Int J Adv Manuf Technol (2012) 62:1–13
DOI 10.1007/s00170-011-3787-3

ORIGINAL ARTICLE

# An association rule mining and maintaining approach in dynamic database for aiding product–service system conceptual design

Xiuli Geng · Xuening Chu · Zaifang Zhang

Received: 21 March 2010 / Accepted: 15 November 2011 / Published online: 15 December 2011


**Abstract** Product–service system (PSS) conceptual design process plays a critical and strategic rule in offering an optimal set of products and services to customers. Parameter translation driven by customer requirements (CRs) is the mainline in this process. A two-phase parameter translating process is presented based on a three-domain PSS conceptual design framework: translating CRs into design requirements (DRs), and then into module characteristics (MCs) of products and services. DRs and MCs of PSS are both divided into product-related part and service-related part. There exist complex relationships between and within the two parts of parameters in one domain and those between parameters in two adjacent domains. This condition increases the complexity and difficulty to translate CRs into DRs and MCs with definite specifications. The traditional parameter translating methods which depend on the designers' experiences are insufficient to fulfill the goal of domain mapping. The wealthy conceptual design cases for existing PSS can be reused to generate sufficient knowledge for designers. An apriori-based association rule mining algorithm is proposed to elicit parameter translating rules for aiding the PSS conceptual design. To ensure the availability of the mined rules, rule redundancy and conflicting solutions are taken into account. Considering the new design data are continuously added in, an incremental updating technique is proposed to maintain the association rules without retracing the original database. A weighting strategy is adopted to highlight the novelty the newly added records, and a competitive strategy is employed to avoid ignoring some promising rules. A case study is carried out to demonstrate the effectiveness of the developed approach for aiding PSS conceptual design.

X. Geng · X. Chu (✉) · Z. Zhang
School of Mechanical Engineering,
Shanghai Jiao Tong University,
Shanghai 200240, People's Republic of China
e-mail: xnchu@sjtu.edu.cn

X. Geng
e-mail: gengxiuli@sjtu.edu.cn

**Keywords** Product–service system (PSS) · Conceptual design · Parameter translating · Association rule mining · Incremental updating

## 1 Introduction

Environment and energy issues have become global concerns attracting attentions of worldwide enterprises. Product–service system (PSS) has emerged as a strategy to promote sustainable production and consumption. PSS strategy can provide higher values to customers not merely depending on material and technology investment. PSS contains a physical product core and non-physical services supplementing product [1]. On one hand, services can substitute some parts of functions of product without abandoning materials. On the other hand, services can ensure value provided by product and create additional value through the entire lifecycle of product.

Most of the early studies on PSS development were primarily conducted from the viewpoint of marketing and management. Manzini and Vezzoli [2] analyzed a strategic design approach to develop PSSs by using some examples. Williams [3] provided a structured overview of some current and planned PSS initiatives at the empirical level in the automotive industry. Recently, various engineering methods and tools have been developed to support the realization of PSSs. Aurich et al. [4, 5] claimed that lifecycle engineering can be used for PSS design and introduced a systematic design process to deliver product-related technical services based upon process modularization. Sakao and Shimomura

[6] and Sakao et al. [7] introduced the concept of service engineering as a new discipline for PSS design, and implemented a computer-aided design tool called Service Explorer. Despite the progresses, these developed methods have not integrated product and services adequately (e.g., they haven't discussed how concurrent design of product and services should be carried out and how influences of product parameters and service parameters should be managed).

In PSS design, since both product and services influence satisfaction of customer, these two aspects should be designed simultaneously. Although product design and service design focus on different aspects, they should be integrated at the very beginning of the conceptual design. The goal to satisfy the customer needs bridges the gap between product design and service design. PSS conceptual design plays an important role in PSS development. According to the general conceptual design process for product or service proposed by Bitran and Pedrosa [8], the specifications of the design requirements (DRs) are identified based on collecting and ranking the customer requirements (CRs), then the specifications of module characteristics (MCs) are identified, and the proper concept of PSS are determined finally. The concept of domain is borrowed from the axiomatic design methodology [9] to form the framework of PSS conceptual design. There are three parameter domains in this framework: customer domain, functional domain, and product–service domain. The relationships between CRs in the customer domain and DRs in the functional domain and which between DRs and MCs in the product–service domain can be modeled by the mapping relationships. The main critical issue in domain mapping is how to determine appropriate DRs and MCs after obtaining a set of CRs in the customer domain.

Quality function deployment (QFD) known as a product planning tool can facilitate the understanding and response to the requirements of customer. It can translate CRs into DRs, and then to part characteristics and other requirements in product design [10]. In general, the parameter translating relationships in QFD are always estimated by designers subjectively based on their experiences. Linguistic data and incomplete data were used in expressing the relationship evaluations in order to cope with the vagueness and uncertainty in the decision-making process [11, 12]. Fung et al. [13] and Harding et al. [14] proposed fuzzy inference model to derive fuzzy rules based on historical design knowledge and experiences, and determined design specifications satisfying customer requirements based on fuzzy rules. This approach can improve the objectivity of the decision process. Although regression or fuzzy regression approaches developed to identify the quantitative translating relationships can determine specific values of the output parameters based on mathematical programming model [15–17], the output parameters must be identified by designers first as a prerequisite. The neural network (NN) approach was proposed to generate translating relationships from example data which can learn complicated relationships [18]. However, the relationships are still in "black boxes" and not tangible. Moreover, the historical data required in NN approach must be complete to cover all input and output parameters. Jiao and Zhang [19] pointed out that association rule mining can be applied to discover useful patterns associated with requirement analysis enacted among customers, marketing folks, and designers, and proposed an association rule mining system for effective product portfolio identification. Shao et al. [20] applied association rule mining approach to discover association rules between product specifications and configuration alternatives.

In PSS conceptual design, both product and services are means to meet customer needs. DRs of product (P-DRs) and DRs of services (S-DRs) are two separate but interrelated sets of descriptions and the inter-dependency relationships between and within CRs and DRs in PSS design are much more complicated than those in simple product or service design. MCs of product (P-MCs) and MCs of services (S-MCs) are also two separate but interrelated sets of descriptions, and complicated relationships exist between and within DRs and MCs. Therefore, subjective decisions are hard to be made for determining the specifications of P-DRs and S-DRs and those of P-MCs and S-MCs after obtaining a set of CRs. Moreover, the specifications of DRs and MCs can be expressed in linguistic variables or exact values, and this condition brings difficulty to choose appropriate approaches to determine them.

