q*-item $a_4 = (it_4, 4)$, a *linguistic q*-item $b_4 = (ic_4, good)$, and an *RL* similarity matrix Sim_{RL}, as shown in Table 2. Then, the degree $sup(a_4,\ b_4) = sup\ ((it_4,4),\ (ic_4, good)) = 0.86$, if $it_4 = ic_4$.

Definition 8. Assume that we have a *multiple-ranks q*-item $a_i = (it_i, (q_{i1}, q_{i2}))$, a *linguistic q*-item $b_j = (ic_j, f_j)$, and an *RL* similarity matrix Sim_{RL}. Then, the degree to which a_i matches b_j, $sup\ (a_i, b_j)$ can be given as follows:

$$sup(a_i, b_j) = \begin{cases} \max(sim^{RL}(q_{i1}, f_j), sim^{RL}(q_{i2}, f_j)), & \text{if } it_i = ic_j \\ 0, & \text{otherwise} \end{cases}$$

Example 8. Suppose we have a *multiple-ranks q*-item $a_5 = (it_5,(3,4))$, a *linguistic q*-item $b_5 = (ic_5, good)$, and an *RL* similarity matrix Sim_{RL}, as shown in Table 2. The degree $sup(a_5, b_5) = sup\ ((it_5,(3,4)),\ (ic_5,\ good)) = \max(sim^{RL}(3,\ good),\ sim^{RL}(4,\ good)) = \max(0.5, 0.86) = 0.86$, if $it_5 = ic_5$.

Definition 9. For two *linguistic ranks*, f_i and f_j, let $sim^{LL}(f_i, f_j)$ denote the similarity between the two *linguistic ranks* f_i and f_j. Assume $sim^{LL}(f_i, f_j) = sim^{LL}(f_j,f_i)$. In this study, the *LL* similarity matrixSim_{LL} stores the similarity values between *linguistic rank q*-items.

Example 9. Table 3 is an *LL* similarity matrix between two linguistic ranks. From this matrix, we know that sim^{LL}(*very poor, very poor*) = 1.0, sim^{LL}(*very poor, poor*) = 0.5, sim^{LL}(*poor, poor*) = 1.0, sim^{LL}(*poor, average*) = 0.5, sim^{LL}(*good, good*) = 1.0, ..., sim^{LL}(*very good, very good*) = 1.0.

Table 3
A similarity matrix Sim_{LL} for 5 *linguistic ranks*

	Very poor	*Poor*	*Average*	*Good*	*Very good*
Very poor	1	0.5	0	0	0
Poor	0.5	1	0.5	0	0
Average	0	0.5	1	0.5	0
Good	0	0	0.5	1	0.5
Very good	0	0	0	0.5	1

Definition 10. Assume that we have a *linguistic rank q*-item $a_i = (it_i, f_i)$, a *linguistic q*-item $b_j = (ic_j, f_j)$, and an *LL* similarity matrix Sim_{LL}, where $sim^{LL}\ (f_i, f_j)$ denotes the similarity between f_i and f_j. Then, the degree to which a_i matches b_j, $sup(a_i, b_j)$ can be given as follows:

$$sup(a_i, b_j) = \begin{cases} sim^{LL}(f_i, f_j), & \text{if } it_i = ic_j \\ 0, & \text{otherwise} \end{cases}$$

Example 10. Assume that we have a *linguistic rank q*-item $a_6 = (it_6, good)$, a *linguistic q*-item $b_6 = (ic_6, good)$, and an *LL* similarity matrix Sim_{LL}, as shown in Table 3. Then, the degree $sup(a_6, b_6) = sup((it_6,good), (ic_6, good)) = 1.0$, if $it_6 = ic_6$.

Definition 11. Assume that we have a *multiple-linguistic rank q*-item $a_i = (it_i,\ (f_{i1},\ f_{i2}))$, a *linguistic q*-item $b_j = (ic_j,\ f_j)$, and an*LL* similarity matrix Sim_{LL}. Then, the degree to which a_i matches b_j, $sup(a_i, b_j)$ can be given as follows:

$$sup(a_i, b_j) = \begin{cases} \max(sim^{LL}(f_{i1}, f_j), sim^{LL}(f_{i2}, f_j)), & \textit{if } it_i = ic_j \\ 0, & \textit{otherwise} \end{cases}$$

Example 11. Suppose we have a *multiple-linguistic rank q*-item $a_7 = (it_7,(average,\ good))$, a *linguistic q*-item $b_7 = (ic_7, good)$, and an *LL* similarity matrix Sim_{LL}, as shown in Table 3. If $it_7 = ic_7$, then the degree $sup\ (a_7, b_7) = sup((it_7,(average,\ good)), (ic_7, good))$= max ($sim^{LL}$(*average, good*), sim^{LL}(*good, good*)) = max(0.5, 1) = 1.

Definition 12. A *rule q*-item can be a *category q*-item, *list q*-item, and *linguistic q*-item. For simplicity, we use $b_i = (ic_i, f_i)$ to denote a *rule q*-item. A *rule q*-itemset *B* is a set of *rule q*-items, where all *q*-items' items must be distinct. We use B={(ic_1, f_1), (ic_2, f_2), ..., (ic_n, f_n)} to denote a *rule q*-itemset.

Example 12. For example, {(ic_1, *male*), (ic_2, (BBI, CAI)), (ic_3, *long*), (ic_5, *good*)} is a *rule q*-itemset.

