

The Study of Enterprise Personalized Knowledge Service Based on the Knowledge Scenario

Min Zhuang^{1, 2, a*}, Shunxin Li^{1, 2, b} and Jiawei Song^{1, 2}

¹College of Computer Science & Technology, Wuhan University of Science & Technology, China

²Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, China

^amihawkzhuang@126.com, ^blishunxin72@163.com

Keywords: Knowledge management; The personalized recommendation technology; The knowledge scenario

Abstract. In recent years, more and more enterprises establish the knowledge management system to share knowledge within a company, track projects and improve work efficiency. However, the increasingly complex content affects the efficiency of gain and share the knowledge. The article investigates the personalized recommendation technology and introduces the recommendation technology based on context-aware into the knowledge management system. Through the knowledge scenario such as the context of domain, goal and so on, increase the accuracy of the knowledge recommendation. Accordingly, the knowledge sharing in enterprises become more and more convenient, and the knowledge transfer become more efficient.

Introduction

While the continuous development of enterprise information, the knowledge for enterprise is becoming increasingly important. At present, the knowledge management encounters the following problem: how to find the knowledge which the staff need in short time.

Faced with this problem, personalized knowledge service emerges. The way to implement personalized knowledge service is that, through getting the feature information from interacting with users directly or tracking a user's behavior and then the feature information of user and inference rules is applied to filter the knowledge/information provided to the user. Finally, turn the filtered knowledge into a user terminal acceptable way including personalized recommendation and searching. The paper focuses on the personalized knowledge recommendation. We know only the knowledge is in the particular background and situation, it is valuable and effective. So we introduce the knowledge scenario into personalized knowledge recommendation and make effective recommendations based on the knowledge scenario.

Related Work

Personalized recommendation can't be given without related technical support. At present, the personalized recommendation technology based on algorithm to classify generally includes the following two kinds: one is recommendation based on the content [1, 2]; the other is based on collaborative filtering [3, 4, 5].

Recommendation based on content is mainly through the analysis the content of the resources which users have access to, so as to extract the interests of users based on the content. And then analyzing the content of the proposed projects, thus we can offer users the items with more features and similarity to users' interests. The defect of content-based filtering algorithm is poor flexibility, unable to find potential users' information demand and adjust in time which leads to users' less trust and satisfaction with recommendation [6, 7, 8].

Based on collaborative filtering recommendation algorithm is widely used at present, the core idea of which is based on the analysis of users' rating matrix to categorize the users with similar interests and the projects with strong correlation, then recommending the projects selected by the

user in the same category to other users in the same category, or recommending projects which have strong correlation with historical choices. Although collaborative filtering algorithm can overcome the problems and defects existing in the filtering algorithm based on content, and has applied in practice in many practical areas, with the expansion of scale of the users and recommendation projects, traditional collaborative filtering algorithms gradually exposes the problems such as sparseness and cold start. [9, 10, 11, 12]

Context Awareness technology, first put forward by Schilit in 1994, derives from ubiquitous computing research. At present, the Context Awareness has been widely used in E-commerce, film and music recommendation, E-tourism and other fields but is still rare in enterprise inner knowledge management. Nadoveza [13] put forward the context ontology for enterprise application, realizing the method of providing the required information according to the users' context. Although the study above integrates situational factors into knowledge recommendation, but not consider the influence of different situational factors on the recommendation, sometimes too many situational factors can reduce recommendation quality.

This article, focusing on the deficiencies of current personalized recommendation technology of enterprise knowledge, introduces context awareness to the personalized recommendation system of enterprise knowledge, and through the efficient methods to determine effective scenario elements, calculate situational similarity, which can recommend appropriate knowledge items to the user.

The Improved Method of Knowledge Recommendation

Individuation knowledge recommendation service systems based on the context similarity require integrate the scenario of knowledge with the knowledge item. We build the knowledge model which collaborated with the situation. And then get the historical scenario similar with the current scenario of user from the KAS (Knowledge Application Scenario) Library through similarity measuring methods of the scenario. Finally, the system recommends the knowledge which related to the most relevant historical scenario to the user. The process of recommend is shown as Fig. 1:

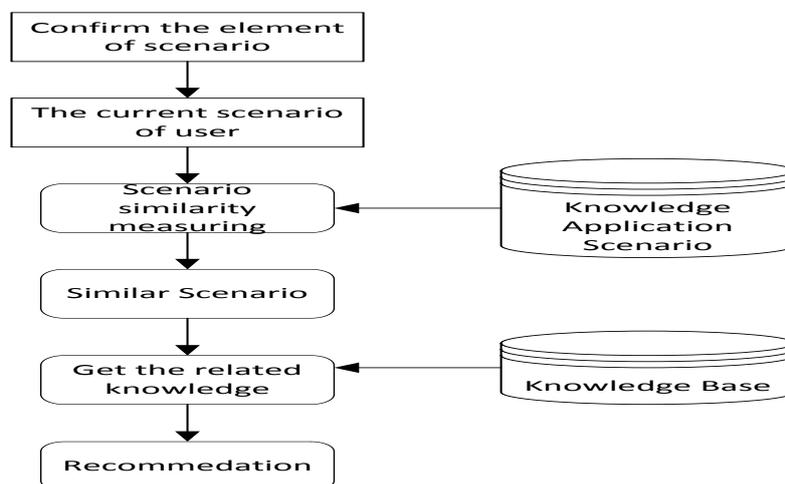


Figure 1. The process of recommend

The process fall mostly into four steps:

(1) Confirm the elements of scenario and get the current scenario of user. When the system recommends the knowledge to the user, it should be based on knowledge that the user need at the current scenario. However, there are too many elements of scenario and each of them include large amounts of information. Moreover, if lots of information of scenario is brought in recommendation, it has no positive effect on improving recommendation quality. So the KAS model should choose the suitable elements to depict the scenario.