Knowledge discovery and data mining are becoming important research topics and have been successfully applied in engineering field. Data in development departments are accumulated design knowledge and experiences. Accessing the valuable knowledge hidden in the vast amount of data is important and necessary, and data mining-based approaches can provide attractive solutions and methods for difficult information-related engineering problems [21]. Sadoyan et al. [22] presented a data mining algorithm based on the rough sets theory to extract "if/then" decision rules for manufacturing process control. Alisantoso et al. [23] constructed an information system based on rough set theory to derive design rules for design concept analysis at early stage. Yin et al. [24] developed a knowledge discovery technology based on data mining algorithm to deal with the finite element analysis data, which combining fuzzy set theory and rough set theory. Discovering association rules is an important data mining problem, the algorithm of which is to determine the relationships between items that occur synchronously in a record. The wealthy historical design data for existing PSS can be reused for supporting current design. The issue of determining mapping relationships between CRs and

DRs and that between DRs and MCs in the PSS conceptual design framework can be deemed as a typical multi-dimension association rule mining problem (i.e., the front-item and back-item are both multiple). Apriori algorithm developed by Agrawal and Srikant [25] is a classical and effective approach to extract association rules from a predefined database. However, the PSS conceptual design database in the real world is dynamic and new design data are unceasing added in. The extracted rule base has to be maintained.

Addition of new records in a database creates an opportunity to obtain new useful association rules, but it is a tough work to maintain the association rules. One possible approach is to re-run the association rule mining algorithm on the whole updated database [26]. However, this approach has some obvious disadvantages, e.g., all the formal computations done are wasted, and these cost a lot of time and resources. The incremental updating technique is a way to solve the issue of record addition in a dynamic database mined process without re-implementing the algorithm on the original database.

This paper proposes an association rule mining and maintaining approach in dynamic database for aiding PSS conceptual design. There are two tasks in this research: (1) extracting two classes of multi-dimensional association rules from two different predefined databases, respectively: translating relationships from CRs to DRs (i.e., P-DRs and S-DRs) and those from DRs to MCs (i.e., P-MCs and S-MCs), (2) maintaining the rule bases after new data are added in the databases. Corresponding to the two tasks, an association rule mining algorithm based on the apriori combining with an incremental updating technique is developed in the approach.

The rest of this paper is organized as follows. Section 2 introduces the framework of PSS conceptual design and formulates the problem of mining and maintaining parameter translating rules from historical data. Section 3 presents the detailed association rule mining algorithm based on apriori for discovering association rules. Section 4 provides the detailed incremental updating technique for maintaining the rule base. Section 5 uses an application to illustrate the effectiveness of the developed approach for aiding PSS conceptual design. Section 6 presents the conclusions achieved in this paper.

## 2 Literature review and problem formulation

The framework of PSS conceptual design contains three domains and two parameter translating processes between adjacent domains. The CRs in the customer domain are identified first through interaction between customers and designers. The main task of designers in the conceptual design process is to identify specifications of DRs (i.e., P-DRs and S-DRs) and MCs (i.e., P-MCs and S-MCs) in the last two domains of the framework, and then adopt suitable optimizing approaches to generate an optimal PSS concept. Identifying specifications of DRs and MCs is based on the extracted information on translating relationships from CRs to DRs and those from DRs to MCs. Suppose databases 1 and 2 store the historical PSS conceptual design records about CRs and DRs and those about DRs and MCs, respectively. Suppose the association rule bases 1 and 2 store the hidden translating relationships between CRs and DRs extracted from database 1 and those between DRs and MCs extracted from database 2, respectively, the two association rule bases can be used to support the mapping process between two adjacent domains in the PSS conceptual design together with expert decision system, as shown in Fig. 1.

Apriori algorithm is commonly used to discover association rules for its simplicity and efficiency. Its implementing process can be divided into two steps, briefly: (1) generate all frequent itemsets that satisfy the minimum support threshold and (2) generate all association rules that satisfy the minimum confidence threshold from the frequent itemsets. Many variants of apriori algorithm have been proposed according to different data and environments [27–29]. However, these researches are limited to discover association rules on statistic database. In general, there exist three updating variants of the original database: inserting new records, modifying some records, and deleting some records. However, in most cases, the historical data are unlikely to be modified or deleted once they have been stored. Therefore, the common case to be studied is the incremental updating problem of data mining. Cheung et al. [30; 31] proposed an incremental mining algorithm, called FUP (Fast UPdate algorithm) for incrementally maintaining mined association rules due to record insertion. The FUP algorithm is a modification of the apriori mining algorithm. It evaluates newly inserted records first, and determines whether or not re-scanning the original database is needed. Although the FUP algorithm can indeed improve mining performance for incrementally growing database, original database still need to be scanned when necessary. Lee et al.[32]proposed the DELI (Difference Estimation for Large Itemsets) algorithm to deal with the incremental updating problem of discovering association rules, which used sampling techniques to evaluate whether or not the transaction database changed significantly enough to start updating of the rule set. Dudek [33] proposed a new approximate algorithm RMAIN (Rule MAINtenance) for incremental maintenance of association rules without re-running through the processed data, which worked repeatedly on subsequent portions of new transactions. However, these algorithms do not consider the fact that recently inserted