Definition 13. Let a *q*-itemset be a set of *q*-items. Assume that we have a *q*-itemset A={$(it_1, q_1), (it_2, q_2), \ldots, (it_m, q_m)$}, where $a_i = (it_i, q_i)$ could be a *category q*-item, *list q*-item, *number q*-item, *rank q*-item, *multiple-ranks q*-item, *linguistic rank*, and *multiple-linguistic ranks q*-item. Also assume that we have a *rule q*-itemset B={$(ic_1, f_1), (ic_2, f_2), \ldots, (ic_n, f_n)$}, where $b_j = (ic_j, f_j)$ is a *rule q*-item. If we can find $a_{i_1} \leqslant a_{i_2} \leqslant \cdots \leqslant a_{i_n}$ in *A*, such that $\sup(a_{i_j}, b_j) > 0$, then *sup*(A, B) can be defined as follows:

$$sup(A, B) = Aggprod\{sup(a_{i_j}, b_j)\} = \prod_{j=1}^{n} sup(a_{i_j}, b_j)$$

Example 13. Suppose we have a q-itemset A={(it_1, male), (it_2, BBI), (it_3, 4), (it_4, 4), (it_5,(average, good))}, a *rule* q-itemset B = {(ic_1, male), (ic_2, (BBI, CAI)), (ic_3, long), (ic_4, good), (ic_5, good)}, an *RL* similarity matrix representing the similarity between *ranks* and *linguistic ranks*, an *LL* similarity matrix representing the similarity between *linguistic ranks* and the membership function of *long*, as shown in Table 2 and 3, respectively, and expression (3). If $it_i = ic_i$ for $1 \leqslant i \leqslant 7$, then the degree $sup(A,\ B) = 1.0 \times 0.5 \times 0.5 \times 0.86 \times 1.0 = 0.215$.

Definition 14. Assume that we have a database D consisting of a set of transactions, where the *sid*-th transaction in D can be represented as a q-itemset $A_{sid} = \{(it_1,\ q_1),\ (it_2,\ q_2),\ \ldots,\ (it_m,\ q_m)\}$. Let B={(ic_1, f_1), (ic_2, f_2), ..., (ic_n, f_n)} be a *rule* q-itemset. Then, the support of B occurring in D,$sup_D(B)$ can be defined as follows:

$$sup_D(B) = \left(\sum_{sid=1}^{|D|} sup(A_{sid}, B)\right) \Big/ |D|$$

where $|D|$ is the total number of transactions in database D.

Example 14. Suppose we have a database D containing 10 transactions as shown in Table 1 and a rule q-itemset B = ((i_1, male), (i_2, (BBI, CAI)), (i_3, long), (i_4, good), (i_5, good)). Then, the degree $sup_D(B) = (1 \times 1 \times 1 \times 0.86 \times 1 + 1 \times 1 \times 1 \times 0.86 \times 1 + 1 \times 1 \times 1 \times 0.86 \times 1 + 1 \times 0.5 \times 0.5 \times 0.86 \times 1)/10 = (0.86 + 0.86 + 0.86 + 0.215)/10 = 0.2795$.

Definition 15. Given a user-specified threshold σ_s, a rule q-itemset B is frequent if $sup_D(B)$ is no less than σ_s. Let B be a frequent *rule* q-itemset, where $B = X \cup Y$ and $X \cap Y = \phi$. Then, the confidence of rule $X \Rightarrow Y$, denoted as $conf(X \Rightarrow Y)$, is defined as $sup_D(B)/sup_D(X)$. Given a confidence threshold σ_c, if $conf(X \Rightarrow Y) \geqslant \sigma_c$, $X \Rightarrow Y$ holds in database D.

Example 15. According to Definitions 12 and 15, only *category* q-items, *list* q-items, and *linguistic* q-items may appear in both sides of the generated fuzzy rules. Therefore, we may have rules like (i_1, male) → (i_2, (BBI)), (i_2, (BBI, CAI)) → (i_4, good), and (i_3, middle) → (i_5, good). We never, however, generate rules like (i_2, (BBI)) → (i_4, (good, very good)), (i_3, (short, middle)) → (i_5, good), (i_3, long) → (i_5, 5)), or (i_3, 5) → (i_5, good).

3. An algorithm for mining fuzzy association rules from questionnaire data

In this section, we introduce an Apriori-like algorithm, named the CLL algorithm, to discover fuzzy association rules from questionnaire data. The CLL algorithm was developed by modifying the well-known Apriori algorithm [2] to mine *Category*, *List,* and *Linguistic* (CLL) patterns. In Section 3.1, we introduce the CLL algorithm, and in Section 3.2 we use an example to illustrate it.

3.1. The proposed algorithm

We now introduce a new algorithm for mining fuzzy association rules from questionnaire data. The algorithm is outlined in Fig. 3. Although the basic structure of the CLL and the Apriori algorithm are similar, they are different in the following respects:

(1) *Data types:* The Apriori algorithm is designed only for handling nominal/Boolean data. The CLL algorithm, however, is developed for handling the seven data types that may appear in questionnaire data.
(2) *Similarity functions:* In the Apriori algorithm, an item can only have a 100% or 0% match with another item. Therefore, the Apriori algorithm does not need a similarity function to measure the similarity between items. Since partial similarity relationships exist in questionnaire data, however, the CLL algorithm uses the similarity functions described in Section 2 to calculate the similarity between items.
(3) *Counting candidates:* In the Apriori algorithm, an itemset is either completely contained in a transaction or not at all. In the CLL algorithm, however, an itemset can be partly contained in a transaction. As a result, the degree that a transaction contains an itemset is a value between 0 and 1, instead of either 1 or 0.

The proposed algorithm is composed of three phases. In the first phase, we apply the similarity matrixes or membership functions in Section 2 to transform the original database into a new database. After the transformation, a transaction in the new database stores the support of every q-item in the corresponding transaction in the original database. In the second phase, we use a level-wise approach to iteratively generate candidate rule q-itemsets and then find frequent rule q-itemsets. In the final phase, we generate fuzzy

```
Input: A questionnaire database, $D_B$; a rank-linguistic similarity matrix, $Sim_{RL}$; a
linguistic-linguistic similarity matrix, $Sim_{LL}$; membership functions ($FS_{f_j}$), a predefined
minimum support α, and a predefined minimum confidence λ.
Output: A set of fuzzy association rules
Method:
// Phase 1 Call the Sup_Transform Subroutine
(1). For each transaction
        Transform each $q$-item data of the transaction into rule $q$-items;
        Store these results as a new transaction in new database $D^T$.
// Phase 2 Call the LargeItemsets_gen Subroutine
(1). For each rule $q$-item $ic_{j,r}$, calculate its support.
(2). Check whether the support of each rule $q$-item $ic_{j,r}$ is no less than the minimum
     support α. If it is so, put it into the set of large one-itemsets ($L_1$).
(3). Generate candidate set $C_{k+1}$ from $L_k$.
(4). Compute the supports of all rule $q$-itemsets in $C_{k+1}$ and then determine $L_{k+1}$.
(5). If $L_{k+1}$ is null, then do the next step; otherwise, set $k = k + 1$ and repeat steps
     (3)–(4).
// Phase 3 Call the FAR_gen Subroutine
(1). Construct the association rules from all large $q$-itemset $B$.
```

Fig. 3. The CLL algorithm.

association rules from the frequent rule q-itemsets obtained in the second phase. In the following, we present and explain in detail the three subroutines in each phase.

