Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A knowledge push technology based on applicable probability matching and multidimensional context driving*

Shu-you ZHANG, Ye GU, Xiao-jian LIU†‡, Jian-rong TAN
State Key Laboratory of Fluid Power & Mechatronic Systems, Zhejiang University, Hangzhou 310027, China
†E-mail: liuxj@zju.edu.cn

Received Nov. 16, 2017; Revision accepted Feb. 17, 2018; Crosschecked Feb. 19, 2018

Abstract: Actively pushing design knowledge to designers in the design process, what we call 'knowledge push', can help improve the efficiency and quality of intelligent product design. A knowledge push technology usually includes matching of related knowledge and proper pushing of matching results. Existing approaches on knowledge matching commonly have a lack of intelligence. Also, the pushing of matching results is less personalized. In this paper, we propose a knowledge push technology based on applicable probability matching and multidimensional context driving. By building a training sample set, including knowledge description vectors, case feature vectors, and the mapping Boolean matrix, two probability values, application and non-application, were calculated via a Bayesian theorem to describe the matching degree between knowledge and content. The push results were defined by the comparison between two probability values. The hierarchical design content models were built to filter the knowledge in push results. The rules of personalized knowledge push were sorted by multidimensional contexts, which include design knowledge, design context, design content, and the designer. A knowledge push system based on intellectualized design of CNC machine tools was used to confirm the feasibility of the proposed technology in engineering applications.

Key words: Product design; Knowledge push; Applicable probability matching; Multidimensional context; Personalization
https://doi.org/10.1631/FITEE.1700763 **CLC number:** TP391

1 Introduction

Knowledge push is one of the key technologies in intelligent product design, which is the future direction of manufacturing (Xu et al., 2013). Knowledge push technology actively provides designers with useful design knowledge in product design, and this can resolve knowledge flooding, knowledge trek, and other issues. The essence is "to push the right knowledge to the right person at the right time in the right way (R4)" (Schreiber, 2000). Current studies are conducted around these R4 aspects, and contain mainly three types of approaches: collaborative filtering recommendation, content-based recommendation, and mixed recommendation.

Collaborative filtering recommendation was proposed by Goldberg et al. (1992) in the experimental system 'Tapestry', and has been rapidly developed later. There are three types of collaborative filtering methods, based on user, project, and model, respectively. The user-based method focuses on the user's characteristics and pushes the items of interest (Fan et al., 2005; Xu et al., 2013). The project-based method focuses on the characteristics of a project and pushes new items with high similarity to the previous one (Zhou et al., 2009; Jiang et al., 2012). The model-based method uses machine learning and data mining algorithms to build a learning model and push knowledge (Shen et al., 2015; Feng et al., 2016). Content-based recommendation focuses on a specific

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No. 51675478), the Natural Science Foundation of Zhejiang Province, China (No. LY15E050004), and Youth Funds of the State Key Laboratory of Fluid Power & Mechatronic Systems, Zhejiang University
ORCID: Xiao-jian LIU, http://orcid.org/0000-0001-8147-9954

user, building projects and demanding models to match and push knowledge (Liu et al., 2016; Wang et al., 2016; Zhang and Li, 2016). Mixed recommendation mixes and develops several recommendation methods (Yoshii et al., 2008). According to the literature, knowledge matching is the basic goal of knowledge push. Nowadays, personalized knowledge push has attracted more attention.

Knowledge matching aims to find the design knowledge relevant to the design tasks or designer demands to push. Some methods and theories have been proposed to improve the matching accuracy. The most common method is similarity calculation between the design tasks and knowledge, including similarity of text (Wang SF et al., 2007; Wang S et al., 2009; Yan et al., 2016) and semantic distance (Xu et al., 2013). Mao et al. (2012) combined these two methods for knowledge matching. Furthermore, Wang et al. (2016) proposed three different types of semantic similarity calculation with function-behavior-structure (FBS). Shen et al. (2015) calculated the characteristic similarity between function and structure by conditional probability. Some researchers built the mapping relation between cases and design knowledge, and compared the similarity between the design task and cases to obtain the design knowledge to push (Zhou et al., 2009; Jiang et al., 2012). Similarly, Xu et al. (2016) used frequent sequential pattern technology to mine the historical knowledge usage data, and finished knowledge matching and pushing according to the current knowledge sequence. There are some other methods to match knowledge without calculating similarities. For example, knowledge push rules were extracted from variable precision rough sets (Ji et al., 2013), and the matching knowledge can be searched in the ontology model (Wang et al., 2015). Knowledge can be matched through hierarchical spreading activation (Liang et al., 2015) or knowledge clustering collection (Li et al., 2017) in knowledge complex network theory. Dong et al. (2013) simulated the immune process, and proposed a knowledge module-artificial immune algorithm (KM-AIA) combined algorithm and entropy method to match the knowledge.

Personalized knowledge push is more challenging. Different designers may need different knowledge in the design process. In the past few years, researchers have paid more attention to personalized knowledge push. The simpler methods are secondary filtering for knowledge streamlined (Feng et al., 2016) and sorted by importance. Ji et al. (2013) calculated designer's familiarity with knowledge by forgetting curves. Li et al. (2017) proposed an acquaintance immune strategy to guarantee the novelty of the product design knowledge in the subsequent push link. Researchers usually built designer capability models (Feng et al., 2016; Jiang et al., 2017), personalized interest models (Zhi et al., 2011), or compound knowledge frameworks (Fan et al., 2005) for personalized knowledge push. Intent capture is an effective way for personalization (Xu et al., 2013; Wang et al., 2016). Chen et al. (2015) used an acoustic energy feature and creative segment theory to capture intent and demands during sketching. With the development of the Internet, big data collection and mining can help analyze the designer's behavior in the web and the knowledge usage for personalized knowledge push (Li et al., 2015).

