Adv. Manuf. (2017) 5:321–334
https://doi.org/10.1007/s40436-017-0205-6

Thinking process rules extraction for manufacturing process design

Jing-Tao Zhou[1] · Xiang-Qian Li[2] · Ming-Wei Wang[1] · Rui Niu[1] · Qing Xu[1]

Received: 4 July 2017 / Accepted: 13 November 2017 / Published online: 5 December 2017

Abstract To realize the reuse of process design knowledge and improve the efficiency and quality of process design, a method for extracting thinking process rules for process design is proposed. An instance representation model of the process planning reflecting the thinking process of technicians is established to achieve an effective representation of the process documents. The related process attributes are extracted from the model to form the related events. The manifold learning algorithm and clustering analysis are used to preprocess the process instance data. A rule extraction mechanism of process design is introduced, which is based on the related events after dimension reduction and clustering, and uses the association rule mining algorithm to realize the similar process information extraction in the same cluster. Through the vectorization description of the related events, the final process design rules are formed. Finally, an example is given to evaluate the method of process design rules extraction.

Keywords Process design rules · Process design rules extraction · Process planning model · Related event · Manifold learning

1 Introduction

Manufacturing enterprises have accumulated a large amount of process data in the process of product design and manufacturing, which has become a knowledge resource. How to acquire and reuse this process knowledge [1, 2] is one of the key problems to be solved in reusing process knowledge and promoting the standardization of process technology, and it is also the application bottleneck of existing computer-aided process planning (CAPP) systems. The experience knowledge of process design mainly exists in the brains of technicians, and it mainly refers to the design rules that have the internal logical relationship in the process of the whole process design decision and reasoning [3]. Therefore, the key to acquiring the experience knowledge of process design is to obtain the design rules with internal logical relationships. Instances are the reflection of human thinking activities, and the knowledge is hidden in a large number of instances. How to extract the process design rules from the existing process design instances is the key to this process and acquiring the expert experience knowledge of the process design is difficult.

2 Related work

At present, research achievements regarding knowledge acquisition of process design experience can be divided into four categories: process knowledge acquisition based on a rough set (RS); process knowledge acquisition based on artificial neural networks (ANNs); process knowledge acquisition based on the case-based reasoning (CBR); and process knowledge acquisition based on the fusion of multiple algorithms.

✉ Jing-Tao Zhou
zhoujt@nwpu.edu.cn

[1] Key Laboratory of Contemporary Design and Integrated Manufacturing Technology, Ministry of Education of China, Northwestern Polytechnical University, Xi'an 710072, People's Republic of China

[2] Department of Software and Information Service Research, Electronic Technology Information Research Institute, Ministry of Industry and Information Technology of the People's Republic of China, Beijing 100040, People's Republic of China

Beynon et al. [4] offered an approach using RS theory to generate rules for use in expert systems and for the traditional statistical task of classification. Zhang et al. [5] presented parallel RS-based methods for knowledge acquisition using MapReduce to mine knowledge from big data. Chao and Sun [6] applied RS to remove the properties without affecting the choice of welding type, and established the welding type selection decision table to obtain valuable welding process knowledge. Augasta and Kathirvalavakumar [7] proposed a new rule extraction algorithm based on ANNs that relies on a reverse engineering technique to prune the insignificant input neurons. Masood and Hassan [8] used feature-based ANNs to realize the pattern recognition for bivariate process mean shifts. Khorasani and Yazdi [9] developed a dynamic surface roughness monitoring system based on an ANN in a milling operation that uses cutting conditions as input and surface roughness as output. Armaghan and Renaud [10] proposed using knowledge acquisition as a basis for seeking solutions from non-compensatory multi-criteria decision aids. Olsson et al. [11] applied CBR to geometric measurements for decision support in manufacturing, which allows product measurements and related adjustments to the production line to be stored as cases in a CBR system. Zhang et al. [12] proposed a body-in-white fixture design method integrating CBR with rule-based reasoning. Prado et al. [13] introduced a new method for fuzzy-rule evolution that forms expert system knowledge: knowledge acquisition with a swarm-intelligence approach, which is based on the use of particle-swarm optimization to obtain the antecedents, consequences, and connectives of the rules. Tsai [14] used self-organizing map (SOM) nets and *K*-means clustering for low-temperature welding process data mining, analyzing the knowledge of process with strong characteristics. Jiang et al. [15] proposed a hybrid approach of RS and CBR to remanufacturing process planning that could reuse knowledge generated from existing parts remanufacturing to rapidly create sound process planning for the new arrival of used parts. On the whole, these methods are not focused on the specific thinking process of technicians in the process of the process design, resulting in the deflection between the results of acquisition and the actual process knowledge, influencing the sharing and reuse of process knowledge.

In addition, Efthymiou et al. [16] introduced a knowledge framework based on semantic technology and artificial intelligence approaches to facilitate the definition, storage, and extraction of knowledge in terms of past production process configurations. Chryssolouris et al. [17] demonstrated the different approaches followed for knowledge management in manufacturing process modeling. Obayashi [18] presented multi-objective design exploration (MODE) and its application for design rule extraction. Anantasarn et al. [19] applied process intensification to achieve sustainable process design. Batoulis et al. [20] introduced a semi-automatic approach to identify decision logic in process models. Kluza and Nalepa [21] proposed a method for generating and designing business processes with business rules.

In this paper, a method of process design rules extraction is proposed. This method builds an instance representation model of process planning that reflects the thinking process of technicians. Based on the model, the related events with causal relationships are established and taken as the object of analysis, and then the process design rules are extracted.

3 Definition and representation of the process planning model

To extract the process design rules effectively, this paper introduces the instance representation model to characterize the process planning.

Definition 1 Instance representation model of process planning (*IRMPP*). This model is composed of the following elements: the processing requirements, the geometric features, the machining features, the processing steps, the processing process, and the relationship between various processing attributes.

<*IRMPP*> = <processing requirements, geometric features, machining features, processing steps, processing process, the relationships between various processing attributes>

It can be formalized as

$$\langle IRMPP\rangle = \langle \boldsymbol{X}, \boldsymbol{J}, \boldsymbol{T}, \boldsymbol{C}, \boldsymbol{G}, \boldsymbol{V}\rangle. \tag{1}$$

Definition 2 The relationships between various processing attributes $\boldsymbol{V}$ have

$$\boldsymbol{V} = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7\}, \tag{2}$$

where V_1 represents the relationship between the processing requirements $\boldsymbol{X}$ and the geometric features $\boldsymbol{J}$, which is a directed line segment with $\boldsymbol{X}$ as the starting point and $\boldsymbol{J}$ as the end point; it indicates that the state of $\boldsymbol{X}$ will affect the state distribution of $\boldsymbol{J}$; V_2 represents the relationship between geometric features $\boldsymbol{J}$ and machining features $\boldsymbol{T}$; V_3 represents the relationship between machining features $\boldsymbol{T}$ and processing step $\boldsymbol{C}$; V_4 represents the relationship between processing step $\boldsymbol{C}$ and processing process $\boldsymbol{G}$; V_5 represents the relationship among machining features; V_6 represents the relationship among processing processes; and V_7 represents the relationship among processing steps. The structure of the model is illustrated in Fig. 1.