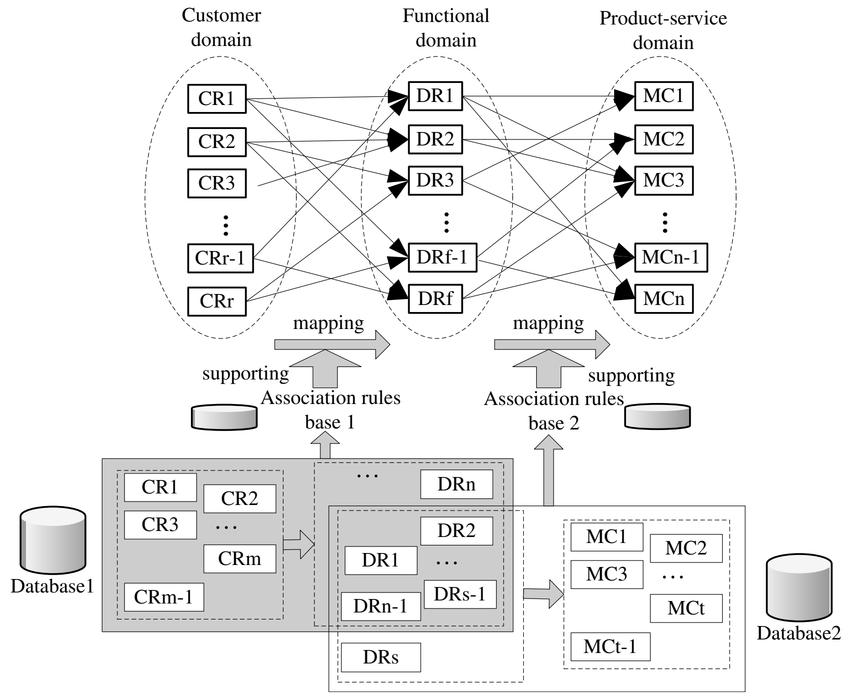


**Fig. 1** The supporting role of association rules in the framework of PSS conceptual design

records may be more interesting than those inserted long ago for that they reflect more closely the current trends. It is not appropriate to treat each record as of equal importance. Zhang et al. [34] proposed a weighting strategy for maintaining association rules in dynamic database. The recently added records are given higher weights, and this strategy can reflect the novelty of newly inserted records.

Most of the incremental updating techniques based on apriori only reuse the information of frequent itemsets derived from the original database, and the promising infrequent itemsets is neglected. However, some infrequent itemsets may become frequent itemsets and be extracted as association rules with addition of new records. Zhang et al. [34] developed a competitive model in the association rules maintaining process to "promote" infrequent itemsets to become frequent itemsets in order to extract effective rules as much as possible. However, this approach ignored dealing with rule redundancy and conflicting.

A new association rule mining and maintaining approach in dynamic database is proposed in this paper. Two parts of this approach are an apriori-based algorithm to mine association rules and an incremental updating technique to maintain the rules. The incremental updating technique adopts the weighting strategy and competitive strategy proposed by Zhang et al. [34]. According to the requirements of the incremental updating technique, the apriori-based approach keeps the promising infrequent itemsets. Rule redundancy and conflicting are taken into account in the approach in order to ensure the availability of the association rules.

## 3 An apriori-based association rule mining algorithm

### 3.1 Basic notions

Suppose $I_1=\{\mathrm{CR}_1, \mathrm{CR}_2, \ldots, \mathrm{CR}_m, \mathrm{DR}_1, \mathrm{DR}_2, \ldots, \mathrm{DR}_n\}$ represent the general item set for database 1. Specification of each item in $I_1$ has many options. $T_1 \subseteq I_1$ represents a record in the database, where $T_1=T_{F1}\cup T_{B1}$, $T_{F1} \subseteq \{\mathrm{CR}_1, \mathrm{CR}_2, \ldots, \mathrm{CR}_m\}$; $T_{B1} \subseteq \{\mathrm{DR}_1, \mathrm{DR}_2, \ldots, \mathrm{DR}_n\}$. Each item in $T_1$ has one exact specification. Suppose $I_2=\{\mathrm{DR}_1, \mathrm{DR}_2, \ldots, \mathrm{DR}_s, \mathrm{MC}_1, \mathrm{MC}_2, \ldots, \mathrm{MC}_t\}$ represent the general item set for database 2. $T_2 \subseteq I_2$ represents a record in the database, where $T_2=T_{F2}\cup T_{B2}$, $T_{F2} \subseteq \{\mathrm{DR}_1, \mathrm{DR}_2, \ldots, \mathrm{DR}_s\}$; $T_{B2} \subseteq \{\mathrm{MC}_1, \mathrm{MC}_2, \ldots, \mathrm{MC}_t\}$. Since the problems of mining association rules in the two classes of databases are the same. A unified record representation in different databases is given. Set $T\sim(F_g, B_g)$ represents the record in a database, $g=1,2,\ldots,N$, where $N$ is the number of records in the database. $F_g$ and $B_g$ ($g=1, 2,\ldots,N$) are the front-item set and the back-item set of $T$, respectively. The mined association rules are represented in the general form: $X \Rightarrow Y$ (i.e., IF $X$ THEN $Y$), where $X \subseteq \cup_g F_g, Y \subseteq \cup_g B_g, g = 1, 2, \ldots, N$. Two measure standards of holding $X \Rightarrow Y$ in association rule mining are *support* and *confidence*.

(1) *Support* is used to evaluate the statistical importance of the association rule.

$$\mathrm{supp}(X \Rightarrow Y) = \mathrm{supp}(X \cup Y) = \frac{|T(X \cup Y)|}{|T|}, \tag{1}$$

where $|T(X \cup Y)|$ is the number of records in database $T$ containing item set $X \cup Y$, $|T|$ is the total number of records in database $T$.

(2) *Confidence* is used to evaluate the level of confidence about the association rule.

$$\mathrm{conf}(X \Rightarrow Y) = \frac{|T(X \cup Y)|}{|T(X)|}, \tag{2}$$

where $|T(X)|$ is the number of records in database $T$ containing item set $X$.

Given a database $D$, the problem of mining association rules is to generate all rules that have values of support equal or larger than the user defined minimum support (Minsupp) threshold and values of confidence equal or larger than user defined minimum confidence (Minconf) threshold.

Some basic definitions of the typical apriori algorithm are given as follows.

Definition 1. $k$-Itemset. If the number of items of an itemset is $k$, the itemset is a $k$-itemset.

Definition 2. Frequent itemset. For an itemset, if the occurrence frequencies of all items are all no less than the predefined Minsupp for the database, the itemset is a frequent itemset.

Definition 3. Apriori-gen function. For a set $L_{k-1}$ containing all frequent $(k-1)$-itemsets, the apriori-gen $(L_{k-1})$ has two steps: join and prune. The join step returns a set of $k$-itemsets which are generated by joining all the $(k-1)$-itemsets. The prune step deletes some $k$-itemsets, at least one $(k-1)$-subsets of which do not belong to $L_{k-1}$. At last, the apriori-gen $(L_{k-1})$ returns the set of candidate $k$-itemsets: $C_k$.

Definition 4. Strong association rule. For a rule $r$, if supp($r$)≥ Minsupp and conf($r$)≥Minconf (Minsupp and Minconf are predefined), the rule is considered as a strong association rule.

Generally, when the association rules are discovered, some redundant or conflicting rules may exist in the rule base. These rules must be processed. Some definitions for dealing with rule redundancy and rule conflicting are given as follows.