The knowledge matching methods, commonly using similarity calculation, set the filtering threshold according to experiments and manual intervention. Recent studies have focused on how to reduce human factors and improve intelligence. This is also the innovation of this study. Further, researchers have proposed various methods for personalized knowledge push. Some advanced methods, such as design intent capture, interest acquisition, are in the early stages of research and are not mature. The methods of secondary filtering and sorting have not been further developed. This study improves the relevant theory of sorting used in practical engineering for personalized knowledge push.

In current studies, a new method of knowledge push technology based on applicable probability matching and multidimensional context driving is proposed to solve the two abovementioned problems. As shown in Fig. 1, the design tasks are first imported in the knowledge push system, and then different designers log into the system. Combining design content with the knowledge center, the applicable probability matching method serves knowledge matching by using probability theory. Next, the matched knowledge needs to be processed for personalization. The methods of personalized knowledge push are sorted by multidimensional contexts, including design knowledge, design context, and the designer. Finally, the highly demanding knowledge can be ranked first in the push queue.

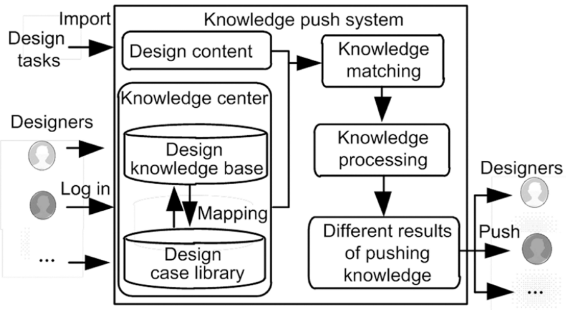

Fig. 1 The framework of the knowledge push system

2 A training set in knowledge and case

The two characters in the traditional design knowledge base, classification and application, are expanded for personalized knowledge push. The knowledge is described as a knowledge description vector. The design cases in the product case library are described as case feature vectors in the form of 'feature-weight'. Then, the training set is built using a set of knowledge description and case feature vectors.

Definition 1 (Knowledge description vector) The knowledge description vector is **doc**={content, classification, application}, where 'content' denotes knowledge details, 'classification' denotes knowledge species, including existing text, formula, chart, drawing, and case, and 'application' is a set that stores the last use time by pushing system logs, and this character can be used to push personalized knowledge later.

Definition 2 (Case feature vector) The case feature vector is **pro**={(key$_1$, s_1), (key$_2$, s_2), …, (key$_n$, s_n)}, where key$_k$ (k=1, 2, …, n) is the k^{th} case feature, s_k is the k^{th} feature weight, and n is the total number of case features. The key in **pro** is extracted by the semantic relation of natural language segmentation (Wang XJ et al., 2007), and s is calculated by the feature position vector $\boldsymbol{\Omega}$ and term frequency–inverse document frequency (TF-IDF) (Wu et al., 2008).

Definition 3 (Feature position vector) The feature position vector is $\boldsymbol{\Omega}$={(position, weight)}, and $\boldsymbol{\Omega}$={(design task, 1), (design context, 0.9), (design specification, 0.8), (designer, 0.7)}.

The feature weight s_k is expressed as

$$s_k=\frac{\left(\sum_{m=1}^{4}\delta_m \mathrm{tf}_{km}\right)\cdot \mathrm{idf}_k}{2+\left(\sum_{m=1}^{4}\delta_m \mathrm{tf}_{km}\right)\cdot \mathrm{idf}_k}. \tag{1}$$

As shown in Eq. (1), m is the position in $\boldsymbol{\Omega}$, δ_m is the weight at position m, tf$_{km}$ is the k^{th} feature term frequency at position m, and idf$_k$ is the k^{th} feature inverse document frequency.

Definition 4 (Design knowledge base) The design knowledge base is defined as $\boldsymbol{\gamma}$={doc$_1$, doc$_2$, …, doc$_N$} and the design case library as $\boldsymbol{\chi}$={pro$_1$, pro$_2$, …, pro$_M$}, where N and M are the number of pieces of design knowledge and design cases, respectively. Both N and M change when $\boldsymbol{\gamma}$ and $\boldsymbol{\chi}$ are updated. The mapping between $\boldsymbol{\gamma}$ and $\boldsymbol{\chi}$ can generate a Boolean matrix $\boldsymbol{A}$, and the training set is shown in Fig. 2.

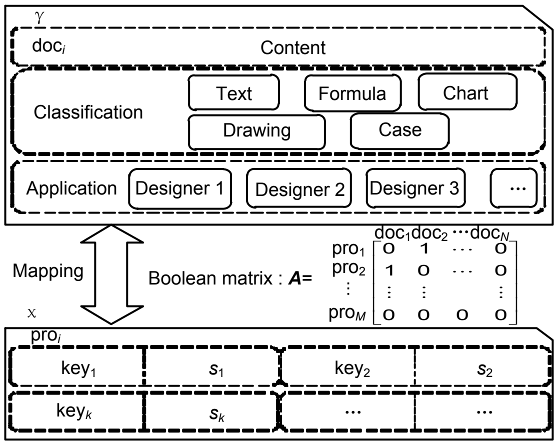

Fig. 2 Generating the training set

In Boolean matrix $\boldsymbol{A}$, 1 represents the knowledge applied to the case, and 0 is non-application. A knowledge description vector **doc** can divide $\boldsymbol{\chi}$ into two categories, i.e., application and non-application. Therefore, the cases in $\boldsymbol{\chi}$ can be divided repeatedly into 2N categories. The training data set is shown in Table 1, in which 'feature' is the basis for classification in **pro**, **doc** is the category, and s should be set to 0 if there is no key in **pro** correspondingly.

Table 1 Training data set

	Feature					Category			
	key$_1$	key$_2$	key$_3$	key$_4$	…	doc$_1$	doc$_2$	doc$_3$	…
pro$_1$	s_{11}	s_{12}	s_{13}	s_{41}	…	0	1	0	…
pro$_2$	s_{21}	s_{22}	s_{23}	s_{42}	…	1	0	1	…
pro$_3$	s_{31}	s_{32}	s_{33}	s_{43}	…	1	0	0	…
pro$_4$	s_{41}	s_{42}	s_{43}	s_{44}	…	0	0	0	…
…	…	…	…	…	…	…	…	…	…

3 Applicable probability knowledge matching method

3.1 Hierarchical design content modeling

The design content in the product design process is described as a design feature vector. In particular, this vector has impact on the push results. The different abilities in the design process lead to the different knowledge requirements for different designers; thus, a hierarchical design content model is built for personalized knowledge push. The design department will prejudge the designer's ability when assigning the design task, and the ability is divided into three categories: skilled, general, and rusty.