As mentioned above, there are seven data types that exist in questionnaire data. To discover frequent patterns, we need to use the similarity matrixes or membership functions to calculate itemsets' support for different data types. Fig. 4 shows the pseudocode for the *Sup_Transform* subroutine, which is used to calculate its support for each different data type. In steps 1–10, each q-item a_i of each transaction of Database D will match the corresponding data type and calculate its support. Finally, all q-item a_i will be transformed into ($ic_{j,r}$, $\mu_{j,r}$), where $ic_{j,r}$ is a *rule* q-item and $\mu_{j,r}$ is its support in this transaction, and stored in database D^T. Steps 3–9 are used to calculate the support for different data types, such as *categories*, *lists*, *numbers*, *ranks*, *multiple-ranks*, *linguistic ranks*, and *multiple-linguistic ranks*. With the help of similarity functions, the *Sup_Transform* subroutine can transform the original questionnaire dataset into a new form and store it in database D^T. In the next section, we will introduce a new method, *LargeItemsets_gen* subroutine, for mining *Category*, *List,* and *Linguistic* (CLL) patterns from database D^T.

Unlike the Apriori algorithm, which calculates the counts of candidate itemsets by adding either a one or a zero, depending on whether that particular itemset appears in the transaction or not, the *LargeItemsets_gen* subroutine in the proposed algorithm can add a fractional value to the counts of itemsets. Fig. 5 shows the pseudocode for the *LargeItemsets_gen* subroutine. Step 1 finds the frequent 1-itemsets, L_1. In steps 2–10, $L_{k-1}$is used to generate candidates C_k in order to find L_k. The *apriori_gen* subroutine [1,2] generates the candidates and then uses the downward closure property to eliminate those that have a non-frequent subset (step 3). Once all the candidates have been generated, the database is scanned (step 4). For each transaction, a *subset* function is used to find all subsets of the transaction that are candidates (step 5), and the count for each of those candidates is accumulated (steps 6 and 7). Finally, all those candidates satisfying the minimum support form the set of frequent itemsets, L. After generating the frequent patterns, we will use those patterns to generate the *Fuzzy association rules (FAR)*. In the next section, we will introduce a method, named *FAR_gen* subroutine, to generate *Fuzzy association rules (FAR)*from large itemsets *L.*

Finally, we generate the fuzzy association rules from the frequent *rule* q-itemsets obtained in the second phase. Fig. 6 shows the pseudocode for the *FAR_gen* subroutine. Obviously, the procedure can generate all the fuzzy rules, satisfying Definition 15.

3.2. An example

An example is given to illustrate the proposed data mining algorithm. The dataset includes 10 transactions, as shown in Table 1.

STEP 1. Assume that we have an *RL* similarity matrix (Sim_{RL}), an *LL* similarity matrix (Sim_{LL}), and three membership functions (FS_{f_j}) as shown in Table 2, 3, and expressions (1)–(3). For simplicity, the possible values of these five attributes are encoded in Table 4. Based on this encoding scheme, $ic_{1,1}$ means the value of Sex is male and $ic_{2,4}$ means the value of Teaching Style is {(BBI, CAI)}. After the computation, the results of D^T are shown in Table 5.

STEP 2.1. For each *rule* q-item $ic_{j,r}$ stored in database D^T, calculate its support and check whether the support of each *rule* q-item $ic_{j,r}$ is larger than or equal to the minimum support α. If it is, put it in the set of large one-itemsets (L_1). For example, let us set α to 0.5. Then we have L_1, as shown in Table 6.

STEP 2.2. We now generate candidate set C_2 from L_1. For example, we obtain C_2 as follows: ($ic_{1,1}$, $ic_{2,1}$), ($ic_{1,1}$, $ic_{2,2}$), ($ic_{1,1}$, $ic_{2,4}$), ($ic_{1,1}$, $ic_{2,6}$), ($ic_{1,1}$, $ic_{2,7}$), ($ic_{1,1}$, $ic_{3,3}$), ($ic_{1,1}$, $ic_{4,4}$), ($ic_{1,1}$, $ic_{5,3}$), ($ic_{1,1}$, $ic_{5,4}$), ..., and ($ic_{5,3}$, $ic_{5,4}$). After computing their supports, we can determine L_2 as shown in Table 7.