Definition 5. Rule preceding. For rule $r_1$ and $r_2$, if conf($r_1$)> conf($r_2$),or conf($r_1$)=conf($r_2$) and supp($r_1$)> supp($r_2$), $r_1$ is called to precede $r_2$.

Definition 6. Rule redundancy. For two strong association rules $r_1(X_1 \Rightarrow Y_1)$ and $r_2$ $(X_2 \Rightarrow Y_2)$, if $X_1 \subseteq X_2$, and $Y_2 \subseteq Y_1$, $r_2$ is a redundant rule.

Definition 7. Rule conflicting. For two strong association rules $r_1(X_1 \Rightarrow Y_1)$ and $r_2$ $(X_2 \Rightarrow Y_2)$,$Y_1 \neq Y_2$,if $X_1=X_2$ or $X_1 \supseteq X_2$ and $r_1$ does not precede $r_2$, $r_1$ conflicts with $r_2$.

### 3.2 Procedure of the association rule mining algorithm

The typical apriori algorithm focuses on the frequent itemsets only. With the addition of incremental database, not only the frequent itemsets may become infrequent itemsets, but also the infrequent itemsets have the potential to become frequent itemsets. Considering the problem of incremental updating, the potential promising infrequent itemsets need to be output in the association rule mining process. The competitive set (CS) is used to store all promising infrequent items, the support values of which must be no less than a threshold called Mincruc which is predefined.

Given a database $D$, the threshold Minsupp and Minconf are set first. The procedure of the apriori-based association rule mining algorithm is as follows.

Step 1 Find all frequent itemsets.

(a) Generate a candidate two-itemset $C_2 = \{(X_2, Y_2) \mid X_2 \in \cup_g F_g, Y_2 \in \cup_g B_g\}$;

(b) $L_2 = \{c_2 \in C_2 \mid \mathrm{supp}(c_2) \geq \mathrm{Minsupp}\}$ and $CS_2 = \{c_2 \in C_2 \mid \mathrm{Minsupp} > \mathrm{supp}(c_2) \geq \mathrm{Mincruc}\}$;//Scan $T$ once to generate frequent two-itemsets and competitive two-itemsets.

(c) For $(k = 3; L_{k-1} \neq \emptyset; k++)$

{$C_k$=Apriori-gen ($L_{k-1}$);

For each record $t \in T$

{$C_t$=subset ($C_k$,$t$);// $C_t$ is a set containing *records* which contain subsets of $C_k$.

For each itemset $c_k$=( $X_k$, $Y_k$ ) , $c_k \in C_t$;

*Count* ( $c_k$)++;}

$L_k$={$c_k \in C_k$|$supp(c_k) \geq Minsupp$} and $CS_k$={$c_k \in C_k$|$Minsupp > supp(c_k) \geq Mincruc$ } ;}

(d) Return $L=\{L_2, L_3,\ldots,L_k\}$ and $CS=\{CS_2,CS_3,\ldots, CS_k\}$.

Step 2 Generate the strong association rule set $R$.

For each frequent itemset $(X, Y)$ in $L$, if conf($(X, Y)$)≥Minconf, the rule $X \Rightarrow Y$ is extracted as a strong association rule. All the strong association rules constitute $R$.

Step 3 Check and delete the redundant rules in $R$.

Step 4 Check and deal with the conflicting rules in $R$.

Check each rule $r_1(X_1 \Rightarrow Y_1)$ in $R$, if there exists a rule $r_2(X_2 \Rightarrow Y_2)$, where $X_1=X_2$, $Y_1 \neq Y_2$. If $r_2$ does not precede $r_1$, then delete $r_2$. If there exists a rule $r_2(X_2 \Rightarrow Y_2)$,where$X_1 \supseteq X_2$, $Y_1 \neq Y_2$.If $r_1$ does not precede $r_2$, then delete $r_1$.

## 4 Incremental updating technique

When new records are inserted in the original database and they constitute an incremental database, the values of support and confidence of association rules in the original database are going to change. Those values of association rules mined in the incremental database are also going to change. In order to highlight the novelty of the new records to mine fashionable rules, a weighting strategy is used to integrate the values of support and confidence of the same rule from the two different databases. Assigning weights to the two databases considers not only the size, but also the novelty of the database.

Let $w_1$ and $w_2$ be the weights of the original database and the incremental database, respectively. For any association rule $X \Rightarrow Y$, supp($X \cup Y$) and supp($X^+ \cup Y^+$) are the support of $X \Rightarrow Y$ in the original database $D$ and the incremental database $D^+$ to $D$, respectively. The weighted $\text{supp}_w(X \cup Y)$ in $D \cup D^+$ is defined as:

$$\text{supp}_w(X \cup Y) = w_1 \times \text{supp}(X \cup Y) + w_2 \times \text{supp}(X^+ \cup Y^+) \tag{3}$$

Suppose the Minsupp and Minconf thresholds do not change. The problem of incremental updating of association rule mining is described as follows.

Input: $D$: the original database; $D^+$: the incremental database; $R$: the association rule base extracted from $D$; $CS$: the competitive itemsets set extracted from $D$;
Given: $w_1$: the assigned weight to $D$; $w_2$: the assigned weight to $D^+$; Minsupp, Minconf, and Mincruc: threshold values;
Output: $R_u$: the updated association rule base extracted from $D \cup D^+$; $CS_u$: the updated competitive itemsets set extracted from $D \cup D^+$.

The procedure of the incremental updating approach is as follows.

Step 1 Mine rules from $D^+$ to obtain $R^+$.

Step 2 Re-examine the rules in $R$.

```
Let temp = ∅ ;//temp is a set to store the infrequent itemsets generated in the updating process.
For any ( X ⇒ Y ) ∈ R   do
begin
  let supp(X∪Y)=w1* supp(X∪Y)+ w2* supp(X+∪Y+);
  let conf(X ⇒ Y) = w1 * conf(X ⇒ Y) + w2 * conf(X+ ⇒ Y+);
  if supp(X∪Y)≥Minsupp and  conf(X ⇒ Y) ≥Minconf
  then add  X ⇒ Y into R'  ;
  else  temp = temp ∪ {X ∪ Y} ;
end;
```

Step 3 Examine the competitive itemsets in *CS*.