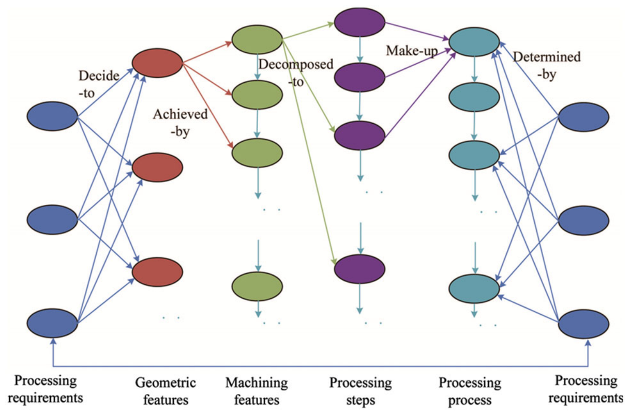

Fig. 1 Structure of the model

In the process instance representation model built in this paper, the processing requirements are used to represent the problem description of the instance, and the other four attributes and the relationships of the attributes are used to represent the solution of the problem.

Definition 3 Related process attributes. This refers to two process attributes that have a relationship in the process planning model. In this paper, it refers to: (geometric features, machining features), (machining features, processing steps), (processing process, processing steps), (processing requirements, processing process), (processing requirements, geometric features).

Definition 4 Process variable. Each process attribute is composed of many constraint elements that are called process variables.

Definition 5 Independent attribute instance a. This refers to a specific instance of each class of process attribute.

Definition 6 Longitudinal related event $\boldsymbol{B}$. This refers to the process attribute instance chain composed of the same process attributes that have a causal relationship. The related event can be formalized as

$$\boldsymbol{B} = (b_1, b_2, \cdots, b_p), \tag{3}$$

where b_i $(i \leq p)$ is an instance of this class of process attribute. Since each class of process attribute contains a number of process variables, then Eq. (3) can be further expressed as

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & b_{22} & \cdots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pk} \end{bmatrix}, \tag{4}$$

where b_{ij} $(j \leq k)$ represents the jth process variable instance of the longitudinal related event in the ith process attribute instance.

Definition 7 Related event $\boldsymbol{F}$. This consists of an independent attribute instance and a longitudinal related event, and the independent attribute instance has a causal relationship with the longitudinal related event.

Owing to the five kinds of related process attributes defined in this paper, the related events are divided into five categories according to the independent attribute instance. The formal representation of the related event is

$$\boldsymbol{F} = (a, \boldsymbol{f}(a)) \tag{5}$$

where a represents an instance of independent attributes and $\boldsymbol{f}(a)$ represents the longitudinal related event corresponding to the independent attribute instance.

Definition 8 Related event set. This refers to the collection of the same kind of related events, and it is satisfied that the independent attribute instances of each related event in the set are the same or similar.

The independent attribute instances are the same or similar in a related event set, and an independent attribute instance corresponds to multiple longitudinal related events. Then the formal representation of the corresponding relationship can be expressed as

$$\boldsymbol{f}(a) = [B_1, \quad B_2, \quad \cdots, \quad B_i], \tag{6}$$

where B_i corresponds to a longitudinal related event in the longitudinal related event set.

Definition 9 Instance of a related variable. This refers to a certain number of instances of process variables with a high frequency of longitudinal related events that occur in the event of a related event set.

Definition 10 Similar process information. This refers to a combination of process variables with high similarity in a large number of instances of process planning.

4 Process design rules extraction approach

4.1 Method overview

The general method of this paper is shown in Fig. 2. It mainly includes the representation of the process planning instance, the representation of the process instance data, and the extraction of the process design rules.

(i) Process instance representation model. This part is mainly concerned with the model representation of the process planning instance. Through the construction of the causal relationship among the process attributes, it is further combined to form an instance representation model of the process planning.

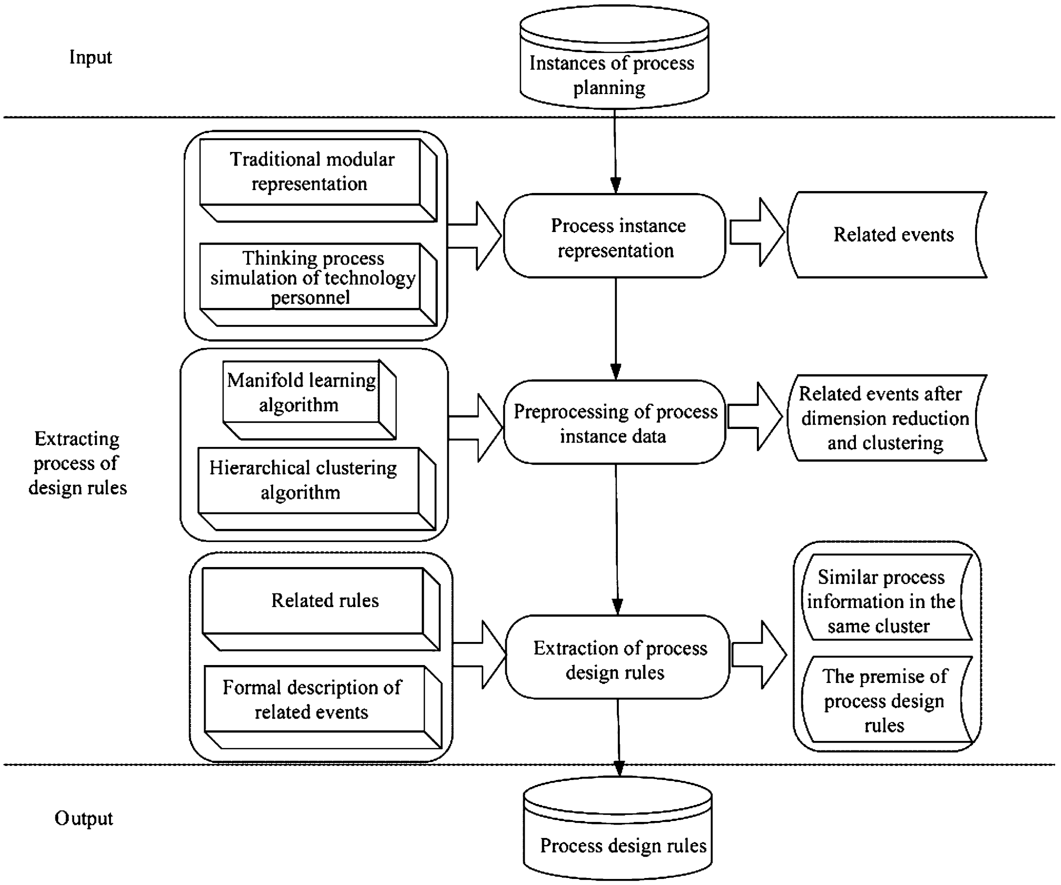
Input
Instances of process planning
Traditional modular representation
Thinking process simulation of technology personnel
Process instance representation
Related events
Extracting process of design rules
Manifold learning algorithm
Hierarchical clustering algorithm
Preprocessing of process instance data
Related events after dimension reduction and clustering
Related rules
Formal description of related events
Extraction of process design rules
Similar process information in the same cluster
The premise of process design rules
Output
Process design rules

Fig. 2 General method of process design rules extraction

(ii) Preprocessing of process instance data. This part is mainly concerned with the dimension reduction and clustering of related events in the model, which is based on the manifold dimension reduction algorithm and agglomerative hierarchical clustering algorithm. The two algorithms are used to reduce the dimension of the longitudinal related events and improve the similarity of the longitudinal related events of the same cluster, to better extract the similar process information.