Definition 5 (Design feature vector) The design feature vector is **content**={(word$_1$, q_1), (word$_2$, q_2), …, (word$_t$, q_t)}, where word$_j$ (j=1, 2, …, t) is the j^{th} content feature, q_j is the j^{th} feature weight, and t is the total number of content features.

For different designers with the same design content, features are constant and weights are changed. The rusty designers need more knowledge, and thus the values of low weight features should be raised. Conversely, the values of high weight features are reduced for skilled designers. The weights for general designers are unchanged, and the feature weight q is expressed by Eqs. (2)–(4):

$$q_{\mathrm{s}}=\begin{cases} q_{\mathrm{g}}\eta_{\mathrm{s}}, & q_{\mathrm{g}}>\alpha_{\mathrm{s}}, \\ q_{\mathrm{g}}, & q_{\mathrm{g}}\le\alpha_{\mathrm{s}}, \end{cases} \tag{2}$$

$$q_{\mathrm{g}}=\frac{\left(\sum_{m=1}^{4}\delta_m \mathrm{tf}_{jm}\right)\cdot \mathrm{idf}_j}{2+\left(\sum_{m=1}^{4}\delta_m \mathrm{tf}_{jm}\right)\cdot \mathrm{idf}_j}, \tag{3}$$

$$q_{\mathrm{r}}=\begin{cases} q_{\mathrm{g}}\eta_{\mathrm{r}}, & q_{\mathrm{g}}<\alpha_{\mathrm{r}}, \\ q_{\mathrm{g}}, & q_{\mathrm{g}}\ge\alpha_{\mathrm{r}}, \end{cases} \tag{4}$$

where q_{s}, q_{g}, and q_{r} denote skilled, general, and rusty weights, respectively. q_{g} is similar to s_k in Eq. (1), η_{s}, η_{r} are variable-weight coefficients, and α_{s}, α_{r} are variable-weight intervals. The variable-weight design feature vector can filter the push results properly, but the parameters need to be determined by repeated experiments.

3.2 Applicable probability matching

In this study, knowledge matching is performed using the method of applicable probability matching. Two probability values, i.e., application and non-application, are calculated to describe the matching degree between knowledge and content. $\lambda_{\mathrm{c}i}$ is the matching result and is expressed as

$$\lambda_{\mathrm{c}i}=\begin{cases} 1, & \text{application}, \\ 0, & \text{non-application}, \end{cases} \quad i=1,2,\cdots,N. \tag{5}$$

If the applicable probability $P(\lambda_{\mathrm{c}i}=1|\textbf{content})$ is larger than the non-application probability $P(\lambda_{\mathrm{c}i}=0|\textbf{content})$, $\lambda_{\mathrm{c}i}=1$ and doc$_i$ applies to **content**, and the knowledge set {doc$_i$|$\lambda_{\mathrm{c}i}$=1} shows the push results in this design content.

The Naive Bayes classifier (Friedman et al., 1997) is adopted to calculate the applicable probability based on training sets. There are N knowledge description vectors to be matched with the design feature vector **content**, and it is necessary to classify the **content** N times. The classification result is described by Eq. (5), and defined as ε in the following.

Using the Bayesian theorem the posterior probability can be transformed into an a priori probability:

$$P(\varepsilon|\textbf{content})=\frac{P(\varepsilon)P(\textbf{content}|\varepsilon)}{P(\textbf{content})}, \tag{6}$$

where $P(\varepsilon)$ is the a priori probability, $P(\textbf{content}|\varepsilon)$ is a class-conditional probability, and $P(\textbf{content})$ is the normalized evidence constant (Friedman et al., 1997).

$$P(\varepsilon)=|\boldsymbol{\chi}_\varepsilon|/|\boldsymbol{\chi}|, \tag{7}$$

where $\boldsymbol{\chi}_\varepsilon$ is the set of ε cases in $\boldsymbol{\chi}$, and $|\cdot|$ means the number of elements in the set.

$P(\textbf{content}|\varepsilon)$ is defined in Eq. (8), if features are independent of one another. $q_{\mathrm{c}j}$ is the j^{th} feature weight in **content**, d is the total number of features in the training data set, and $P(q_{\mathrm{c}j}|\varepsilon)$ is described in Eq. (9).

$$P(\textbf{content}|\varepsilon)=\prod_{j=1}^{d}P(q_{\mathrm{c}j}|\varepsilon), \tag{8}$$

$$P(q_{\mathrm{c}j}|\varepsilon)=\frac{1}{\sqrt{2\pi}\sigma_{\varepsilon,\mathrm{c}j}}\exp\left(-\frac{(q_{\mathrm{c}j}-\mu_{\varepsilon,\mathrm{c}j})^2}{2\sigma_{\varepsilon,\mathrm{c}j}^2}\right). \tag{9}$$

As shown in Eq. (9), the feature weight is calculated using a probability density function, where $\mu_{\varepsilon,\mathrm{c}j}$ is the weight mean and $\sigma_{\varepsilon,\mathrm{c}j}^2$ is the weight variance in the ε cases set. $P(q_{cj}|\varepsilon)$ is calculated in the training set.

According to Eqs. (6) and (8), the matching result in Eq. (5) is defined by comparing $P(\lambda_{ci}=1)\cdot\prod_{j=1}^{d}P(q_{cj}|\lambda_{ci}=1)$ with $P(\lambda_{ci}=0)\prod_{j=1}^{d}P(q_{cj}|\lambda_{ci}=0)$, where $P(\varepsilon)\prod_{j=1}^{d}P(q_{cj}|\varepsilon)$ is the applicable probability relative value.