STEP 2.3. Since L_2 is not null, we repeat the previous steps to find L_3, as shown in Table 8. In the next turn, we find C_4 is empty after pruning; therefore, we stop the iterations.

```
Subroutine: Sup_Transform Subroutine. Transform every q-item data ($it_i$, $q_i$) in each
transaction of $D$ into ($ic_{j,r}$, $\mu_{j,r}$), where $ic_{j,r}$ is a rule q-item and $\mu_{j,r}$ is its support in this
transaction.
Input: Questionnaire Database, $D$; a rank-linguistic similarity matrix, $Sim_{RL}$; a
linguistic-linguistic similarity matrix, $Sim_{LL}$; a membership function, $FS_{f_j}$.
Output: $D^T$, a new database where each transaction contains a set of ($ic_{j,r}$, $\mu_{j,r}$).
(1) for each transactions $t \in D$ {
(2)     for each q-item $a_i$=($it_i$, $q_i$){
          // the following procedure do not generate a rule q-item if its support is zero//
(3)       if $a_i \in$ category q-item, then create a rule q-item $b_j$=($it_i$, $q_j$) with support =1;
(4)       if $a_i \in$ list q-item, then for each sublist $f_j$ of the complete list
              create a rule q-item $b_j$=($it_i$, $f_j$) with support as defined in Definition 3.
(5)       if $a_i \in$ number q-item, then for each linguistic term $f_j$
              create a rule q-item $b_j$= ($it_i$, $f_j$) with support as defined in Definition 5.
(6)       if $a_i \in$ rank q-item, then for each linguistic term $f_j$
              create a rule q-item $b_j$= ($it_i$, $f_j$) with support as defined in Definition 7.
(7)       if $a_i \in$ multiple rank, then for each linguistic term $f_j$
              create a rule q-item $b_j$= ($it_i$, $f_j$) with support as defined in Definition 8.
(8)       if $a_i \in$ linguistic rank q-item, then for each linguistic term $f_j$
              create a rule q-item $b_j$= ($it_i$, $f_j$) with support as defined in Definition 10.
(9)       if $a_i \in$ multiple linguistic rank q-item, then for each linguistic term $f_j$
              create a rule q-item $b_j$= ($it_i$, $f_j$) with support as defined in Definition 11.}
(10) store the results as a transaction in the new database $D^T$;}
(11) return $D^T$;
```

Fig. 4. The *Sup_Transforms* function.

```
Subroutine: LargeItemsets_gen Subroutine. Find frequent itemsets using an
iterative level-wise approach based on candidate generation.
Input: Database $D^T$; the minimum support α.
Output: L, frequent itemset in $D^T$.
Method:
    $L_1$ = find_frequent_1-itemsets($D^T$);
    for (k=2; $L_{k-1}$≠∅; k++) {
        $C_k$ = apriori_gen($L_{k-1}$, min_sup);
        for each transactions t∈ $D^T$ { //Scan $D^T$ for counts
            $C_t$=subset($C_k$, t); //get the subsets of t that are candidates;
            for each candidate c ∈ $C_t$
                c.count=c.count + $\prod_{j=1}^{k}\mu_j$ ;
    // where $\prod_{j=1}^{k}\mu_j$ is the support of c in transaction t according to Definition 13
        }
    $L_k$ = {c ∈ $C_t$ | c.count/|$D^T$| ≥ minsup};
    // where |$D^T$| is the total number of transactions in database $D^T$
    return L=$\cup_k L_k$;
```

Fig. 5. The *LargeItemsets_gen* function.

STEP 3. Construct the association rules from all large q-itemsets. We can generate fuzzy rules from L_2 and L_3. For brevity, we only show the rules generated from L_3 in Table 9.

4. Experiment results

We conducted some experiments to evaluate our approach. Survey data concerning teaching evaluations of high school and college courses in Taiwan were used to show the feasibility of the proposed mining algorithm. A total of 383 survey data responses were collected. Each transaction shows information about the teacher's teaching performance and the student's learning performance. The algorithms were implemented using Sun Java language (J2SDK 1.3.1) and tested on a PC with a single Intel Pentium III 866 MHz processor and 512MB main memory running the Windows XP operating system. Neither multi-threading technology nor parallel computing skills were used in our implemented programs.

In the past, almost all of the existing papers in fuzzy mining assumed that the fuzzy functions are given by experts, because this can streamline the presentation of the paper [12,34] and enable us to focus on the design of mining algorithms. Due to the same reasons, in the experiment, we invited a senior faculty in our department to set the values of the six membership functions, matrix Sim_{RL}, and matrix Sim_{LL}, as shown in Section 2.

There are three experiments in this section. In the first experiment, we investigate how the run time of the CLL algorithm changes as we vary the minimum support value and the database size. In the second experiment, we compare the performances of the CLL algorithm and the traditional Apriori algorithm. Since these two algorithms have different data types, we preprocessed the questionnaire dataset so that the comparison could be performed in the same environment. After preprocessing, the two algorithms were tested by varying the minimum support value and the database size. Finally, the third experiment applies the CLL algorithm to discover rules from a real questionnaire dataset. Our algorithm's usefulness is proven through the discovery of some interesting rules that would have never been found using the previous algorithm.

In the first experiment, we wanted to investigate how the run time changes as we vary the minimum support value and the database size. Therefore, we first fixed the database size at 383 and varied the minimum support. In Fig. 7, it is readily apparent that the run time increases with a decrease in minimum support value. This is especially true when the minimum support becomes very small; the run time increases sharply. These results concur with the results from previous association mining algorithms [26,31]. Next, we set the minimum support at 0.5 and varied the number of transactions by repeatedly duplicating the database until the intended size was reached. From Fig. 8, we find that the run time increases linearly with respect to the database size. This linear relationship indicates that the proposed algorithm has a good scalability.

In the second experiment, we wanted to study the performance differences between the Apriori algorithm and the CLL algorithm. Since the only data type that both algorithms can handle is categorical data, we kept the categorical data in the dataset but eliminated the others. Two tests were performed in this experiment. The first test compared the run times of the two algorithms by varying the database size and the second by varying the minimum support.

```
Subroutine: FAR_gen Subroutine. Generate all fuzzy association rules with high
confidence and support.
Input: L, frequent itemset in $D^T$; a predefined minimum confidence λ.
Output: FAR, Fuzzy association rules in $D^T$.
Method:
    For each frequent itemset B=X∪Y, where X∩Y=ϕ
        For every subset X of B
            if the confidence of rule X⇒Y is no less than the minimum confidence λ
                then output the rule
    return FAR
```

Fig. 6. The *FAR_gen* function.