(iii) Extraction of process design rules. This part is mainly used to extract the similar process information in the same cluster and formalize the independent attribute instances. After association rules analysis, the similar process information is extracted and as a result of process design rules, the extraction of the process design rules is finally completed by the formal description of the independent attribute instances.

The representation form of the process design rules is

$$Rule\ \ i : K \rightarrow L, \tag{7}$$

where *Rule i* is the process design rule *i*, *K* the premise of the process design rule and is composed of some process variables, and *L* represents the result of the process design rule and is also composed of some process variables.

4.2 Extraction of related events

This paper is based on a large number of process planning instances to extract the process design rules. The extraction of process data from an instance of the process planning is the basis for subsequent work. In this paper, an instance representation model of the process planning is constructed, and the five kinds of process attributes are used as the main body of the model to realize the extraction of the original data. The five kinds of process attributes in this paper refer to processing requirements, geometric features, machining features, processing steps and processing process.

The extraction of similar process information is the key to the process design rule extraction in this paper, and it is equivalent to the case of the related variables.

In the model, there is a causal relationship between the two adjacent process attributes, so any two adjacent attribute instances can be used to form a class of related events. The related event types in the model are classified into five categories. According to the process constraint variable information contained in each process attribute, the related events can be divided into the following five categories.

Class 1. Related event: (processing requirements, geometric features), as shown in Fig. 3.
Class 2. Related event: (geometric features, machining features), as shown in Fig. 4.
Class 3. Related event: (machining features, processing steps), as shown in Fig. 5.
Class 4. Related event: (processing steps, processing process), as shown in Fig. 6.
Class 5. Related event: (processing requirements, processing process), as shown in Fig. 7.

4.3 Preprocessing of process instance data

The dimensionality reduction of process data of the longitudinal related event scan eliminate the redundant process data and better reflect the intrinsic structure of the longitudinal related events. Clustering analysis of the longitudinal related events after dimensionality reduction can reduce the effect of noise, improve the similarity of the longitudinal related events of the same cluster, and improve the accuracy of similar process information extraction from the same cluster longitudinal related events. Therefore, we need to preprocess the process instance data before extracting the process design rules, which includes two steps: the dimensionality reduction of the process instance data and the clustering analysis of longitudinal related events after dimensionality reduction.

4.3.1 Dimensionality reduction of process instance data

Manifold learning is a mathematical analysis method used to obtain the low-dimensional representation of a high-dimensional observation space. It is applied to reduce the dimension of the process information data and find the intrinsic dimension of the data, which is helpful for the subsequent data clustering and further extraction of process design rules. The mathematical definition of manifold learning is as follows [22].

Definition 11 Manifold learning. The observation space data ($\boldsymbol{X} = \{\boldsymbol{X}_i, i = 1, \cdots, n\} \subset \boldsymbol{R}^M$) is given, assuming that the low-dimensional spatial data of the sample $\boldsymbol{X}$ is $\boldsymbol{Y} = \{\boldsymbol{Y}_i, i = 1, \cdots, n\} \subset \boldsymbol{R}^m$ with the unknown nonlinear transformation $f : \boldsymbol{Y}_i = f(\boldsymbol{X}_i)$, $m \ll M$. The embedded mapping of $f : \boldsymbol{R}^M \to \boldsymbol{R}^m$ is C^∞.

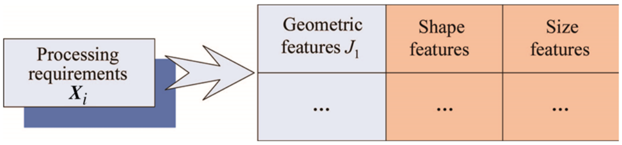

Fig. 3 Schematic diagram of the relationship between processing requirements and geometric features

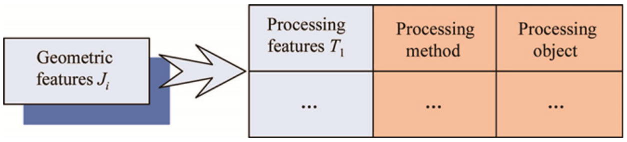

Fig. 4 Schematic diagram of the relationship between geometric features and machining features

The three classic algorithms of manifold learning are Laplacian eigenmaps (LE) [23], locally-linear embedding (LLE) [24], and isometric mapping (ISOMAP) [25]. Compared with the other two typical dimensionality reduction algorithms, the LLE algorithm has no dimension limitation, the parameters of the algorithm are only the nearest-neighbor number and the target dimension, and there is a global optimal solution. The iterative process of the ISOMAP algorithm is avoided and the computation is reduced relatively. It is more difficult to determine the parameters of the adjacency matrix in the LE algorithm, and it is more sensitive to noise. Compared with the LE algorithm, the LLE algorithm is relatively small and easy to implement. Therefore, the LLE algorithm is used to reduce the dimension of the process instance data in this paper.

The dimension reduction process of process instance data based on the LLE algorithm is as follows.

(i) Convert the data of each longitudinal related event into the row vector

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & b_{22} & \cdots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pk} \end{bmatrix}. \tag{8}$$

The data of each row vector represents a process attribute instance in the data matrix of the longitudinal related events. The adjacent process attribute instances that the adjacent row vectors represent have a causal relationship. According to the sequence of the data in the process instance representation model, the data of each process attribute in the longitudinal related event can be transformed into the vector data. The vector formula that the matrix is transformed into is

$$\boldsymbol{B} = (b_{11}, b_{12}, \cdots, b_{pk}). \tag{9}$$

Each variable of a longitudinal related event constitutes a dimension of the vector. A sample set $\boldsymbol{X}$ of the LLE algorithm is formed after conversion of the longitudinal

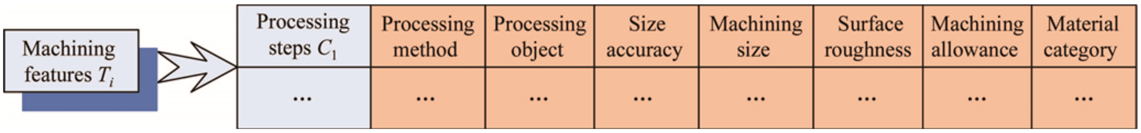

Fig. 5 Schematic diagram of the relationship between machining features and processing steps

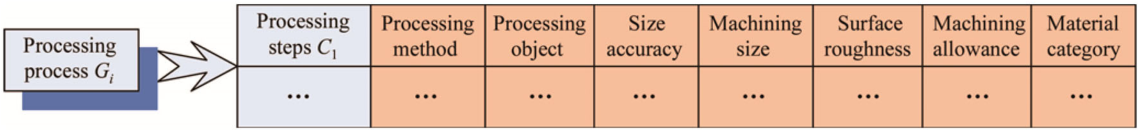

Fig. 6 Schematic diagram of the relationship between processing process and processing steps

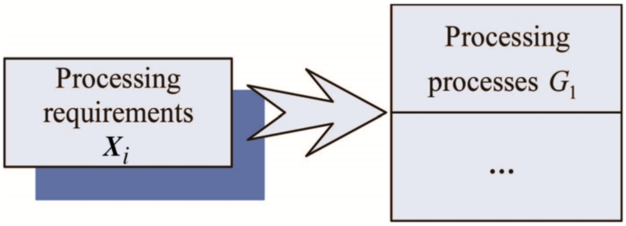

Fig. 7 Schematic diagram of the relationship between processing requirements and processing process

related event data set. The process data $\boldsymbol{B}$ of each longitudinal related event corresponds to a sample data $\boldsymbol{X_i}$. The sample data are mapped to the new data set $\boldsymbol{Y}=\{\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_n\},\boldsymbol{Y}_i\in\boldsymbol{R}^m$ by the LLE algorithm.