4 Sorting personalized push results driven by multidimensional context

4.1 Multidimensional context-driven sorting rules

The multidimensional context decides the final sorting rules from the four dimensions, including design knowledge, design context, design content, and the designer. The following four requirements should be met:

1. The sorting method is based on design knowledge. The knowledge in the push queue is divided into five categories based on the classification in the knowledge description vector **doc**.

2. The sorting method based on design context is influenced by different contexts in product collaborative design. In the scheme design stage, the drawing knowledge and case knowledge are placed in the head of the push queue. In the technical design stage, text knowledge, chart knowledge, and formula knowledge are placed in the head of the push queue. Each design task has its matching context and is related to each other. As shown in Fig. 3, the inheritance and feedback information of the design content result is defined as τ, which should be noted and is the feedback in real time.

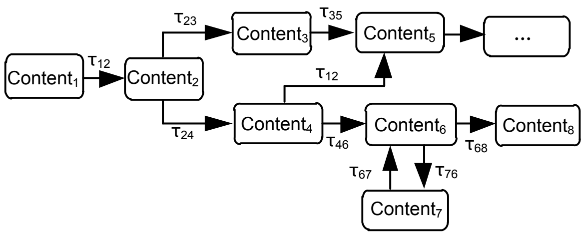

Fig. 3 Flow diagram in design context

3. The sorting based on design content is carried out according to the design specification in the various stages of product design. If there are keywords in the design specification, such as ‘calculate’, ‘check’, and ‘formula’, formula knowledge will be sorted before chart knowledge. If the keywords are ‘compare’, ‘inquire’, and ‘select’, chart knowledge will be sorted before formula knowledge.

4. The sorting method based on the designer is divided into two parts, explicit behavior and implicit behavior. Designers with different abilities would have different knowledge requirements in the same design process, which is the most uncontrollable factor, and thus personalized knowledge push requires much work and effort. This sorting method is based on designer’s abilities and knowledge application.

Explicit behavior is defined as the designer participation in old cases. We compare the designer who completes a drawing or has case knowledge in the push results with the designer in the current context. If they are the same, this knowledge will be put in the back of the push queue; otherwise, this knowledge has precedence and will be put in the front.

Implicit behavior is defined as a designer’s knowledge, adopting the Ebbinghaus forgetting curve (Dong et al., 2017). Through the application in **doc**, the familiarity with design knowledge is calculated among all designers in Eq. (10), and the knowledge in the push results will be sorted second in categories.

$$W_t = 68.2\exp(-1.068\Delta T) + 31.8\exp(-2.88\times10^{-4}\Delta T), \tag{10}$$

where W_t is the familiarity (the maximum value is 100), and ΔT is the time interval in days.

4.2 Workflow of design knowledge push

Fig. 4 shows the workflow of knowledge push based on applicable probability matching and multidimensional context driving. It consists of the following four stages:

1. Knowledge and case models are built and the training set is generated;

2. Knowledge is matched according to the applicable probability, and the results to be pushed are obtained;

3. The multidimensional contexts including design knowledge, design context, design content, and the designer affect the sorting rules for personalized knowledge push;