Table 4
Encoded values for the attributes

Encoded values	Sex style	Teaching time	Review skill	Teaching performance	Learning
1	Male	(BBI)	Short	Very poor	Very poor
2	Female	(CAI)	Middle	Poor	Poor
3		(AVI)	Long	Average	Average
4		(BBI, CAI)		Good	Good
5		(BBI, AVI)		Very good	Very good
6		(CAI, AVI)			
7		(BBI, CAI, AVI)			

Table 5
The constructed temporary set D^T

TID	Sex	Teaching style			Review time	Teaching skill	Learning performance
1	$(ic_{1,1}, 1.00)$	$(ic_{2,1}, 1.00)$ $(ic_{2,2}, 1.00)$	$(ic_{2,4}, 1.00)$ $(ic_{2,5}, 0.50)$	$(ic_{2,6}, 0.50)$ $(ic_{2,7}, 0.66)$	$(ic_{3,2}, 1.00)$	$(ic_{4,2}, 0.50)$ $(ic_{4,3}, 1.00)$ $(ic_{4,4}, 0.50)$	$(ic_{5,2}, 0.50)$ $(ic_{5,3}, 1.00)$ $(ic_{5,4}, 0.50)$
2	$(ic_{1,1}, 1.00)$	$(ic_{2,1}, 1.00)$ $(ic_{2,2}, 1.00)$	$(ic_{2,4}, 1.00)$ $(ic_{2,5}, 0.50)$	$(ic_{2,6}, 0.50)$ $(ic_{2,7}, 0.66)$	$(ic_{3,3}, 1.00)$	$(ic_{4,3}, 0.43)$ $(ic_{4,4}, 0.86)$ $(ic_{4,5}, 0.43)$	$(ic_{5,3}, 0.50)$ $(ic_{5,4}, 1.00)$ $(ic_{5,5}, 1.00)$
3	$(ic_{1,2}, 1.00)$	$(ic_{2,1}, 1.00)$ $(ic_{2,2}, 1.00)$	$(ic_{2,4}, 1.00)$ $(ic_{2,5}, 0.50)$	$(ic_{2,6}, 0.50)$ $(ic_{2,7}, 0.66)$	$(ic_{3,1}, 0.50)$ $(ic_{3,2}, 0.50)$	$(ic_{4,1}, 0.43)$ $(ic_{4,2}, 0.86)$ $(ic_{4,3}, 0.43)$	$(ic_{5,1}, 0.50)$ $(ic_{5,2}, 1.00)$ $(ic_{5,3}, 0.50)$
4	$(ic_{1,2}, 1.00)$	$(ic_{2,2}, 1.00)$ $(ic_{2,4}, 0.50)$	$(ic_{2,6}, 0.50)$ $(ic_{2,7}, 0.33)$		$(ic_{3,1}, 0.50)$ $(ic_{3,2}, 0.50)$	$(ic_{4,1}, 0.86)$ $(ic_{4,2}, 0.86)$ $(ic_{4,3}, 0.43)$	$(ic_{5,1}, 0.50)$ $(ic_{5,2}, 1.00)$ $(ic_{5,3}, 0.50)$
5	$(ic_{1,1}, 1.00)$	$(ic_{2,1}, 1.00)$ $(ic_{2,2}, 1.00)$	$(ic_{2,4}, 1.00)$ $(ic_{2,5}, 0.50)$	$(ic_{2,6}, 0.50)$ $(ic_{2,7}, 0.66)$	$(ic_{3,3}, 1.00)$	$(ic_{4,3}, 0.43)$ $(ic_{4,4}, 0.86)$ $(ic_{4,5}, 0.76)$	$(ic_{5,3}, 0.50)$ $(ic_{5,4}, 1.00)$ $(ic_{5,5}, 1.00)$
6	$(ic_{1,1}, 1.00)$	$(ic_{2,1}, 1.00)$ $(ic_{2,2}, 1.00)$	$(ic_{2,4}, 1.00)$ $(ic_{2,5}, 0.50)$	$(ic_{2,6}, 0.50)$ $(ic_{2,7}, 0.66)$	$(ic_{3,3}, 1.00)$	$(ic_{4,3}, 0.43)$ $(ic_{4,4}, 0.86)$ $(ic_{4,5}, 0.43)$	$(ic_{5,3}, 0.50)$ $(ic_{5,4}, 1.00)$ $(ic_{5,5}, 0.50)$
7	$(ic_{1,2}, 1.00)$	$(ic_{2,2}, 1.00)$ $(ic_{2,4}, 0.50)$	$(ic_{2,6}, 0.50)$ $(ic_{2,7}, 0.33)$		$(ic_{3,1}, 0.50)$ $(ic_{3,2}, 0.50)$	$(ic_{4,1}, 0.43)$ $(ic_{4,2}, 0.86)$ $(ic_{4,3}, 0.43)$	$(ic_{5,1}, 0.50)$ $(ic_{5,2}, 1.00)$ $(ic_{5,3}, 0.50)$
8	$(ic_{1,1}, 1.00)$	$(ic_{2,1}, 1.00)$ $(ic_{2,4}, 0.50)$	$(ic_{2,5}, 0.50)$ $(ic_{2,7}, 0.33)$		$(ic_{3,2}, 0.50)$ $(ic_{3,3}, 0.50)$	$(ic_{4,3}, 0.43)$ $(ic_{4,4}, 0.86)$ $(ic_{4,5}, 0.76)$	$(ic_{5,2}, 0.50)$ $(ic_{5,3}, 1.00)$ $(ic_{5,4}, 1.00)$ $(ic_{5,5}, 0.50)$
9	$(ic_{1,2}, 1.00)$	$(ic_{2,1}, 1.00)$ $(ic_{2,2}, 1.00)$ $(ic_{2,3}, 1.00)$	$(ic_{2,4}, 1.00)$ $(ic_{2,5}, 1.00)$	$(ic_{2,6}, 1.00)$ $(ic_{2,7}, 1.00)$	$(ic_{3,3}, 1.00)$	$(ic_{4,3}, 0.43)$ $(ic_{4,4}, 0.86)$ $(ic_{4,5}, 0.43)$	$(ic_{5,3}, 0.50)$ $(ic_{5,4}, 1.00)$ $(ic_{5,5}, 1.00)$
10	$(ic_{1,1}, 1.00)$	$(ic_{2,3}, 1.00)$ $(ic_{2,5}, 0.50)$	$(ic_{2,6}, 0.50)$ $(ic_{2,7}, 0.33)$		$(ic_{3,3}, 1.00)$	$(ic_{4,3}, 0.43)$ $(ic_{4,4}, 0.86)$ $(ic_{4,5}, 0.43)$	$(ic_{5,3}, 0.50)$ $(ic_{5,4}, 1.00)$ $(ic_{5,5}, 0.50)$