For any $B \in CS$ , $B = (F_b, B_b)$ satisfying $F_b \subset \cup_g F_g$ and $B_b \subset \cup_g B_g$ do

begin

let $supp(B) = w_1 * supp(B) + w_2 * supp(B^+)$ ;

if $supp(B) \geq Minsupp$ then

for any $A$ satisfying $A \subset B$ , $A \subset \cup_g F_g$ ,and $(B - A) \subset \cup_g B_g$ do

begin

let $supp(A) = w_1 * supp(A) + w_2 * supp(A^+)$ ;

$conf(A \Rightarrow (B - A)) = supp(B)/supp(A)$ ;

if $conf(A \Rightarrow (B - A)) \geq Minconf$ then

begin

let add $A \Rightarrow (B - A)$ into $R'$ ;

end;

else let $temp = temp \cup \{A, B - A\}$

end;

end;

Step 4 Re-examine the rules in $R^+$.

For any ( $X_1^+ \Rightarrow Y_1^+$ ) $\in R^+$ do

begin

let $supp(X_1^+ \Rightarrow Y_1^+) = w_2 * supp(X_1^+ \Rightarrow Y_1^+) + w_1 * supp(X_1 \Rightarrow Y_1)$ ;

let $conf(X_1^+ \Rightarrow Y_1^+) = w_2 * conf(X_1^+ \Rightarrow Y_1^+) + w_1 * conf(X_1 \Rightarrow Y_1)$ ;

if $supp(X_1^+ \Rightarrow Y_1^+) \geq Minsupp$ and $conf(X^+ \Rightarrow Y^+) \geq Minconf$

then add $X_1^+ \Rightarrow Y_1^+$ into $R_1'$ ;

end;

Step 5 Output $R_u$.

Check and delete the redundant rules and then check and deal with the conflicting rules in $R' \cup \mathrm{R}'_1$. $R_u = R' \cup \mathrm{R}'_1$.

Step 6 Output new *CS*.

For any $T \in temp$ do

begin

if $supp(T) \geq Mincruc$ then

add *T* into *CS*;

end;

**Table 1** Customer requirements

| No. | Customer requirements | Options |
|---|---|---|
| $CR_1$ | Flow accuracy | VL, L, ML, M, MH, H, VH |
| $CR_2$ | Operating reliability | VL, L, ML, M, MH, H, VH |
| $CR_3$ | Broken time | VL, L, ML, M, MH, H, VH |
| $CR_4$ | Performance/cost | VL, L, ML, M, MH, H, VH |

**Table 2** Design requirements

| No. | Design requirements | Options |
|---|---|---|
| $DR_1$ | Flow adjustment level | 30:1, 50:1, 80;1 |
| $DR_2$ | Error of flow, % | 0.3, 0.5, 0.8 |
| $DR_3$ | Self-monitoring ability | L, M, H |
| $DR_4$ | Component reliability | ML, M, MH |
| $DR_5$ | Service response time, h | 12, 24, 48 |
| $DR_6$ | Service completing time, h | 1, 3, 6 |
| $DR_7$ | Reliability of preventive maintenance (PM) | L, M, H |
| $DR_8$ | Technical supporting level | L, M, H |

## 5 Case study

As one of the world's top 500 companies, company H manufactures world-class metering pumps and provides services to their pump products in the Asia Pacific region. The main products of its Chinese subsidiary are the reciprocating metering pumps and the leak-proof centrifugal pumps, which are widely used in waste water treatment, petrochemical and chemical industry, pharmaceutical and food industry, etc. Although a reciprocating metering pump is not used alone, it serves as the key module in the whole system. The function of a metering pump is to dose chemical medicament into the system at specific time and with specific quantity. Its operating stability and robustness have direct impact on the quality of the system output such as the composition of the treated waste water.

Waste water treatment is important in municipal management. Among various water treatment methods, the chemical treatment method is the most popular one. Dosing chemical medicament is a critical task in the process of chemical treatment method. The reciprocating metering pump is used to treat the waste water by dosing polymer into it. Customers usually have high requirements for the operating stability and reliability of the metering pump. Customers pay attention not only to the dosing function of the machine, but also the maintenance of the required function in its lifecycle. The PSS conceptual design for the metering pump needs to consider both the pump and its related technical services to satisfy the individual customer requirement and enhance the company's competitiveness.

Identification of customer requirements is the beginning of the PSS conceptual design. The identified CRs in the customer domain need to be translated to design parameters in the functional domain and the product–service domain. The parameter translating process is the mainline of the PSS conceptual design, which is supported by the translating rules extracted from the dynamic historical design databases for existing PSS. Since the process of discovering association rules reflecting the mapping relationships between CRs and DRs is similar to the process of discovering those reflecting the mapping relationships between DRs and MCs, only the mining and maintaining process of the association rules about CRs and DRs in dynamic database is shown in the example to demonstrate the proposed approach in this paper.

**Table 3** Design data of relationships between customer requirements and design requirements

| No. | $CR_1$ | $CR_2$ | $CR_3$ | $CR_4$ | $DR_1$ | $DR_2$ | $DR_3$ | $DR_4$ | $DR_5$ | $DR_6$ | $DR_7$ | $DR_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | MH | M | L | M | 30 | 0.5 | L | MH | 24 | 3 | M | M |
| 2 | M | H | ML | M | 50 | 0.8 | M | M | 24 | 1 | H | M |
| 3 | MH | MH | M | MH | 30 | 0.5 | M | MH | 24 | 3 | M | H |
| 4 | MH | MH | L | MH | 30 | 0.5 | H | MH | 12 | 3 | M | H |
| 5 | MH | H | VL | H | 80 | 0.5 | H | MH | 12 | 1 | H | H |
| 6 | MH | VH | ML | H | 80 | 0.3 | H | MH | 24 | 1 | H | H |
| 7 | H | VH | M | MH | 50 | 0.3 | M | MH | 24 | 1 | H | H |
| 8 | H | MH | M | VH | 50 | 0.3 | L | M | 24 | 3 | M | M |
| 9 | H | MH | ML | H | 50 | 0.3 | L | MH | 24 | 3 | M | H |
| 10 | H | VH | M | MH | 50 | 0.3 | H | H | 24 | 3 | H | L |
| 11 | H | VH | ML | H | 50 | 0.3 | M | M | 24 | 1 | H | M |
| 12 | H | VH | VL | MH | 30 | 0.5 | H | MH | 12 | 1 | H | H |
| 13 | VH | VH | ML | MH | 80 | 0.3 | M | MH | 24 | 1 | H | H |
| 14 | VH | H | ML | H | 80 | 0.5 | M | MH | 12 | 3 | M | M |
| 15 | VH | MH | ML | H | 80 | 0.3 | H | MH | 12 | 3 | M | H |