4. Knowledge push is completed, and the data including the training set and the application in **doc** is updated.

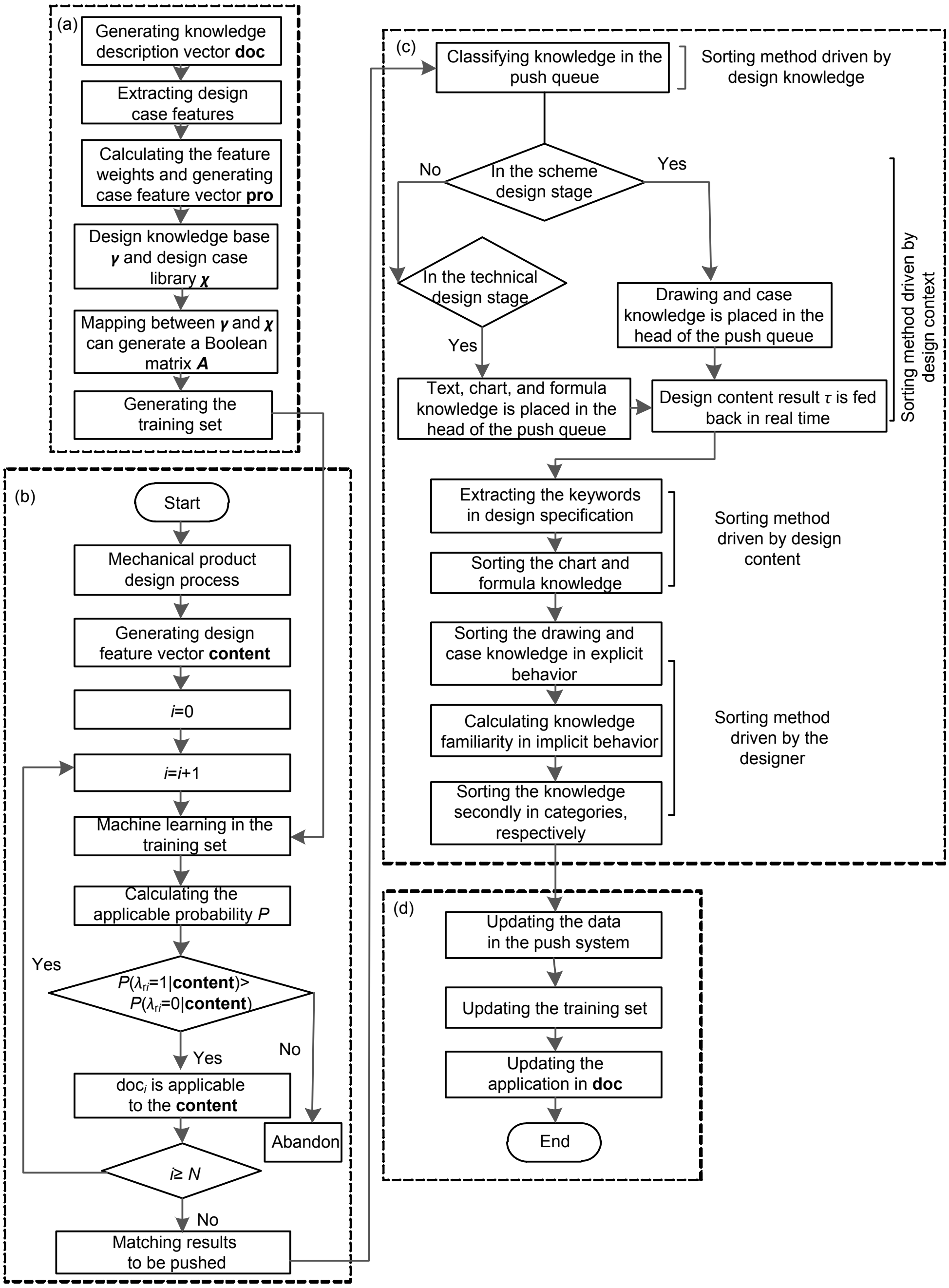

Fig. 4 Flow diagram for design knowledge push: (a) generating the training set; (b) knowledge matching; (c) personalized knowledge push; (d) completing and updating

5 An illustrative example

A knowledge push system based on applicable probability matching and multidimensional context driving was developed and used in the intellectualized design of CNC machine tools. As it is typical in complex mechanical equipment, there is a lot of design knowledge in designing CNC machine tools. The horizontal lathe guides are the main parts of the machine. These are taken as an example to evaluate our proposed method in the design platform of CNC machine tools.

Fig. 5 shows the graphical interface of the CNC machine tools' design platform. This applies to Chinese CNC machine tools enterprises. 'User info' shown on the top left of the interface is the current login designer's information, including ID, position, field, and ability. 'Knowledge choice' is shown on the bottom left of the interface. We search the design resources by the keywords of the horizontal lathe guides, and the search results are shown at the right of the interface. There are 86 pieces of design knowledge and 35 design cases in the design resources related to design tasks. The characters of design resources include 'No.', 'Title', 'Classification', 'Belonging', 'Designer', and 'Last use time'. Then a 35×86 Boolean matrix $\boldsymbol{A}$ and a training set are generated. The design feature vector is **content**={(Horizontal, 0.5163), (Precision, 0.6172), (Function, 0.4198), (Sliding, 0.8143), (Rectilinear, 0.7061), (Cast-iron, 0.8423), (Open, 0.8052), (Feed, 0.8446), (Crawl, 0.5432), (Dovetail, 0.9168), (Gap, 0.6578), (Unloading, 0.4803), (Processing, 0.2578), (Hydraulic, 0.0842)}.

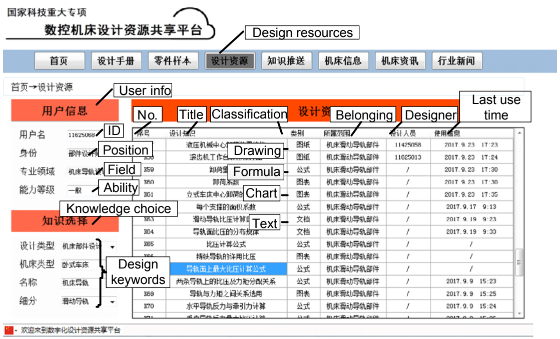

Fig. 5 Design resources for the horizontal lathe guides

The knowledge matching result (Fig. 6) is calculated using the training set and applicable probability matching method. The points in Fig. 6 show the larger value in $P(\lambda_{ci}=1)$ and $P(\lambda_{ci}=0)$. The 28 red points denote $P(\lambda_{ci}=1|\mathbf{content})>P(\lambda_{ci}=0|\mathbf{content})$, which means the applicable knowledge. The green points represent the non-application knowledge.

The training set consists of 35 design cases. The number of related cases should be large enough to ensure an accurate matching. Fig. 7 shows the comparison of matching results with the training sets consisting of 5, 15, and 35 training samples. Dashed lines highlight the results for 35 training samples. The matching results for 15 training samples include knowledge such as K26, K36, K54, and K57 but do not include knowledge K70 and K77, compared with those for 35 training samples. For example, K26 is 'vertical lathe guide size', K36 is 'flat head guide gap adjustment', and K54 is 'mechanical unloading device'. They are irrelevant to the design content. K70 is 'calculation of reaction force and traction force' and K77 is 'case of pressure calculation in dovetail guide'. The lost knowledge is obviously useful to the design process. Similarly, the result for 5 training samples is also less accurate.

The design knowledge results in the above sections are suitable for the general designers. The hierarchical design content is modeled by Eqs. (2)–(4). The experiment is designed by 100 sets of random design feature vectors in 1×10^6 Monte Carlo steps (MCSs), and the results are averaged over 10 independent realizations. The evolution of knowledge matching results is determined in the stationary state after a sufficiently long relaxation lasting up to 2×10^7 MCSs. As shown in Fig. 8, (η_s, α_s) and (η_r, α_r) are set as (0.6, 0.5) and (1.15, 0.2) independently.

Substituting the determined values (η_s, α_s) and (η_r, α_r) into Eqs. (2)–(4), Fig. 9 shows the comparison of matching results for designers with different abilities. It shows that skilled designers reduce knowledge like K17, K24, K81, and K83, and rusty designers increase knowledge like K10 and K15. It shows that the design knowledge matching result can be filtered by distinguishing the abilities of designers and changing the feature weights in **content**.

Fig. 10 shows the knowledge push interface of the design platform. At the left of the interface are 'User info' and 'Design content', and at the right are the last push results of sliding guides pressure calculation by two different designers. By analyzing the push results, knowledge K81 and K83 is not necessary in the push results for the skilled designer 11 325 045, compared with the general designer 11 625 068. The orders of knowledge in the push results are also different. K77 and K78 are the case knowledge and are completed by designer 11 325 045. Ultimately, the knowledge in the first row is K70, called 'calculation of reaction force and traction force' for designer 11 325 045, and K77 called 'case of pressure calculation in dovetail guide' is the first knowledge for designer 11 625 068.