(ii) Calculate the k-order neighborhood of each longitudinal related event in the longitudinal related event set.

According to the input data sets of a longitudinal related event $\boldsymbol{X}=\{\boldsymbol{X}_i,i=1,\cdots,n\}$, the Euclidean distance $d_x(i,j)$ between each sample point $\boldsymbol{X}_i(i=1,\cdots,n)$ and the other sample point $\boldsymbol{X}_j(j\neq i)$ is determined. Take the nearest k sample points as the nearest neighborhood points.

(iii) Calculate the reconstruction weights $\boldsymbol{w}_i=\{w_{i1},w_{i2},\cdots,w_{ik}\}$ of the sample point $\boldsymbol{X}_i$ and the neighborhood points.

The sample point $\boldsymbol{X}_i$ is approximated by a linear combination of nearest neighborhood points, minimizing the following reconstruction function values

$$\varepsilon(\boldsymbol{W})=\sum_{i=1}^{n}\left(\boldsymbol{X}_i-\sum_{j=1}^{k}w_{ij}g_j\right)^2, \qquad (10)$$

where $g_j(j=1,2,\cdots,k)$ are the k nearest neighborhood points of $\boldsymbol{X}_i$ and w_{ij} is the reconstruction weight of the nearest neighbor point g_j to the sample point $\boldsymbol{X}_i$.

(iv) All the sample points are mapped into the low-dimensional space so that the mapping satisfies the following formula

$$\varphi(\boldsymbol{Y})=\sum_{i=1}^{n}\left(\boldsymbol{Y}_i-\sum_{i=1}^{k}w_{ij}\boldsymbol{h}_j\right)^2, \qquad (11)$$

where $\varphi(\boldsymbol{Y})$ is the loss function, $\boldsymbol{Y}_i$ the output vector of $\boldsymbol{X}_i$, $\boldsymbol{h}_j(j=1,2,\cdots,k)$ the k nearest neighborhood points of $\boldsymbol{Y}_i$, and satisfy the following formula

$$\begin{cases}\sum\limits_{i=1}^{n}\boldsymbol{Y}_i=0,\\ \dfrac{1}{N}\sum\limits_{i=1}^{n}\boldsymbol{Y}_i\boldsymbol{Y}_i^{\mathrm{T}}=I.\end{cases} \qquad (12)$$

4.3.2 *Clustering analysis of longitudinal related events after dimensionality reduction*

In this paper, we use the agglomerative hierarchical clustering algorithm to cluster the longitudinal related events. To specify the clustering process of related events, in this paper the process sequence of shaft sleeve parts is used as an example of dimensionality reduction for clustering analysis. The clustering process of other related events is similar to the given example.

The related mathematical description of the clustering process of the shaft sleeve parts process sequence is as follows.

(i) The formal representation of the shaft sleeve parts process sequence in Table 1 is

$$\boldsymbol{X}_{n\times n}=\begin{bmatrix}x_{11} & x_{12} & \cdots & x_{1n}\\ x_{21} & x_{22} & \cdots & x_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ x_{n1} & x_{n2} & \cdots & x_{nn}\end{bmatrix}, \qquad (13)$$

Table 1 Information for the shaft sleeve parts machining process

Serial number	Part name	Step						
1	Screw	Blanking	Rough turning	Half finishing turning	Turning thread	Milling	Clamping	Inspection
2	Guide pillar	Blanking	Rough turning	Finishing turning	Milling	Grinding	Clamping	Inspection
3	Stud	Blanking	Rough turning	Half finishing turning	Turning thread	Grinding	Inspection	
4	Threaded bushing	Blanking	Rough turning	Finishing turning	Drilling	Turning thread	Grinding	Inspection
5	Orifice bushing	Blanking	Rough turning	Finishing turning	Drilling	Boring	Grinding	Inspection
6	Bolt	Blanking	Rough turning	Half finishing turning	Turning thread	Milling	Grinding	Inspection
7	Shaft	Blanking	Rough turning	Finishing turning	Milling	Grinding		Inspection
8	Bearing sleeve	Blanking	Rough turning	Finishing turning	Drilling	Boring	Clamping	Inspection
9	Dowel pin	Blanking	Rough turning	Finishing turning	Grinding	Clamping	Inspection	

where element x_{ij} in the matrix $\boldsymbol{X}$ is the jth processing procedure of the ith shaft sleeve parts process sequence in Table 1.

(ii) Distance between the process sequences of different parts. The dissimilarity matrix for each object in the Eq. (13) is

$$\boldsymbol{D}_{n\times n} = \begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ \vdots & \vdots & \ddots & \\ d(n,1) & d(n,2) & \cdots & 0 \end{bmatrix}. \tag{14}$$

This is an $n \times n$ matrix that represents the similarity of two objects in n objects.

Any element $d(i, j)$ in Eq. (14) represents the dissimilarity of the process sequence between the ith part and the jth part in Table 1(distance of these two objects). A variety of distance measurement methods are available and the Euclidean distance measurement method is widely used. In this paper, the Euclidean distance formula is used to calculate the distance between the process sequence of each part. The formula for Euclidean distance is

$$d(i,j) = \|\boldsymbol{X}_i - \boldsymbol{X}_j\|_2 = \left[\sum_{k=1}^{q} |\boldsymbol{X}_{ik} - \boldsymbol{X}_{jk}|^2\right]^{1/2}. \tag{15}$$

(iii) Distance of clusters of the shaft sleeve part process sequence. According to the agglomerative hierarchical clustering algorithm, in the process of process sequence clustering of the part, first the distance of clusters is calculated and the clusters whose distance is small are combined, then the number of these clusters is reduced. The formulas for calculating the distance between clusters include the minimum distance, the maximum distance, the mean distance, and the average distance methods. In this paper, we use the minimum distance to measure the distance between two clusters. The formula for calculating the minimum distance is

$$d_{\min}(c_i, c_j) = \min_{p\in c_i, p'\in c_j} |p - p'| \,, \tag{16}$$

where c_i, c_j represent two clustering clusters; m_i, m_j represent the average values of two clusters and $|p - p'|$ represents the distance between two objects.

4.4 Extraction of process design rules

After clustering, the similarity of longitudinal related events in the same cluster is increased. The similar process information is extracted by analyzing the longitudinal related events of the same cluster, which is the result of the process design rules. Through the formal description of the independent attribute instances in the related events, the premise of the process design rules is obtained, and finally the complete process design rules are formed.

4.4.1 Similar process information extraction in the same cluster

Each longitudinal related event is composed of a number of process variables. In the same cluster, the similar process information of the longitudinal related events corresponds to the related variables of high frequency. Finding the high-frequency related variables is the key to the extraction of similar process information.

A data mining algorithm based on association rules [26] is used to reveal the relationship of the intrinsic properties of things. It can find a number of frequent item sets in the input objects. Using association rules to deal with the longitudinal related events of the same cluster, the process variables combinated with higher frequency in the longitudinal related events can be regarded as a number of frequent item sets. The association rules mining algorithm can effectively extract the related variables of higher frequency in the same cluster longitudinal related events.