**Table 4** The strong association rules between CRs and DRs

| No. | IF (CRs) | THEN (DRs) | Support | Confidence |
|---|---|---|---|---|
| 1 | Flow accuracy=H | Error of flow=0.3 | 0.33 | 0.83 |
| 2 | Flow accuracy=H | Service response time=24 | 0.33 | 0.83 |
| 3 | Flow accuracy=H | Flow adjustment level=50:1 | 0.33 | 0.83 |
| 4 | Flow accuracy=H | Error of flow=0.3<br>Flow adjustment level=50:1 | 0.33 | 0.83 |
| 5 | Flow accuracy=H | Error of flow=0.3<br>Service response time=24 | 0.33 | 0.83 |
| 6 | Flow accuracy=H | Service response time=24<br>Flow adjustment level=50:1 | 0.33 | 0.83 |
| 7 | Flow accuracy=H | Flow adjustment level=50:1<br>Error of flow=0.3<br>Service response time=24 | 0.33 | 0.83 |
| 8 | Flow accuracy=MH | Component reliability=MH | 0.33 | 0.33 |
| 9 | Operating reliability=MH | Service completion time=3 | 0.33 | 1 |
| 10 | Operating reliability=MH | Reliability of PM=M | 0.33 | 1 |
| 11 | Broken time=ML | Error of flow=0.3 | 0.33 | 0.71 |
| 12 | Broken time=ML | Component reliability=MH | 0.33 | 0.71 |
| 13 | Broken time=ML | Error of flow=0.3<br>Component reliability=MH | 0.33 | 0.71 |
| 14 | Broken time=ML | Service response time=24 | 0.33 | 0.71 |
| 15 | Broken time=ML | Component reliability=MH<br>Service response time=24 | 0.33 | 0.71 |
| 16 | Performance/cost=H | Component reliability=MH | 0.33 | 0.83 |

Four CRs and eight DRs are involved in the objective database as shown in Tables 1 and 2, the specifications of which have many options. In order to capture the imprecision and uncertain in the decision-making process, linguistic variables are used to express the options of CRs' specifications and some DR's specifications. The corresponding linguistic variables are very low (VL), low (L), medium low (ML), medium (M), medium high (MH), high (H), and very high (VH).There are 15 records gathered in the original database as shown in Table 3. The Minsupp and Minconf

**Table 5** The infrequent itemsets stored in the competitive set

| No. | CRs | DRs | Support |
|---|---|---|---|
| 1 | Flow accuracy=H | Component reliability=MH | 0.26 |
| 2 | Flow accuracy=H | Reliability of PM=H | 0.26 |
| 3 | Flow accuracy=H | Technical supporting level=H | 0.26 |
| 4 | Flow accuracy=MH | Error of flow=0.5 | 0.26 |
| 5 | Flow accuracy=MH | Technical supporting level=H | 0.26 |
| 6 | Operating reliability=MH | Component reliability=MH | 0.26 |
| 7 | Operating reliability=MH | Technical supporting level=H | 0.26 |
| 8 | Broken time=ML | Service completion time=1 | 0.26 |
| 9 | Broken time=ML | Technical supporting level=H | 0.26 |
| 10 | Broken time=ML | Reliability of PM=H | 0.26 |
| 11 | Broken time=ML | Flow adjustment level=80:1 | 0.26 |
| 12 | Broken time=ML | Self-monitoring ability=M | 0.26 |
| 13 | Performance/cost=H | Error of flow=0.3 | 0.26 |
| 14 | Performance/cost=H | Service response time=24 | 0.26 |

**Table 6** The new-added design data of relationships between customer requirements and design requirements

| No. | $CR_1$ | $CR_2$ | $CR_3$ | $CR_4$ | $DR_1$ | $DR_2$ | $DR_3$ | $DR_4$ | $DR_5$ | $DR_6$ | $DR_7$ | $DR_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | H | H | ML | H | 50 | 0.3 | M | MH | 24 | 3 | H | H |
| 2 | H | MH | M | MH | 30 | 0.3 | L | MH | 12 | 3 | M | H |
| 3 | H | H | ML | H | 30 | 0.3 | H | MH | 24 | 1 | H | H |
| 4 | MH | MH | ML | MH | 80 | 0.5 | M | MH | 12 | 3 | M | M |
| 5 | MH | M | M | MH | 50 | 0.5 | H | M | 24 | 6 | L | M |
| 6 | MH | MH | ML | H | 50 | 0.5 | L | MH | 48 | 3 | M | H |
| 7 | M | M | M | H | 80 | 0.3 | M | M | 24 | 6 | M | M |

thresholds are set as 0.3 and 0.7, respectively. By implementing the proposed mining approach, the frequent itemsets with the confidence values no less than 0.7 are extracted as strong association rules, as shown in Table 4. The strong association rule base has to be checked and reduced according to processing procedures of the rule redundancy and rule conflicting. Rules 1–6, 11, 12, and 14 are deleted from the strong association rule base because they are redundant rules. The remaining association rules are output to support the mapping process between the first two domains of PSS conceptual design. The infrequent itemsets stored in the CS are shown in Table 5.

The database mined is dynamic, new records are unceasing inserted into the database. Records in Table 6 are the new-added design data of translating information from CRs to DRs. The association rule base must to be updated to support the PSS conceptual design.

Step 1. Extract the association rules from the new database by implementing the proposed association rule mining approach. The new rules are shown in Table 7.

Step 2. Re-examine the rules output from the original database.

The weights of the original database and the new database are set as 0.6 and 0.4, respectively. The support and confidence values of the rules in the original rule base are modified as shown in Table 8. If the weighted support value of a rule is equal or more than 0.3, the rule is maintained in the updated association rule base. Otherwise, the rule is stored in the temporary set temp.

Step 3. Examine the infrequent itemsets stored in the CS.

The support and confidence values of the itemsets in the CS are modified as shown in Table 9. If the weighted support value of an itemset is equal or more than 0.3, the itemset changes to be an association rule in the updated association rule base. Otherwise, the itemset is stored in the temporary set temp.

Step 4. Re-examine the rules output from the incremental database.