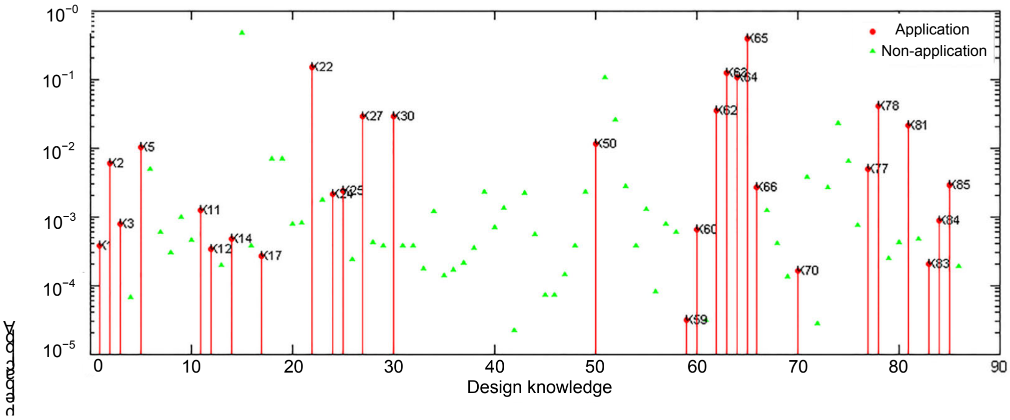

Fig. 6 Results for design knowledge matching (References to color refer to the online version of this figure)

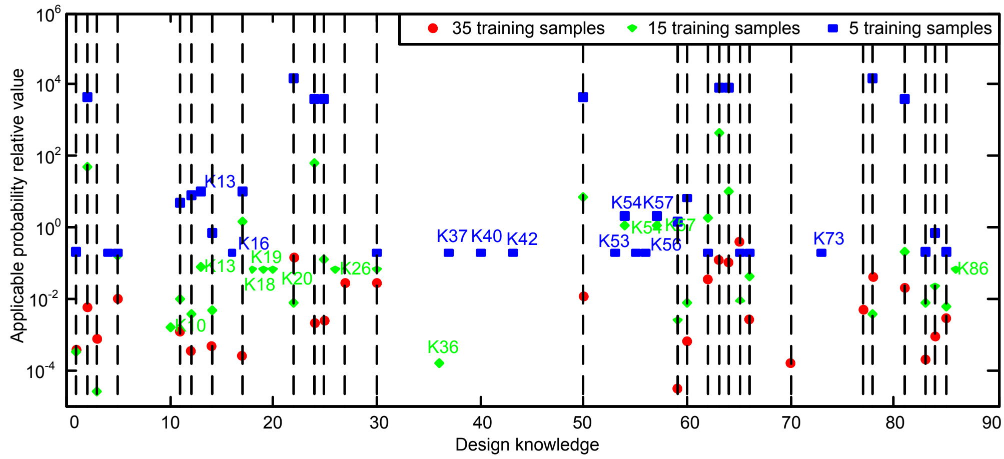

Fig. 7 Results for design knowledge matching in different training sets (References to color refer to the online version of this figure)

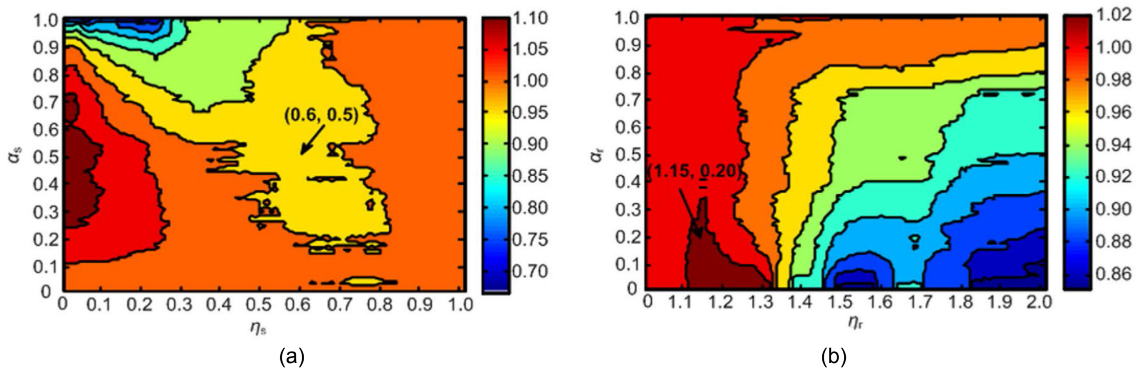

Fig. 8 Evolution of design knowledge matching results: (a) variable-weight coefficients η_s and intervals α_s for skilled designers; (b) variable-weight coefficients η_r and intervals α_r for rusty designers (References to color refer to the online version of this figure)

The results show that the design knowledge matched by applicable probability can avoid manual setting of the filter threshold, which makes up for the shortcomings of similarity calculation in previous studies. Furthermore, some researchers (Zhou et al., 2009; Jiang et al., 2012) found design cases similar to the design content from the knowledge center to obtain the push results. However, if there is no design case similar to the design content, knowledge matching will not be realized. The applicable probability