(i) Related concepts

The analysis object of the association rules is the thing, and the set of things is usually made up of things identifier and an items set. Here $\boldsymbol{I} = \{I_1, I_2, \cdots, I_n\}$ is the items set, and $\boldsymbol{A} \Rightarrow \boldsymbol{B}$ is the association rule, where A, B meet $\boldsymbol{A} \subset \boldsymbol{I}, \boldsymbol{B} \subset \boldsymbol{I}, \boldsymbol{A} \neq \varnothing, \boldsymbol{B} \neq \varnothing, \boldsymbol{A} \cap \boldsymbol{B} \neq \varnothing$.

If $\boldsymbol{A} \Rightarrow \boldsymbol{B}$ is established in the things set $\boldsymbol{D}$, its support level is s and its confidence level is c.

Here s is the proportion of the total number of things in things set $\boldsymbol{D}$ that contains $\boldsymbol{A}$ and $\boldsymbol{B}$ at the same time from the total number of things in $\boldsymbol{D}$. The formal representation of s is

$$sup(\boldsymbol{A} \Rightarrow \boldsymbol{B}) = P(\boldsymbol{A} \cup \boldsymbol{B}). \tag{17}$$

The confidence level c is the probability that thing $\boldsymbol{B}$ happens at the same time that $\boldsymbol{A}$ happens. The formal representation of c is

$$confidence(\boldsymbol{A} \Rightarrow \boldsymbol{B}) = P(\boldsymbol{B}|\boldsymbol{A}) = sup(\boldsymbol{A} \Rightarrow \boldsymbol{B})/sup(\boldsymbol{A}). \tag{18}$$

When the support and confidence levels of association rules are greater than or equal to the minimum support and confidence levels, the strong association rules are generated.

Concept of an item set. The collection of items is called the item set. A set containing K items is called a K item set, and the number of things of the K item set is called the support count or frequency of the K item set. If the support of the item set $\boldsymbol{I}$ is more than the minimum support, it is called a frequent item set.

(ii) Apriori algorithm

In this paper, the Apriori algorithm is used to mine the association rules, and is based on the longitudinal related events after clustering.

The core idea of the algorithm is that it is a recursive algorithm based on the idea of a two-stage set. The K items set is derived from the $K-1$ items set recursively, and is based on iterative search method. The specific process is to find a frequent 1 items set $\boldsymbol{L}_1$, then the objects in $\boldsymbol{L}_1$ are combined to find the candidate 2 items set, and then the frequent 2 items set $\boldsymbol{L}_2$ is found. We wait until the K items set is not frequent. The algorithm has two important properties: a non-empty subset of a frequent items set is also a frequent items set; a frequent items set cannot contain a non-frequent items set.

Two important steps in the process of the Apriori algorithm. Connection step: The connection step is the process of generating candidate items set $\boldsymbol{C}_k$ from frequent items set $\boldsymbol{L}_{k-1}$. First, the frequent items in the frequent items set $\boldsymbol{L}_{k-1}$ are sorted according to a certain order. $\boldsymbol{I}_i$ is one of the frequent items in $\boldsymbol{L}_{k-1}$, $\boldsymbol{I}_i$ = (I_{i1}, I_{i2}, ···, $I_{i(k-1)}$). Taking two of the frequent items $\boldsymbol{I}_1$, $\boldsymbol{I}_2$, if $\boldsymbol{I}_1$, $\boldsymbol{I}_2$ are the same as the first $k-2$ elements, the last element is different, then through the combination of the elements of $\boldsymbol{I}_1$, $\boldsymbol{I}_2$ to generate a K items set $\boldsymbol{C}_k$, $\boldsymbol{I}_i = (I_{11}, I_{12}, \cdots, I_{1(k-2)}, I_{1(k-1)}, I_{1(k)})$.

Pruning step: The items set $\boldsymbol{C}_k$ generated in the connection step is a super-set of the frequent items set $\boldsymbol{L}_{k-1}$. Items in the items set $\boldsymbol{C}_k$ may be frequent, and may be infrequent. Frequent sub-items of $\boldsymbol{C}_k$ cannot contain infrequent $K-1$ items. Based on the above analysis, if a K-item in the candidate items set $\boldsymbol{C}_k$ does not contain the frequent $K-1$ item, the candidate K-item is not a frequent item and is deleted from the candidate items set. The entire $\boldsymbol{C}_k$ is traversed to determine the $\boldsymbol{L}_k$.

(iii) Similar process information extraction

After the related events of the same cluster are analyzed by the association rules, a number of process variables with high frequency in the same cluster longitudinal related events are combined and outputted. By analyzing the definition of the longitudinal related events, we can see that each longitudinal related event is composed of a certain number of process variables. The formal representation of longitudinal related events is

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & b_{22} & \cdots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pk} \end{bmatrix}. \tag{19}$$

Each element of the matrix represents the corresponding process variable instance of the longitudinal related event. Here b_{ij} $(i \leq p, j \leq k)$ represents an instance of the process variable in the longitudinal related event. Each process variable has some variables, which means that each element in the matrix has some values

$$\boldsymbol{b}_{ij} = ((\boldsymbol{b}_{ij})_1, (\boldsymbol{b}_{ij})_2, \cdots, (\boldsymbol{b}_{ij})_l). \tag{20}$$

After the operation of association rules, the frequent item sets of process variables are outputted. We assume that there are binomial frequent item sets in the output result

$$\begin{cases} (b_{ij})_h \Rightarrow (b_{dn})_z, \\ \quad d \le p, \\ \quad n \le k, \end{cases} \tag{21}$$

where h and z represent the possible values of process variables. We use the frequent items sets as the similar process information for the longitudinal related events of the cluster.

4.4.2 Formation of process design rules

The similar process information extracted from the longitudinal related events of the same cluster is the result of the process design rules, and the premise of the process design rules is not formalized. Here, we combine the related events (processing requirements, processing process), as shown in Fig. 7, to analyze and formalize the independent attribute instances of processing requirements, to determine the premise of process design rules, and ultimately to form the process design rules of the related events set.

(i) Formal description of processing requirements

Because the shaft sleeve parts are discussed as an example in this paper, for such parts, the differences of the processing requirements are mainly reflected in the following aspects: whether the internal diameter exists or not (i.e., shaft or sleeve parts), whether to process the internal thread, whether to process the external thread, the roughness of outer diameter, the roughness of inner diameter, the face roughness, and so on.

In view of the above, consider the different characteristics of the processing requirements of the shaft sleeve parts: inner diameter, external thread, internal thread, outer diameter roughness, inner diameter roughness, face roughness, and we can formalize the processing requirements as $[x_1 \quad x_2 \quad x_4 \quad x_5 \quad x_6 \quad x_7]$, where x_1 is used to denote the presence or absence of the inner diameter, and $x_1 \in \{0, 1\}$, respectively, denote the presence and absence of the inner diameter; similarly, x_2 and x_3 are used to indicate the presence or absence of external and internal threads; x_4 indicates the outer diameter roughness; x_5 represents the inner diameter roughness, and we set the corresponding value of x_5 to null for the shaft part (i.e., no inner diameter); x_6 indicates the face roughness. Then x_4, x_5 and x_6 are normalized, mapped to the [0, 1] interval, and the null value of x_5 is set to 1. Eventually, a vector representation of the processing requirements is formed. In this way, the premise of the process design rules can be accurately described, to provide the basis for the formation of process design rules in the next step, and to provide a convenient retrieval method in the reuse of process design rules.