The support and confidence values of the rules in the new rule base are modified as shown in Table 10. If the weighted support value of a rule is equal or more than 0.3, the rule is added into the updated association rule base. Otherwise, the rule is stored in the temporary set temp.

Step 5. Output the updated association rule base.

Step 6. Output the new CS

Set Mincruc=0.25. Check the support value of each itemset in temp, if it is equal or more than 0.25, the itemset is added into the new CS.

**Table 7** The association rules extracted from the new database

| No. | IF (CRs) | THEN (DRs) | Support | Confidence |
|---|---|---|---|---|
| 1 | Flow accuracy=H | Error of flow=0.3<br>Component reliability=MH<br>Technical supporting level=H | 0.43 | 1 |
| 2 | Flow accuracy=MH | Error of flow=0.5 | 0.43 | 1 |
| 3 | Operating reliability=MH | Component reliability=MH<br>Service completion time=3<br>Reliability of PM=M | 0.43 | 1 |
| 4 | Broken time=ML | Component reliability=MH | 0.43 | 0.75 |
| 5 | Performance/cost=H | Error of flow=0.3 | 0.43 | 0.75 |
| 6 | Performance/cost=H | Service response time=24 | 0.43 | 0.75 |
| 7 | Performance/cost=H | Technical supporting level=H | 0.43 | 0.75 |

**Table 8** Results of re-examining the original association rule base

| No. | IF (CRs) | THEN(DRs) | Support | Confidence | Is a rule in the updated rule base |
|---|---|---|---|---|---|
| 1 | Flow accuracy=H | Flow adjustment level=50:1<br>Error of flow=0.3<br>Service response time=24 | 0.26 | 0.63 | N |
| | Flow accuracy=H | Error of flow=0.3 | 0.37 | 0.90 | Y |
| | Flow accuracy=H | Service response time=24 | 0.31 | 0.76 | N |
| | Flow accuracy=H | Flow adjustment level=50:1 | 0.26 | 0.63 | N |
| 2 | Flow accuracy=MH | Component reliability=MH | 0.31 | 0.76 | N |
| 3 | Operating reliability=MH | Reliability of PM=M | 0.37 | 0.90 | Y |
| 4 | Operating reliability=MH | Reliability of PM=M | 0.37 | 0.90 | Y |
| 5 | Broken time=ML | Error of flow=0.3<br>Component reliability=MH | 0.31 | 0.76 | N |
| | Broken time=ML | Error of flow=0.3 | 0.31 | 0.76 | N |
| | Broken time=ML | Component reliability=MH | 0.37 | 0.90 | Y |
| 6 | Broken time=ML | Component reliability=MH<br>Service response time=24 | 0.31 | 0.76 | N |
| | Broken time=ML | Service response time=24 | 0.31 | 0.76 | N |
| 7 | Performance/cost=H | Component reliability=MH | 0.37 | 0.90 | Y |

## 6 Conclusions

Manufacturing companies are striving to increase customer satisfaction and obtain a larger share of the market to enhance competitiveness and profitability. The concept of PSS, which has emerged to promote sustainable production and consumption, can be deemed as a competitive strategy to satisfy diverse requirements from customers. In this paper, an association rule mining and maintaining approach in dynamic database is proposed to obtain parameter translat-

**Table 9** Results of re-examining the competitive itemsets set

| No. | CRs | DRs | *Support* | *Confidence* | Is a rule in the updated rule base |
|---|---|---|---|---|---|
| 1 | Flow accuracy=H | Component reliability=MH | 0.33 | 0.53 | N |
| 2 | Flow accuracy=H | Reliability of PM=H | 0.27 | 0.51 | N |
| 3 | Flow accuracy=H | Technical supporting level=H | 0.33 | 0.53 | N |
| 4 | Flow accuracy=MH | Error of flow=0.5 | 0.33 | 0.61 | Y |
| 5 | Flow accuracy=MH | Technical supporting level=H | 0.22 | 0.57 | N |
| 6 | Operating reliability=MH | Component reliability=MH | 0.33 | 0.61 | Y |
| 7 | Operating reliability=MH | Technical supporting level=H | 0.27 | 0.59 | N |
| 8 | Broken time=ML | Service completion time=1 | 0.22 | 0.43 | N |
| 9 | Broken time=ML | Technical supporting level=H | 0.27 | 0.45 | N |
| 10 | Broken time=ML | Reliability of PM=H | 0.27 | 0.45 | N |
| 11 | Broken time=ML | Flow adjustment level=80:1 | 0.22 | 0.43 | N |
| 12 | Broken time=ML | Self-monitoring ability=M | 0.22 | 0.43 | N |
| 13 | Performance/cost=H | Error of flow=0.3 | 0.33 | 0.61 | Y |
| 14 | Performance/cost=H | Service response time=24 | 0.33 | 0.61 | Y |

**Table 10** Results of re-examining the association rules attracted from the new database

| No. | IF (CRs) | THEN (DRs) | Support | Confidence | Is a rule in the updated rule set |
|---|---|---|---|---|---|
| 1 | Flow accuracy=H | Error of flow=0.3<br>Component reliability=MH<br>Technical supporting level=H | 0.25 | 0.7 | N |
| | Flow accuracy=H | Error of flow=0.3 | 0.41 | 1 | Y |
| | Flow accuracy=H | Component reliability=MH | 0.29 | 0.7 | N |
| | Flow accuracy=H | Technical supporting level=H | 0.29 | 0.7 | N |
| 2 | Flow accuracy=MH | Error of flow=0.5 | 0.33 | 0.88 | Y |
| 3 | Operating reliability=MH | Component reliability=MH<br>Service completion time=3<br>Reliability of PM=M | 0.25 | 0.64 | N |
| | Operating reliability=MH | Component reliability=MH | 0.37 | 0.64 | Y |
| | Operating reliability=MH | Service completion time=3 | 0.29 | 1 | N |
| | Operating reliability=MH | Reliability of PM=M | 0.33 | 0.64 | Y |
| 4 | Broken time=ML | Component reliability=MH | 0.33 | 0.56 | N |
| 5 | Performance/cost=H | Error of flow=0.3 | 0.33 | 0.7 | Y |
| 6 | Performance/cost=H | Service response time=24 | 0.33 | 0.7 | Y |
| 7 | Performance/cost=H | Technical supporting level=H | 0.33 | 0.7 | Y |

ing rules aiding two domain mapping process in PSS conceptual design.