Table 6
L_1 large itemsets

Itemsets	Support	Itemsets	Support	Itemsets	Support	Itemsets	Support
$ic_{1,1}$	0.600	$ic_{2,1}$	0.700	$ic_{2,2}$	0.800	$ic_{2,4}$	0.750
$ic_{2,6}$	0.500	$ic_{2,7}$	0.562	$ic_{3,3}$	0.550	$ic_{4,4}$	0.566
$ic_{5,3}$	0.600	$ic_{5,4}$	0.650				

Table 7
L_2 large itemsets

Itemsets	Support	Itemsets	Support	Itemsets	Support	Itemsets	Support
$(ic_{1,1}, ic_{2,1})$	0.500	$(ic_{1,1}, ic_{5,4})$	0.550	$(ic_{2,1}, ic_{2,2})$	0.600	$(ic_{2,1}, ic_{2,4})$	0.650
$(ic_{2,1}, ic_{5,4})$	0.550	$(ic_{2,2}, ic_{2,4})$	0.700	$(ic_{2,4}, ic_{5,4})$	0.500	$(ic_{3,3}, ic_{5,4})$	0.550
$(ic_{4,4}, ic_{5,4})$	0.541						

Table 8
The L_3 large itemsets for this example

Itemsets	Support
$(ic_{2,1}, ic_{2,2}, ic_{2,4})$	0.600
$(ic_{2,1}, ic_{2,4}, ic_{5,4})$	0.500

In the first test, we set the minimum support at 0.5 and varied the number of transactions by repeatedly duplicating the database until the intended size was reached. From the results of Fig. 9, we find that the Apriori algorithm performs slightly better than the proposed algorithm. This result is quite logical because although the two algorithms have a similar structure, the CLL algorithm deals with more complicated data types and uses fuzzy operators to compute supports, which require more complicated computations. In the second test, we set the database size at 100 K and varied the minimum support from 0.01 to 0.5. Fig. 10 indicates that the Apriori algorithm has a better run time than the CLL algorithm. The reason for this result is the same as that stated for the first test. Please note, although the Apriori algorithm is the winner in run time tests, it cannot be used for mining rules from closed questionnaire data, because it can only handle categorical data.

Table 9
The association rules generated from the large three-itemsets

No.	Rule	No.	Rule
1	If ($ic_{2,1}$, $ic_{2,2}$) then $ic_{2,4}$; (confidence = 1.00)	2	If $ic_{2,4}$ then ($ic_{2,1}$, $ic_{2,2}$); (confidence = 0.80)
3	If ($ic_{2,1}$, $ic_{2,4}$) then $ic_{2,2}$; (confidence = 0.92)	4	If $ic_{2,2}$ then ($ic_{2,1}$, $ic_{2,4}$); (confidence = 0.75)
5	If ($ic_{2,2}$, $ic_{2,4}$) then $ic_{2,1}$; (confidence = 0.86)	6	If $ic_{2,1}$ then ($ic_{2,2}$, $ic_{2,4}$); (confidence = 0.86)
7	If ($ic_{2,1}$, $ic_{2,4}$) then $ic_{5,4}$; (confidence = 0.77)	8	If $ic_{5,4}$ then ($ic_{2,1}$, $ic_{2,4}$); (confidence = 0.77)
9	If ($ic_{2,1}$, $ic_{5,4}$) then $ic_{2,4}$; (confidence = 0.91)	10	If $ic_{2,4}$ then ($ic_{2,1}$, $ic_{5,4}$); (confidence = 0.67)
11	If ($ic_{2,4}$, $ic_{5,4}$) then $ic_{2,1}$; (confidence = 1.00)	12	If $ic_{2,1}$ then ($ic_{2,4}$, $ic_{5,4}$); (confidence = 0.71)

The third experiment demonstrated some interesting rules that could actually be mined from the questionnaire dataset by the CLL algorithm. Since traditional algorithms suffer from the limitations that (1) they cannot handle mixed data types, (2) they cannot handle multiple-choice data, and (3) they cannot handle linguistic terms in the raw data, it is inappropriate to use traditional algorithms to mine questionnaire data. Therefore, in this experiment, we only show the rules that were discovered by the CLL algorithm. We also explain why these rules are interesting and why these rules could not be found by previous algorithms.

In this experiment, we set the minimum support at $\alpha = 0.2$ and minimum confidence at $\lambda = 0.7$. Some rules that were found by the CLL algorithm are shown in Table 10. All the rules in the table were derived from raw data involving multiple data types and linguistic terms. For example, Rule #1 indicates that if a student takes a programming course and he or she rates the course's content as average, then the student will consider the lecturer's teaching skill as average with a confidence of 79.19%. This rule cannot be found by traditional algorithms because the raw data contains mixed data types, including categorical data (Course), rank/M-rank data (Content), and Linguistic/M-linguistic data (Teaching skill). Similarly, Rule #4 cannot be discovered by traditional algorithms either, since the raw data contains categorical data (Course), Linguistic/M-linguistic data (Learning performance), and rank/M-rank data (Content).

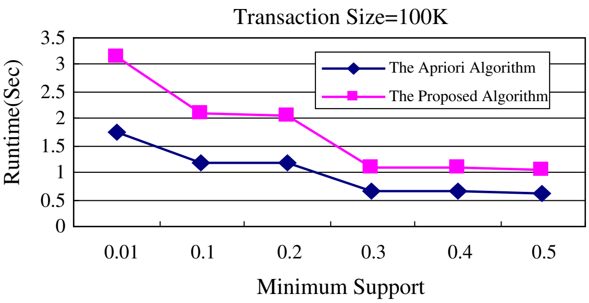

Fig. 10. Run time vs. minimum support.

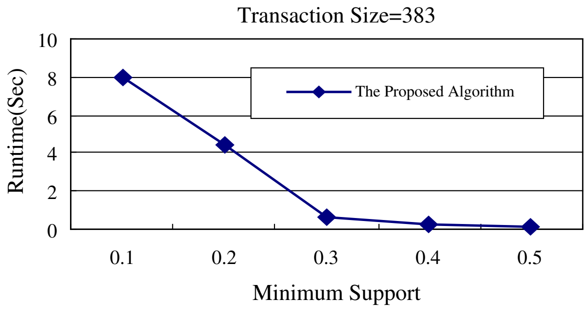

Fig. 7. Run time vs. minimum support.