(ii) Formation of process design rules

After obtaining the vector representation of the processing requirements, the mean vector is obtained for all the processing requirement vectors in the related events set of the same cluster, and as the premise of process design rules. Then, the process design rules of the related events set are formed.

The formal description of the process design rules for the related events set

$$Rule\ \ i : \overline{(\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n)} \rightarrow ((\boldsymbol{b}_{ij})_h, (\boldsymbol{b}_{dn})_z), \tag{22}$$

where $\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n$ denote the processing requirements vectors in the related events set of the same cluster; the superscript denotes the mean vector.

(iii) Reuse of process design rules

We extract the process design rules to improve its reuse efficiency, when a new part requires process design. Firstly, determine the vector representation of its processing requirements. Secondly, the similarity between the processing requirement vectors of the part and the mean vector of each cluster is compared. Finally, the most similar clustering cluster is used to analyze the association rules, and the corresponding process design rules are obtained as a reference for the process design of the new part.

Through the above steps, we can solve the problem of knowledge reuse in the process design of new parts, and we discuss this in detail in the following section using a case study.

5 Case evaluation

To evaluate the validity of the method proposed in this paper, the process sequences of shaft sleeve parts are selected as the objects for evaluation. The details for the shaft sleeve parts machining process are given in Table 1.

5.1 Data preprocessing

According to the common sense in processing technology, the sequence of each processing process in the processing is determined. In addition, the processing accuracy of the processing process is different, and in accordance with the different processing accuracy, the processing processes are sorted from low to high.

Classification assignment. Because a rough turning process generally exists in the processing of shaft sleeve parts, select the rough turning process as the standard process, and assign it to 15. The processing accuracy of each processing process is compared with the rough turning process, and the numerical value is determined. For example, there are half finishing turning and finishing turning in the turning process, which can be assigned to 20 and 35, respectively, according to the processing accuracy. The milling process is generally adjacent to the turning process, and the processing accuracy is lower than that of finishing turning, so it is assigned to 30. The assignment of each processing process is completed in accordance with this.

The minimal-maximum normalization of each processing process. The artificial classification assignment may cause deviation in the numerical size of each processing process. To reduce the deviation and maintain the different processing accuracy of each processing process, the assignment value of each processing process is treated by minimal-maximum normalization

$$v' = \frac{v - m_A}{M_A - m_A}. \tag{23}$$

where v is the the numerical size of each processing process. m_A the minimum value, M_A the maximum value and v' is the value of v after minimal-maximum normalization.

The minimum value of the assignment process is 5 and the maximum value is 90. The value of milling process after the conversion-based Eq. (24) is 0.235 3. For example, the assignment result of the first shaft sleeve part is [10, 15, 20, 25, 30, 35, 40], and according to the Eq. (24), the normalized result is [0.058 8, 0.117 6, 0.117 6, 0.176 5, 0.235 3, 0.294 1, 0.411 8]. The numerical values of each process sequence are shown in Table 2.

The dimension of the data is reduced with the LLE algorithm. The data matrix after dimension reduction is $\boldsymbol{Y}$, then the input sample data $\boldsymbol{X}$ is

$$\boldsymbol{X} = \begin{bmatrix} 0.058\,8 & 0.058\,8 & 0.058\,8 & 0.058\,8 & 0.058\,8 & 0.058\,8 & 0.058\,8 & 0.058\,8 & 0.058\,8 \\ 0.117\,6 & 0.117\,6 & 0.117\,6 & 0.117\,6 & 0.117\,6 & 0.117\,6 & 0.117\,6 & 0.117\,6 & 0.117\,6 \\ 0.176\,5 & 0.352\,9 & 0.176\,5 & 0.352\,9 & 0.352\,9 & 0.176\,5 & 0.352\,9 & 0.352\,9 & 0.352\,9 \\ 0.235\,3 & 0.294\,1 & 0.235\,3 & 0.235\,3 & 0.235\,3 & 0.235\,3 & 0.294\,1 & 0.235\,3 & 0.352\,9 \\ 0.294\,1 & 0.352\,9 & 0.352\,9 & 0.235\,3 & 0.235\,3 & 0.294\,1 & 0.352\,9 & 0.235\,3 & 0.352\,9 \\ 0.352\,9 & 0.352\,9 & 0.411\,8 & 0.352\,9 & 0.352\,9 & 0.352\,9 & 0.411\,8 & 0.352\,9 & 0.411\,8 \\ 0.411\,8 & 0.411\,8 & 0.411\,8 & 0.411\,8 & 0.411\,8 & 0.411\,8 & 0.411\,8 & 0.411\,8 & 0.411\,8 \end{bmatrix}.$$

The ith column data of the matrix $\boldsymbol{X}$ represent the process sequence pretreatment data of the ith shaft sleeve part in Table 1.

Here the number k of the nearest neighbors is set to 3, and the target dimension m is set to 3. A value of k that is too large or too small will affect the smoothness of the manifold or low-dimensional topology. A value of d that is too large or too small will cause too much noise or low-dimensional overlap. The values of k and d are generally based on experience to select a more active value, and may take a number of attempts to determine. We also refer to the study by Hu et al. [27] for further discussion in this area. Through the operation of the LLE algorithm, the process sequence data after dimension reduction is

$$\boldsymbol{Y} = \begin{bmatrix} -0.086\,5 & -0.478\,3 & 0.238\,0 & -1.340\,0 & -0.761\,2 & -0.086\,5 & 1.201\,5 & -0.761\,2 & 2.074\,1 \\ -1.353\,1 & 0.701\,8 & -1.528\,3 & 0.604\,9 & 0.737\,9 & -1.353\,1 & 0.701\,5 & 0.737\,9 & 0.750\,4 \\ 1.000\,0 & 1.000\,0 & 1.000\,0 & 1.000\,0 & 1.000\,0 & 1.000\,0 & 1.000\,0 & 1.000\,0 & 1.000\,0 \end{bmatrix}.$$

Each column of the matrix represents the corresponding output data of the process sequence after LLE dimension reduction.

The distance between the process sequences of parts can be calculated as

$$\boldsymbol{D} = \begin{bmatrix} 0 & & & & & & & & \\ 2.0919 & 0 & & & & & & & \\ 0.3688 & 2.3424 & 0 & & & & & & \\ 2.3248 & 0.8671 & 2.6534 & 0 & & & & & \\ 2.1971 & 0.2852 & 2.4767 & 0.5939 & 0 & & & & \\ 0 & 2.0919 & 0.3688 & 2.3248 & 2.1971 & 0 & & & \\ 2.4249 & 1.6798 & 2.4291 & 2.5433 & 1.9630 & 2.4249 & 0 & & \\ 2.1971 & 0.2852 & 2.4767 & 0.5939 & 0 & 2.1971 & 1.9630 & 0 & \\ 3.0154 & 2.5528 & 2.9264 & 3.4172 & 2.8353 & 3.0154 & 0.8740 & 2.8353 & 0 \end{bmatrix}.$$

The elements of the matrix $\boldsymbol{D}$ represent the Euclidean distance between the process sequences of the nine shaft sleeve parts in Table 1.

The minimum distance is used as the dissimilarity measure formula, and the minimum distance threshold of the clustering object distance is set to 1. The algorithm will terminate when the minimum distance of the clustering object is more than 1.