The major characteristics of this research are summarized as follows.

1. Parameter translating rules are extracted from historical PSS design data. These rules can provide the designers with sufficient knowledge to aid analyzing parameter translation between two adjacent domains to decrease the subjectivity and complexity of the decision-making process.
2. The incremental dynamicity of the design database is considered in the paper. An incremental updating technique is proposed to maintain the association rules without retracing the original database. Moreover, it can highlight the novelty of the new-added records and extract fashionable rules as much as possible.
3. According to the requirements of the incremental updating technique, the promising infrequent itemsets are also output in the apriori-based association rule mining algorithm. Rule redundancy and conflicting are considered in the whole rule mining and maintaining process in order to ensure the effectiveness and availability of rules.

**Acknowledgments** The project was supported by the National Natural Science Foundation, China (no.51075261), Shanghai Science and Technology Innovation Action Plan (no.09dz1124600, no.10dz1121600), Shanghai Jiao Tong University Innovation Fund for Postgraduates. The authors would also like to express their grateful appreciation to the anonymous referees for their helpful comments to improve the quality of this paper.

## References

1. Aurich JC, Wolf N, Siener M, Schweitzer E (2009) Configuration of product-service systems. J Manuf Technol Manag 20(5):591–605
2. Manzini E, Vezzoli C (2003) A strategic design approach to develop sustainable product service systems: examples taken from the“environmentally friendly innovation” Italian prize. J Clean Prod 11(8):851–857
3. Williams A (2007) Product service systems in the automobile industry: contribution to system innovation? J Clean Prod 15(11–12): 1093–1103
4. Aurich JC, Fuchs C, DeVries MF (2004) An approach to life cycle oriented technical service design. CIRP Ann Manuf Technol 53 (1):151–154
5. Aurich JC, Fuchs C, Wagenknecht C (2006) Life cycle oriented design of technical Product–Service Systems. J Clean Prod 14 (17):1480–1494
6. Sakao T, Shimomura Y (2007) Service engineering: a novel engineering discipline for producers to increase value combining service and product. J Clean Prod 15(6):590–604
7. Sakao T, Shimomura Y, Sundin E, Comstock M (2009) Modeling design objects in CAD system for service/product engineering. Computer-Aided Des 41(3):197–213
8. Bitran G, Pedrosa L (1998) A structured product development perspective for service operations. Eur Manag J 16(2):169–189
9. Suh N (1990) The principles of design. Oxford University Press, USA

10. Griffin A, Hauser JR (1993) The voice of the customer. Mark Sci 12(1):1–27
11. Shen XX, Tan KC, Xie M (2001) The implementation of quality function deployment based on linguistic data. J Intell Manuf 12:65–75
12. Han CH, Kim JK, Choi SH (2004) Prioritizing engineering characteristics in quality function deployment with incomplete information: a linear partial ordering approach. Int J Prod Econ 91:235–249
13. Fung R, Popplewell K, Xie J (1998) An intelligent hybrid system for customer requirements analysis and product attribute targets determination. Int J Prod Res 36(1):13–34
14. Harding J, Popplewell K, Fung R, Omar A (2001) An intelligent information framework relating customer requirements and product characteristics. Comput Ind 44(1):51–65
15. Chen YZ, Tang J, Fung RYK et al (2004) Fuzzy regression-based mathematical programming model for quality function deployment. Int J Prod Res 42(5):1009–1027
16. Fung RYK, Chen YZ, Tang JF (2006) Estimating the functional relationships for quality function deployment under uncertainties. Fuzzy Set Syst 157:98–120
17. Kim KJ, Moskowitz H, Dhingra A, Evans G (2000) Fuzzy multi-criteria models for quality function deployment. Eur J Oper Res 121:504–518
18. Zhang XP, Bode J, REn SJ (1996) Neural networks in quality function deployment. Comput Ind Eng 31(3/4):669–673
19. Jiao J, Zhang Y (2005) Product portfolio identification based on association rule mining. Computer-Aided Des 37(2):149–172
20. Shao X, Wang Z, Li P, Feng C (2006) Integrating data mining and rough set for customer group-based discovery of product configuration rules. Int J Prod Res 44(14):2789–2812
21. Romanowski C, Nagi R (2004) A data mining approach to forming generic bills of materials in support of variant design activities. J Comput Inf Sci Eng 4(4):316–328
22. Sadoyan H, Zakarian A, Mohanty P (2006) Data mining algorithm for manufacturing process control. Int J Adv Manuf Technol 28:342–350
23. Alisantoso D, Khoo LP, Ivan Lee BH, Fok SC (2005) A rough set approach to design concept analysis in a design chain. Int J Adv Manuf Technol 26:427–435
24. Yin J, Li D, Peng Y (2006) Knowledge acquisition from metal forming simulation. Int J Adv Manuf Technol 29:279–286
25. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large database. Proceedings of 20th International Conference on Very Large Data Bases, Santiago, Chile, 1–32
26. Agrawal R, Imieliński T, Swami A (1993) Mining association rules between sets of items in massive databases. Proceedings of ACM/SIGMOD International Conference on Management of Data, 207–216
27. Cheung D, Ng V, Fu A, Fu Y (1996) Efficient mining of association rules in distributed databases. IEEE Trans Knowl Data Eng 8(6):911–922
28. Chen Y, Weng C (2008) Mining association rules from imprecise ordinal data. Fuzzy Set Syst 159(4):460–474
29. Delgado M, Marin N, Sánchez D, Vila M (2003) Fuzzy association rules: general model and applications. IEEE Trans Fuzzy Syst 11(2):214–225
30. Cheung D, Han J, Ng V, Wong C (1996b) Maintenance of discovered association rules in large databases: An incremental updating technique. Proceeding of the 12th international conference on data engineering, New Orleans, Louisiana, March 1996: 106–114
31. Cheung DW, Lee SD, Kao B (1997) A general incremental technique for maintaining discovered association rules. Proceeding of the 5th international conference on database systems for advanced applications, MElboume, 1997:185–194
32. Lee S, Cheung D, Kao B (1998) Is sampling useful in data mining? A case in the maintenance of discovered association rules. Data Min Knowl Disc 2(3):233–262
33. Dudek D (2009) RMAIN: Association rules maintenance without reruns through data. Inform Sci 179(24):4123–4139
34. Zhang S, Zhang C, Yan X (2003) Post-mining: maintenance of association rules by weighting. Inf Syst 28(7):691–707