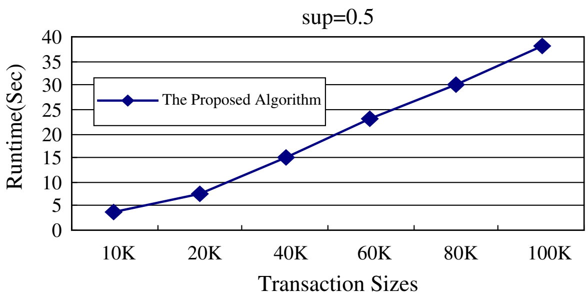

Fig. 8. Run time vs. database size.

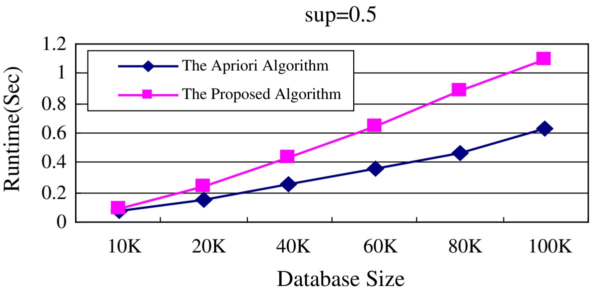

Fig. 9. Run time vs. database size.

Furthermore, the CLL algorithm can also handle a special data type, List, which is composed of multiple-choices in the questionnaire. After setting the minimum support at $\alpha = 0.2$ and minimum confidence at $\lambda = 0.7$, we can see some of the fuzzy association rules discovered by the proposed algorithm in Table 11.

Take Rule #1 and Rule #3, for example. Rule #1 indicates that if blackboard instruction and computer-aided instruction are integrated and the course's content is good, the students' learning performances will be good with a confidence of 74.95%. Rule #3 means that if blackboard instruction and computer-aided instruction are integrated and the lecturer's teaching skill is good, the students' learning performances will be good with a confidence of 74.49%. From the two rules above, we know that if a lecturer prepares good content or has good teaching skills, the students' learning performances will improve. These results show that the proposed CLL algorithm can handle mixed data types, multiple-choice data, and linguistic terms simultaneously. In addition, more interesting rules can be found from the closed questionnaire data set.

5. Conclusion

Association rule mining is one of most popular data mining techniques that can discover relationships between data. Association rule mining algorithms have been applied in various applications and datasets, due to its practical usefulness; however, no association mining algorithms have been used to analyze questionnaire data. This is because previous mining algorithms could not

Table 10
Some rules derived from raw data involving multiple data types and linguistic terms

No.	Rules	Data types in raw data		Support (%)	Confidence (%)
		Antecedent	Consequent		
1	{(Course, Programming), (Content, Average)} → {(Teaching-Skill, Average)}	(categorical, rank/M-rank)	Linguistic/M-linguistic	20.92	79.19
2	{(Course, Programming), (Teaching-Skill, Average)} → {(Content, Average)}	(categorical, linguistic/M-linguistic)	Rank/M-rank	20.92	88.09
3	{(Course, Programming), (Content, Average)} → {(Learning-Performance, Average)}	(categorical, rank/M-rank)	Linguistic/M-linguistic	21.51	81.45
4	{(Course, Programming), (Learning-Performance, Average)} → {(Content, Average)}	(categorical, linguistic/M-linguistic)	Rank/M-rank	21.51	87.31
5	{(Review-Time, Short), (Teaching-Skill, Good)} → {(Learning-Performance, Good)}	(quantitative, linguistic/M-linguistic)	Linguistic/M-linguistic	29.92	70.05
6	{(Review-Time, Short), (Learning-Performance, Good)} → {(Teaching-Skill, Good)}	(quantitative, linguistic/M-linguistic)	Linguistic/M-linguistic	29.92	78.71

Table 11
Some rules containing linguistic terms and multiple-choice lists

No.	Rules	Data type of raw data		Support (%)	Confidence (%)
		Antecedent	Consequent		
1	{(Teaching-Method, (BBI, CAI)), (Content, Good)} → {(Learning-Performance. Good)}	(list, rank/M-rank)	Linguistic/M-linguistic	25.10	74.95
2	{(Teaching-Method, (BBI, CAI)), (Learning-Performance, Good)} → {(Content, Good)}	(list, linguistic/M-linguistic)	Rank/M-rank	25.10	75.29
3	{(Teaching-Method, (BBI, CAI)), (Teaching-Skill, Good)} → {(Learning-Performance, Good)}	(list, linguistic/M-linguistic)	Linguistic/M-linguistic	26.74	74.49
4	{(Teaching-Method, (BBI, CAI)), (Learning-Performance, Good)} → {(Teaching-Skill, Good)}	(list, linguistic/M-linguistic)	Linguistic/M-linguistic	26.74	80.22

handle the mixed data types that may appear in a questionnaire dataset.

This paper has made several contributions. First, we identified the seven data types that may appear simultaneously in a questionnaire. We then introduced the questionnaire data mining problem and proposed useful rule patterns that could be mined from questionnaire data. Second, a unified approach was developed based on fuzzy techniques, so that all different data types could be handled in a uniform manner. To this end, various similarity measures and membership degrees were defined for all seven data types, based on fuzzy techniques. Third, an algorithm was developed to discover fuzzy association rules from a questionnaire dataset. Finally, to evaluate the performance of the proposed algorithm, we compared our algorithm with previous algorithms. The results indicate that our method can find interesting association rules that could never be found with previous mining algorithms.