Taking the data of Table 1 as the samples, the samples of each part are clustered at the beginning of clustering. In the initial algorithm, there are nine clusters: {1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}. Each element in the cluster represents the number identification of the process sequence of these samples in Table 1. The algorithm is terminated after 10 steps. The clustering results are {1, 3, 6},{2,4, 5, 8},{7, 9}.

In addition, we need to determine the processing requirements of the parts, where N represents absence, Y represents presence, NaN represents a null value, and the results are shown in Table 3.

Use 0 instead of N and 1 instead of Y. After the roughness is normalized and the NaN value is replaced by 1, the processing requirements vector is obtained, as shown in Table 4.

The parts {1, 3, 6} are a cluster and the mean vector of the processing requirements is [0, 1, 0, 1, 1, 1]. Similarly, the mean vector of the processing requirements for the parts {2, 4, 5, 8} is [0.75, 0, 0.25, 0, 0.5, 0]. The mean vector of the processing requirements for the parts {7, 9} is [0, 0, 0, 0, 1, 0].

5.2 Rule extraction

Here, the process sequence of cluster {1, 3, 6} is selected as the sample to extract the frequent item set of the process sequence. The variable instances of each longitudinal related event of the cluster {1, 3, 6} are shown in Table 5.

The data is analyzed by the association rules algorithm. Assume that the minimum support count is 2 and the minimum confidence level is 60% (superscript i represents the ith process). The first four processes are the same, and we analyze them from the fourth process.

Step 1: Select candidate 1 items set C_1 as shown in Table 6.

Table 2 The results of the process pretreatment

Serial number	Step						
1	0.058 8	0.117 6	0.176 5	0.235 3	0.294 1	0.352 9	0.411 8
2	0.058 8	0.117 6	0.352 9	0.294 1	0.352 9	0.352 9	0.411 8
3	0.058 8	0.117 6	0.176 5	0.235 3	0.352 9	0.411 8	0.411 8
4	0.058 8	0.117 6	0.352 9	0.235 3	0.235 3	0.352 9	0.411 8
5	0.058 8	0.117 6	0.352 9	0.235 3	0.235 3	0.352 9	0.411 8
6	0.058 8	0.117 6	0.176 5	0.235 3	0.294 1	0.352 9	0.411 8
7	0.058 8	0.117 6	0.352 9	0.294 1	0.352 9	0.411 8	0.411 8
8	0.058 8	0.117 6	0.352 9	0.235 3	0.235 3	0.352 9	0.411 8
9	0.058 8	0.117 6	0.352 9	0.352 9	0.352 9	0.411 8	0.411 8

Table 3 Processing requirements of the parts

Serial number	Part name	Inner diameter	External thread	Internal thread	Outer diameter roughness	Inner diameter roughness	Face roughness
1	Screw	N	Y	N	6.3	NaN	6.3
2	Guide pillar	N	N	N	1.6	NaN	3.2
3	Stud	N	Y	N	6.3	NaN	6.3
4	Threaded bushing	Y	N	Y	1.6	3.2	3.2
5	Orifice bushing	Y	N	N	1.6	1.6	3.2
6	Bolt	N	Y	N	6.3	NaN	6.3
7	Shaft	N	N	N	1.6	NaN	3.2
8	Bearing sleeve	Y	N	N	1.6	1.6	3.2
9	Dowel pin	N	N	N	1.6	NaN	3.2

Table 4 Processing requirements vector

Serial number	Part name	Inner diameter	External thread	Internal thread	Outer diameter roughness	Inner diameter roughness	Face roughness
1	Screw	0	1	0	1	1	1
2	Guide pillar	0	0	0	0	1	0
3	Stud	0	1	0	1	1	1
4	Threaded bushing	1	0	1	0	1	0
5	Orifice bushing	1	0	0	0	0	0
6	Bolt	0	1	0	1	1	1
7	Shaft	0	0	0	0	1	0
8	Bearing sleeve	1	0	0	0	0	0
9	Dowel pin	0	0	0	0	1	0

Table 5 Data for machining process parameters

Serial number	Part name	Step						
1	Screw	Blanking	Rough turning	Half finishing turning	Turning thread	Milling	Clamping	Inspection
3	Stud	Blanking	Rough turning	Half finishing turning	Turning thread	Grinding	Inspection	
6	Bolt	Blanking	Rough turning	Half finishing turning	Turning thread	Milling	Grinding	Inspection

Table 6 Candidate 1 items set

Candidate 1 item	Frequency
Turning thread^4	3
Milling^5	2
Grinding^5	1
Clamping^6	1
Inspection^6	1
Grinding^6	1
Inspection^7	2

Table 7 Frequent 1 items set

Frequent 1 item	Frequency
Turning thread^4	3
Milling^5	2
Inspection^7	2

Table 8 Candidate 2 items set

Candidate 2 item	Frequency
Turning thread^4 $\Rightarrow$ milling^5	2
Turning thread^4 $\Rightarrow$ inspection^7	2
Milling^5 $\Rightarrow$ inspection^7	2

Table 9 Frequent 2 items set

Frequent 2 item	Frequency
Turning thread^4 $\Rightarrow$ milling^5	2
Turning thread^4 $\Rightarrow$ inspection^7	2
Milling^5 $\Rightarrow$ inspection^7	2

Table 10 Candidate 3 items set

Candidate 3 item	Frequency
Turning thread^4 $\Rightarrow$ milling^5 $\Rightarrow$ inspection^7	2

Table 11 Frequent 3 items set

Frequent 3 item	Frequency
Turning thread^4 $\Rightarrow$ milling^5 $\Rightarrow$ inspection^7	2

Step 2: Select frequent 1 items set L_1 as shown in Table 7.
Step 3: Select candidate 2 items set C_2 as shown in Table 8.
Step 4: Select frequent 2 items set L_2 as shown in Table 9.
Step 5: Select candidate 3 items set C_3 as shown in Table 10.
Step 6: Select frequent 3 items set L_3 as shown in Table 11.

Table 12 The confidence level of frequent item

Frequent item	Confidence level
Turning thread[4] ⇒ milling[5]	2/3
Turning thread[4] ⇒ inspection[7]	2/3
Milling[5] ⇒ inspection[7]	2/3
Turning thread[4] ⇒ milling[5] ⇒ inspection[7]	2/3

Step 7: Stop the iterative algorithm to find the frequent items, and the confidence level is calculated as shown in Table 12.

The four strong rules in this cluster are: turning thread[4] ⇒ milling[5], turning thread[4] ⇒ inspection[7], milling[5] ⇒ inspection[7], turning thread[4] ⇒ milling[5] ⇒ inspection[7]. We take the longest term as the result of the process design rule, and combine with the mean vector of the processing requirements for the cluster {1, 3, 6}: [0, 1, 0, 1, 1, 1] to form a process design rule:

Rule 1: [0, 1, 0, 1, 1, 1] → (blanking, rough turning, half finishing turning, turning thread, milling, unknown, inspection)

Similarly, the association rule analysis of the cluster {2, 4, 5, 8} gives the following process design rule:

Rule 2: [0.75, 0, 0.25, 0, 0.5, 0] → (blanking, rough turning, finishing turning, drilling, unknown, unknown, inspection)

For the association rule analysis of the cluster {7, 9}, the following process design rule is available:

Rule 3: [0, 0, 0, 0, 1, 0] → (blanking, rough turning, finishing turning)

5.3 Rule reuse

A new type of shaft sleeve part, a connecting rod screw, is introduced. The actual processing process is blanking, rough turning, half finishing turning, turning thread, milling, grinding, and inspection.