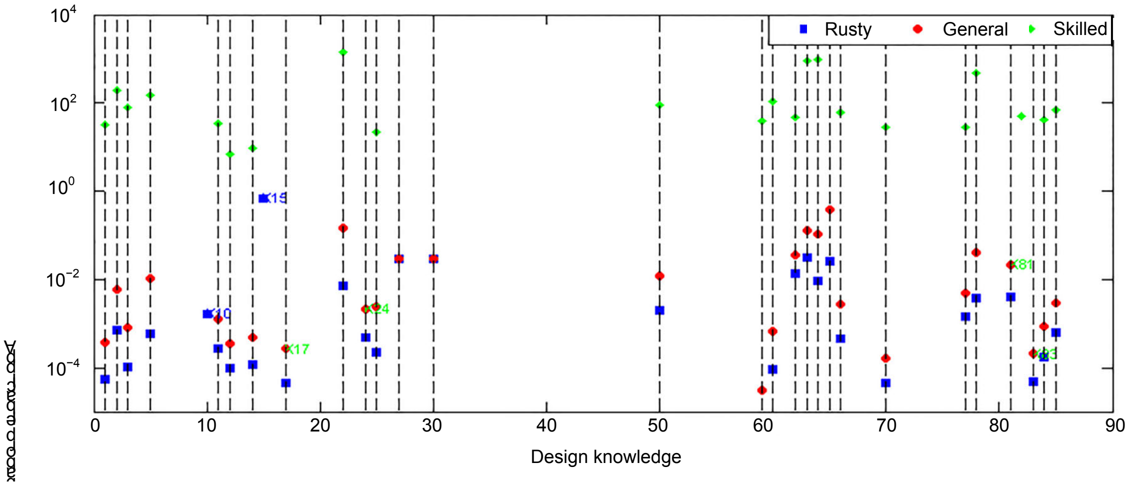

Fig. 9 Results of design knowledge matching for designers with different abilities (References to color refer to the online version of this figure)

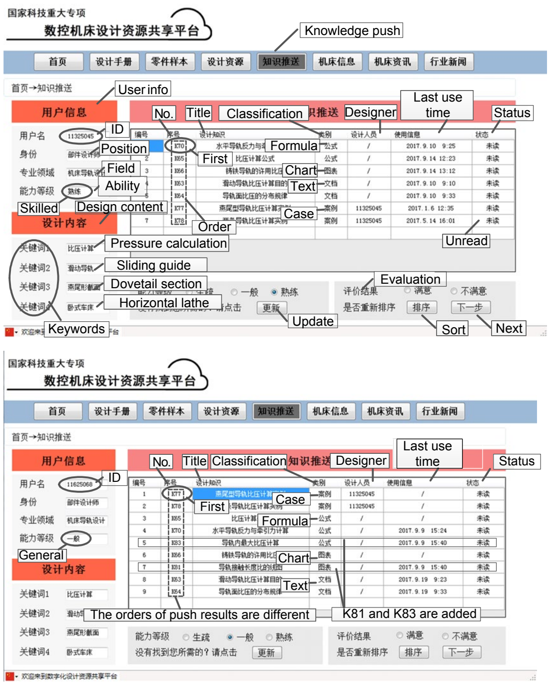

Fig. 10 Push results for two different designers

matching method proposed here can provide accurate matching results even for unknown design content by using a training set and the Bayesian theorem.

The hierarchical design content models and multidimensional context-driven sorting rules are used to serve the personalized knowledge push in this study.

A group of designers made a positive evaluation of the knowledge push system in a machine tool design test. Designers could benefit from the knowledge in the head of the push queue, which can mostly meet their own demands, and correspondingly they can find the demanding knowledge more quickly. Sorting theory is a simple way to achieve personalization. In this study, we developed and consummated the sorting theory compared with previous studies (Liang et al., 2015; Feng et al., 2016) to make the results more personalized. Some advanced methods, such as intent capture, electroencephalogram (EEG) recognition, and behavior modeling, are still in the early stages of research. In the current stage, the method proposed in this study can be better applied in engineering. Different designers can shorten the time to browse knowledge and improve design efficiency.

6 Conclusions

In this paper, a knowledge push technology based on applicable probability matching and multidimensional context driving was proposed to improve the efficiency and quality of knowledge push:

1. We proposed the applicable probability matching method in knowledge matching. Two probability values, i.e., application and non-application, were calculated using a Bayesian theorem to describe the matching degree between knowledge and content. Different from traditional knowledge matching methods, this method can set the filtering threshold automatically.

2. We proposed hierarchical design content models and multidimensional context-driven sorting rules to serve personalized knowledge push. The hierarchical design content models overcame the deficiency that a traditional model cannot meet different designers' requirements. The multidimensional context driven sorting of the push results ensures that designers can find the highly demanded knowledge as soon as possible.

3. We developed a knowledge push system based on the intellectualized design of CNC machine tools and provided an illustrative example on the design of horizontal lathe guides. Experimental results showed that the push results are accurate, and that the system can provide a personalized knowledge push service.

Our proposed knowledge push technology can solve the problems of knowledge matching and personalized knowledge push, but there are some aspects that need to be improved. In hierarchical design content models, the parameter values in Eqs. (2)–(4) are difficult to set; once they are chosen, they cannot guarantee uniformity in different design contexts. This may affect the accuracy and satisfaction in personalized knowledge push. The refinement and personalization improvement of push results will be the focus of our future work.

References

Chen S, Yang ZY, Sun LY, et al., 2015. Research on design knowledge analytical method during sketching-combining acoustic energy feature and creative segment theory. *J Zhejiang Univ Eng Sci*, 49(9):2073-2082 (in Chinese). https://doi.org/10.3785/j.issn.1008-973X.2015.09.007

Dong LY, Wang YQ, He JN, et al., 2017. Collaborative filtering recommendation algorithm based on time decay. *J Jilin Univ Eng Technol Ed*, 47(4):1268-1272 (in Chinese). https://doi.org/10.13229/j.cnki.jdxbgxb201704036

Dong SY, Xu JX, Wang KQ, et al., 2013. Active push model of manufacturing process knowledge in CAD platform based on immune process. *Comput Integr Manuf Syst*, 19(7):1520-1531 (in Chinese). https://doi.org/10.13196/j.cims.2013.07.82.dongsy.016

Fan ZP, Feng Y, Sun YH, et al., 2005. A framework on compound knowledge push system oriented to organizational employees. Proc 1st Int Workshop on Internet and Network Economics, p.622-630. https://doi.org/10.1007/11600930_62