Although the proposed method works well, there is still much work to be done in this field. First, our method assumes that the membership functions are known in advance. In future research, we will attempt to automatically infer membership functions from the raw data, avoiding the bottleneck during the acquisition of membership functions. Additionally, this work does not consider open questions that may appear in a questionnaire. In the future, we will integrate text mining and association mining techniques to mine questionnaire data, including both open and closed questions. Finally, if a survey is given many times in a particular timeframe, an immediate problem that arises is how to analyze associations among data from mixed data types along the time dimension. In future work, we will attempt to design efficient algorithms to handle this problem.

References

[1] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, in: Proceedings of ACM SIGMOD, Washington, DC, USA, 1993, pp. 207–216.

[2] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proceedings of the VLDB Conference, 1994, pp. 487–499.

[3] M.R. Anderberg, Cluster Analysis for Applications, Academic Press, Orlando, FL, 1973.

[4] D. Arotaritei, S. Mitra, Web mining: a survey in the fuzzy framework, Fuzzy Sets and Systems 148 (1) (2004) 5–19.

[5] S. Auephanwiriyakul, J.M. Keller, A. Adrian, Management questionnaire analysis through a linguistic hard C-means, in: Fuzzy Information Processing Society, NAFIPS, 19th International Conference of the North American, Atlanta, GA, USA, 2000, pp. 402–406.

[6] J.P. Benzecri, Correspondence Analysis Handbook, Mercel Dekker, New York, 1992.

[7] M. Berry, G. Linoff, Data Mining Techniques: For Marketing, Sales, and Customer Support, Wiley, New York, 1997.

[8] S.E. Chang, S.W. Changchien, R.H. Huang, Assessing users' product-specific knowledge for personalization in electronic commerce, Expert Systems with Applications 30 (4) (2006) 682–693.

[9] R. Chau, C.H. Yeh, A multilingual text mining approach to web cross-lingual text retrieval, Knowledge-Based Systems 17 (5–6) (2004) 219–227.

[10] Y.L. Chen, C.H. Weng, Mining association rules from imprecise ordinal data, Fuzzy Sets and Systems 159 (4) (2008) 460–474.

[11] D.W. Cheung, V.T. Ng, A.W. Fu, Y. Fu, Efficient mining of association rules in distributed databases, IEEE Transactions on Knowledge and Data Engineering 8 (6) (1996) 911–922.

[12] A. Conci, E.M.M.M. Castro, Image mining by content, Expert Systems with Applications 23 (4) (2002) 377–383.

[13] M. Delgado, N. Marin, D. Sanchez, M.A. Vila, Fuzzy association rules: general model and applications, IEEE Transactions on Fuzzy Systems 11 (2) (2003) 214–225.

[14] N. Doherty, F. Ellis-Chadwick, C. Hart, An analysis of the factors affecting the adoption of the Internet in the UK retail sector, Journal of Business Research 56 (11) (2003) 887–897.

[15] J.W. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, 2001.

[16] T.P. Hong, K.Y. Lin, S.L. Wang, Fuzzy data mining for interesting generalized association rules, Fuzzy Sets and Systems 138 (2) (2003) 255–269.

[17] T.P. Hong, C.S. Kuo, S.L. Wang, A fuzzy AprioriTid mining algorithm with reduced computational time, Applied Soft Computing 5 (1) (2004) 1–10.

[18] Y.C. Hu, R.S. Chen, G.H. Tzeng, Discovering fuzzy association rules using fuzzy partition methods, Knowledge-Based Systems 16 (3) (2003) 137–147.

[19] J. Hun, Y. Fu, Discovery of multiple-level association rules from large databases, in: Proceedings of the 21st International Conference on Very Large Databases, Zurich, Switzerland, 1995, pp. 420–431.

[20] K. Lagus, S. Kaski, T. Kohonen, Mining massive document collections by the WEBSOM method, Information Sciences 163 (1–3) (2004) 135–156.

[21] Y. Li, S.M. Chung, J.D. Holt, Text document clustering based on frequent word meaning sequences, Data & Knowledge Engineering 64 (1) (2008) 381–404.
[22] W. Lian, D.W. Cheung, S.M. Yiu, An efficient algorithm for finding dense regions for mining quantitative association rules, Computers and Mathematics with Applications 50 (3–4) (2005) 471–490.
[23] G. Marshall, The purpose, design and administration of a questionnaire for data collection, Radiography 11 (2) (2005) 131–136.
[24] W. McKnight, Building business intelligence: text data mining in business intelligence, in: DM Review, 2005, pp. 21–22.
[25] T.W. Miller, Data and Text Mining: A Business Applications Approach, Pearson/Prentice Hall, New Jersey, 2005.
[26] J.S. Park, M.S. Chen, P.S. Yu, An effective hash-based algorithm for mining association rules, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, San Jose, CA, USA, 1995, pp. 175–186.
[27] C. Romero, S. Ventura, Educational data mining: a survey from 1995 to 2005, Expert Systems with Applications 33 (1) (2007) 135–146.
[28] D. Roussinov, J.L. Zhao, Automatic discovery of similarity relationships through Web mining, Decision Support Systems 35 (1) (2003) 149–166.
[29] A. Savasere, E.R. Ommcinskl, S.B. Navathe, An efficient algorithm for mining association rules in large databases, in: Proceedings of the 21st International Conference on Very Large Databases, Zurich, Switzerland, 1995, pp. 432–444.
[30] A. Scharl, C. Bauer, Mining large samples of web-based corpora, Knowledge-Based Systems 17 (5–6) (2004) 229–233.
[31] R. Srikant, R. Agrawal, Mining Quantitative Association Rules in Large Relational Tables, SIGMOD, Montreal, Que., Canada, 1996. pp. 1–12.
[32] R. Srikant, Q. Vu, R. Agrawal, Mining Association Rules with Item Constraints, in: Knowledge Discovery in Databases, 1997, pp. 67–73.
[33] S.S. Weng, Y.J. Lin, A study on searching for similar documents based on multiple concepts and distribution of concepts, Expert Systems with Applications 25 (3) (2003) 355–368.
[34] X. Wu, C. Zhang, S. Zhang, Database classification for multi-database mining, Information Systems 30 (1) (2005) 71–88.
[35] K. Yamanishi, H. Li, Mining open answers in questionnaire data, IEEE Intelligent Systems 17 (5) (2002) 58–63.