Firstly, we determine the processing requirements vector. The part has an external thread and no inner diameter. The outer diameter roughness and the face roughness are all 6.3 μm. After normalization processing of roughness, the processing requirements vector is [0, 1, 0, 1, 1, 1].

Secondly, the similarity between the processing requirement vector of the part and the mean vector of the three clusters is compared. The Euclidean distance method is used here to obtain three distances, namely 0, 1.73, and 1.97.

Finally, the cluster with the highest similarity is analyzed by the association rules, that is, {1, 3, 6}. Get *Rule* 1: [0, 1, 0, 1, 1, 1] → blanking, rough turning, half finishing turning, turning thread, milling, unknown, inspection.

After comparison, we can see that *Rule* 1 and its actual processing process are the same in the first five and last terms. *Rule* 2 only satisfies the first two and last items. *Rule* 3 only satisfies the first two items, so the method of matching the processing requirements vector is valid. In fact, the connecting rod screw is similar to part 6 in the cluster {1, 3, 6}, which is all bolt parts with nuts, so the processing process is highly similar to part 6.

If data preprocessing is not performed, the processing process of the nine shaft sleeve parts are directly analyzed by the association rules. At 60% confidence, we will only acquire a process design rule: *Rule*: (shaft sleeve parts) → (blanking, rough turning, finishing turning, unknown, unknown, unknown, inspection). It can be seen that the process design rule we have obtained is not accurate enough.

6 Conclusions

Process design rules are an important part of process design knowledge: access to process design rules is conducive to the realization of process design knowledge reuse and thus helps to improve the level of process design. In this paper, a method of thinking process rules extraction for process design has been proposed. Firstly, the instance representation model of process planning reflecting the thinking process of technicians has been constructed, and the related events have been extracted from the model as the basis for the extraction of process design rules. Secondly, the manifold learning algorithm and clustering analysis have been used to preprocess the process instance data. Finally, based on the association rule mining algorithm, the similar process information extraction in the same cluster has been achieved, and the complete process design rules have been formed by vectorizing the independent attribute instances in the related events.

The value of applying the proposed method to real industrial settings is that similar process information can be extracted from a large number of process instance data, and then the process design rules can be extracted in the similar process information. Through the formal description of processing requirements of parts, the process design rules are indexed and reused. However, in the actual process design, due to the diversity of parts, the processing process may be very rich, and the difference is great. The clustering method cannot effectively achieve the process clustering of similar parts. In addition, the algorithm of association rule mining is not ideal for irregular process data. Therefore, when we apply this method, we need to perform some handling of the actual processing process to make the process data more concise and tidy.

Acknowledgements The work was supported by the National Key Technology R&D Program (Grant No. 2015BAF17B01).

References

1. Kusiak A (1989) Knowledge-based systems in manufacturing. Taylor & Francis, London
2. Aalst WVD (2016) Process mining: data science in action. Springer, Berlin
3. Dorst K (2011) The core of 'design thinking' and its application. Des Stud 32(6):521–532
4. Beynon M, Curry B, Morgan P (2010) Classification and rule induction using rough set theory. Exp Syst 17(3):136–148
5. Zhang J, Li T, Pan Y (2012) Parallel rough set based knowledge acquisition using MapReduce from big data. In: International workshop on big data, streams and heterogeneous source mining: algorithms, systems, programming models and applications. pp 20–27
6. Chao Y, Sun W (2015) Welding type oriented association rules acquisition based on rough sets. Comput Eng Appl 15:244–248
7. Augasta MG, Kathirvalavakumar T (2012) Reverse engineering the neural networks for rule extraction in classification problems. Neural Process Lett 35(2):131–150
8. Masood I, Hassan A (2013) Pattern recognition for bivariate process mean shifts using feature-based artificial neural network. Int J Adv Manuf Technol 66(9–12):1201–1218
9. Khorasani AM, Yazdi MRS (2015) Development of a dynamic surface roughness monitoring system based on artificial neural networks (ANN) in milling operation. Int J Adv Manuf Technol 57(3):1–11
10. Armaghan N, Renaud J (2012) An application of multi-criteria decision aids models for case-based reasoning. Inf Sci 210(22):55–66
11. Olsson E, Funk P, Andersson A (2013) Case-based reasoning applied to geometric measurements for decision support in manufacturing. Int J Syst Assur Eng Manag 4(3):223–230
12. Zhang J, Wang F, Wang C (2016) Integrating case-based with rule-based reasoning in body-in-white fixture design. Int J Adv Manuf Technol 85(5–8):1807–1824
13. Prado RP, Garcia-Galan S, Exposito JEM et al (2010) Knowledge acquisition in fuzzy-rule-based systems with particle-swarm optimization. IEEE Trans Fuzzy Syst 18(6):1083–1097
14. Tsai T (2012) Development of a soldering quality classifier system using a hybrid data mining approach. Expert Syst Appl 39(5):5727–5738
15. Jiang Z, Jiang Y, Wang Y et al (2016) A hybrid approach of rough set and case-based reasoning to remanufacturing process planning. J Intell Manuf. https://doi.org/10.1007/s10845-016-1231-0
16. Efthymiou K, Sipsas K, Mourtzis D et al (2015) On knowledge reuse for manufacturing systems design and planning: a semantic technology approach. CIRP J Manufact Sci Technol 8:1–11
17. Chryssolouris G, Papakostas N, Mourtzis D et al (2008). Knowledge management in manufacturing process modeling: case studies in selected manufacturing processes. In: Bernard A, Tichkiewitch S (eds) Methods and tools for effective knowledge life-cycle-management. Springer, Berlin, pp 507–520
18. Obayashi S (2011) Extraction of design rules from multi-objective design exploration (MODE) using rough set theory. Fluid Dyn Res 43(4):1465–1474
19. Anantasarn N, Suriyapraphadilok U, Babi DK (2017) A computer-aided approach for achieving sustainable process design by process intensification. Comput Chem Eng 105:56–73
20. Batoulis K, Meyer A, Bazhenova E et al (2015) Extracting decision logic from process models. Lect Notes Comput Sci 9097:349–366
21. Kluza K, Nalepa GJ (2017) A method for generation and design of business processes with business rules. Inf Softw Technol 91:123–141
22. Tran L, Banerjee D, Wang J et al (2013) High-dimensional MRI data analysis using a large-scale manifold learning approach. Mach Vis Appl 24(5):995–1014
23. Mantziou E, Papadopoulos S, Kompatsiaris Y (2015) Learning to detect concepts with approximate Laplacian eigen maps in large-scale and online settings. Int J Multimed Inf Retr 4(2):95–111
24. Jain VK, Tapaswi S, Shukla A (2012) Location estimation based on semi-supervised locally linear embedding (SSLLE) approach for indoor wireless networks. Wirel Pers Commun 67(4):879–893
25. Kashniyal J, Verma S, Singh KP (2016) Wireless sensor networks localization using progressive isomap. Wirel Pers Commun 92:1–22
26. Srikant R, Agrawal R (2015) Mining generalized association rules. vldb-1995. Future Gener Comput Syst 13(2–3):161–180
27. Hu F, Wang CT, Wu YC et al (2014) Parameters selection of LLE algorithm for classification tasks. Adv Mater Res 1037: 422–427