Feng YX, Zhang SY, Gao YC, et al., 2016. Intelligent push method of CNC design knowledge based on feature semantic analysis. *Comput Integr Manuf Syst*, 22(1):189-201 (in Chinese). https://doi.org/10.13196/j.cims.2016.01.018

Friedman N, Geiger D, Goldszmidt M, 1997. Bayesian network classifiers. *Mach Learn*, 29(2-3):131-163. https://doi.org/10.1002/9780470400531.eorms0099

Goldberg D, Nichols D, Oki BM, et al., 1992. Using collaborative filtering to weave an information tapestry. *Commun ACM*, 35(12):61-70. https://doi.org/10.1145/138859.138867

Ji X, Gu XJ, Dai F, et al., 2013. Technology for product design knowledge push based on ontology and rough sets. *Comput Integr Manuf Syst*, 19(1):7-20 (in Chinese). https://doi.org/10.13196/j.cims.2013.01.9.jix.008

Jiang CQ, Li BS, Lu WX, 2012. Research on knowledge push for mechanical product collaborative design based on case library. *Mech Des Manuf*, (1):257-259 (in Chinese). https://doi.org/10.3969/j.issn.1001-3997.2012.01.096

Jiang H, Yin P, Guo L, et al., 2017. Knowledge push based on design flow and user capacity model. MATEC Web Conf, Article 12. https://doi.org/10.1051/matecconf/201713900012

Li C, Li WQ, Li Y, et al., 2015. Research and application of knowledge resources network for product innovation. *Sci World J*, Article 495 309. https://doi.org/10.1155/2015/495309

Li XR, Yu SH, Chu JJ, et al., 2017. Double push strategy of knowledge for product design based on complex network theory. *Discr Dynam Nat Soc*, Article 2 078 626. https://doi.org/10.1155/2017/2078626

Liang Y, Zhang SY, Liu XJ, et al., 2015. Product design knowledge dynamic push technology based on variable-weight layered spreading activation model. *Comput Integr Manuf Syst*, 21(12):3107-3118 (in Chinese). https://doi.org/10.13196/j.cims.2015.12.002

Liu TY, Wang HF, He Y, 2016. Intelligent knowledge recommending approach for new product development based on workflow context matching. *Concurr Eng*, 24(4): 318-329. https://doi.org/10.1177/1063293X16640319

Mao J, Cao NL, Cao YL, et al., 2012. Matching method for quality knowledge in product designing process. *Trans Chin Soc Agric Mach*, 43(1):197-201 (in Chinese). https://doi.org/10.6041/j.issn.1000-1298.2012.01.035

Schreiber G, 2000. Knowledge Engineering and Management: the Common KADS Methodology. MIT Press, Cambridge, MA.

Shen MY, Qiu LM, Tan JR, et al., 2015. Active push design of product subdivision structure driven by performance requirement. *J Zhejiang Univ Eng Sci*, 49(2):287-295 (in Chinese). https://doi.org/10.3785/j.issn.1008-973X.2015.02.014

Wang FL, Liao WH, Guo Y, et al., 2015. Reduction and push technology of cable harness information for complex mechatronic products based on variable precision rough sets. Proc 5th Int Conf on Simulation and Modeling Methodologies, Technologies and Applications, p.263-270. https://doi.org/10.5220/0005538002630270

Wang S, Yin GF, He ZX, 2009. Active push technology for multidisciplinary auxiliary knowledge in product design.

Proc Int Conf on Technology and Innovation, p.1-5. https://doi.org/10.1049/cp.2009.1511

Wang SF, Gu XJ, Guo JF, et al., 2007. Knowledge active push for product design. *Comput Integr Manuf Syst*, 13(2): 234-239 (in Chinese). https://doi.org/10.3969/j.issn.1006-5911.2007.02.005

Wang XJ, Qin Y, Liu W, 2007. A search-based Chinese word segmentation method. Proc 16th Int Conf on World Wide Web, p.1129-1130. https://doi.org/10.1145/1242572.1242729

Wang ZS, Tian L, Wu YH, et al., 2016. Personalized knowledge push system based on design intent and user interest. *Proc Inst Mech Eng Part C J Mech Eng Sci*, 230(11):1757-1772. https://doi.org/10.1177/0954406215584395

Wu HC, Luk RWP, Wong KF, et al., 2008. Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans Inform Syst*, 26(3), Article 13. https://doi.org/10.1145/1361684.1361686

Xu RZ, Gao Q, Wang H, et al., 2016. Product design knowledge recommendation based on sequential pattern mining. *Comput Integr Manuf Syst*, 22(5):1179-1186. https://doi.org/10.13196/j.cims.2016.05.002

Xu YH, Yin GF, Nie Y, et al., 2013. Research on an active knowledge push service based on collaborative intent capture. *J Netw Comput Appl*, 36(6):1418-1430. https://doi.org/10.1016/j.jnca.2013.04.010

Yan Y, Yang N, Hao J, et al., 2016. A context modeling method of knowledge recommendation for designers. Proc Int Conf on Information System and Artificial Intelligence, p.492-496. https://doi.org/10.1109/ISAI.2016.0111

Yoshii K, Goto M, Komatani K, et al., 2008. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *IEEE Trans Audio Speech Lang Process*, 16(2):435-447. https://doi.org/10.1109/TASL.2007.911503

Zhang FP, Li L, 2016. Research on knowledge push method for business process based on multidimensional hierarchical context model. Proc IEEE Int Conf on Industrial Engineering and Engineering Management, p.656-660. https://doi.org/10.1109/IEEM.2016.7797957

Zhi ZL, Yuan Y, Yan ZG, et al., 2011. Knowledge active push based on personalized interest model in aircraft structure design. Proc Int Conf on E-Business and E-Government, p.1-4. https://doi.org/10.1109/ICEBEG.2011.5877024

Zhou LZ, Liu DF, Wang B, et al., 2009. Research on knowledge active push model for product development. Proc Int Conf on Networking and Digital Society, p.217-220. https://doi.org/10.1109/ICNDS.2009.134