(2) Search the most similar scenario to the current scenario of user in the KAS Library. What knowledge is recommended for the user, depend upon the level of similarity among the scenario. So,

how to compute similarity between the historical scenario and the current scenario of user is very important.

(3) Extract knowledge from the knowledge base. Implement association rules that between the knowledge item and KAS. Select the knowledge items, which related to the most similar historical scenario to the current scenario of user, from the knowledge base.

(4) Knowledge recommendation. The selected knowledge items is rank by the similarity of scenario (from high to low), so that we can get the Top-N recommendation list of knowledge. The list is provided to the user to click, read and bookmark.

Key Issues to be Resolved

User's Interests Feedbacks. The most KBS (Knowledge Base System) don't have function that scoring knowledge item and the users would not like to take their time to estimate it. In order to analyze developers' interest in the knowledge item easier, confirm the elements of scenario and make the calculation of similarity between the scenarios more accurate, we quantify the user's feedbacks. The user's feedbacks is based on the browsing behaviors including click rates, browsing time, and bookmark of knowledge item. We can get the users' browsing behaviors from the users' log data in the server and transform them into the user ratings. High click rate, long browsing time and the behavior of bookmark represent the user is very interested in the knowledge item. The ratings formulas are as follows:

$$R(k) = w_1 S(k) + w_2 \frac{C(k)}{\max\{C(v)\}} + w_3 \frac{T(k)}{\max\{T(v)\}} \quad (1)$$

Where $R(k)$ is users' ratings of the knowledge item k , $S(k)$ is bookmark (the value is 0 or 1), $C(k)$ is the users' click rate to k , $\max\{C(v)\}$ is the highest click rate to the set of knowledge resources v , $T(k)$ is the browsing time, $\max\{T(v)\}$ is the longest browsing time to the set of knowledge resources v . w_1, w_2, w_3 are weight value of bookmark, click rate and browsing time. In this paper, we set $w_1=0.5, w_2=0.3, w_3=0.2$ according to the experience.

Confirm the Elements of Scenario. Sometimes too many scenario elements can reduce recommendation quality, so we should choose the suitable elements to depict the scenario. We implement composition of different element of scenario by GBDT (gradient boosted decision trees) and then calculate the weight value of different composition of elements of scenario by LRE (logistic regression equation).

At first, we define a three-dimensional model and it is different with two-dimensional model ($U(User) \times V(Item) \rightarrow R(Ratings)$) that the traditional recommended system use: $U \times V \times C \rightarrow R$

Where $U = \{u_1, u_2, \dots, u_M\}$ is the user set. $V = \{v_1, v_2, \dots, v_N\}$ is the knowledge item set that recommended. $R = \{R_{i,j} | u_i \in U, v_j \in V\}$ is the user-rating-data matrix. The user u_i rating of knowledge item v_j is $R_{i,j}$ and $R_{i,j} \in [0, 1]$. $C = \{c_1, c_2, \dots, c_q\}$ is the scenario element array and c_i is one element of scenario such as knowledge domain.

And then, we start to construct the construct GBDT (gradient boosted decision trees). The first step is to define a remaining set $S = \{R_{ij} | C_k, u_i, v_j\}$. Where S represent that user u_i give the item v_j rating is R_{ij} on the condition the scenario element array is C_k . The initial remaining set S is whole data set. The process of generating decision tree is how to select the optimal element as the standard of decision tree splitting so that we can separate the remaining set S into left and right sub trees. Iterates the scenario element array from the root and select a scenario element c_i including the corresponding values $value_j$. Put the each element of the remaining set S into the different subset S_{left}, S_{right} based on the condition that $c_i = value_j$. Calculate the mean-square deviation of S_{left}, S_{right} , the initial remaining set and select the optimal element as the standard of decision tree splitting according to the Eq. 2:

$$cost = \left| \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x}) + \frac{1}{M} \sum_{i=1}^M (y_i - \bar{y}) - \frac{1}{M+N} \sum_{i=1}^{N+M} (z_i - \bar{z}) \right| \quad (2)$$

Where $x_i \in S_{left}$, $y_i \in S_{right}$, $z_i \in S$. \bar{x} , \bar{y} and \bar{z} are represented by the mean rating of S_{left} , S_{right} , S .

Then S_{left} , S_{right} are as the new remaining set continuing to split follow the step above until they satisfies one of the two conditions as below:

- (1) All the elements in the scenario element array are iterated
- (2) Meet the preset height of the tree.

It's also noteworthy that the element c_i that in the scenario element array C , has been selected as node should be deleted from the array. At last, calculate mean value in leaf node and update the remaining set S .

According to the Fig. 2, we know the path from the root to the leaf is a combination of scenario elements through the method above. When all the decision trees is traversed, we can get different combinations of scenario elements. And then, we can calculate the weight value of it by LRE (Logistic regression equation). The LRE is shown as Eq. 3 and Eq. 4:

$$L(\omega) = \sum_{i=1}^n y_i(\omega^T x_i) - \sum_{i=1}^n \log(1 + e^{\omega^T x_i}) \tag{3}$$

$$\frac{\partial L(\omega)}{\partial \omega} = \sum_{i=1}^n (y_i - \sigma(\omega^T x_i))x_i \tag{4}$$

Where ω is regression coefficient, σ is sigmoid function $\sigma(x) = (1 + \exp(x))^{-1}$, n is the count of the scenario element. x_i is the i th element of the combination of scenario elements and the value is 0 or 1. y_i is whether the rating in the training set is higher than average rating of the user. At last, we can get the weight value ω of combination of scenario elements by the LRE solving. Through comparing the value of ω , we can confirm which combination of scenario elements is better, and we choose the best one.

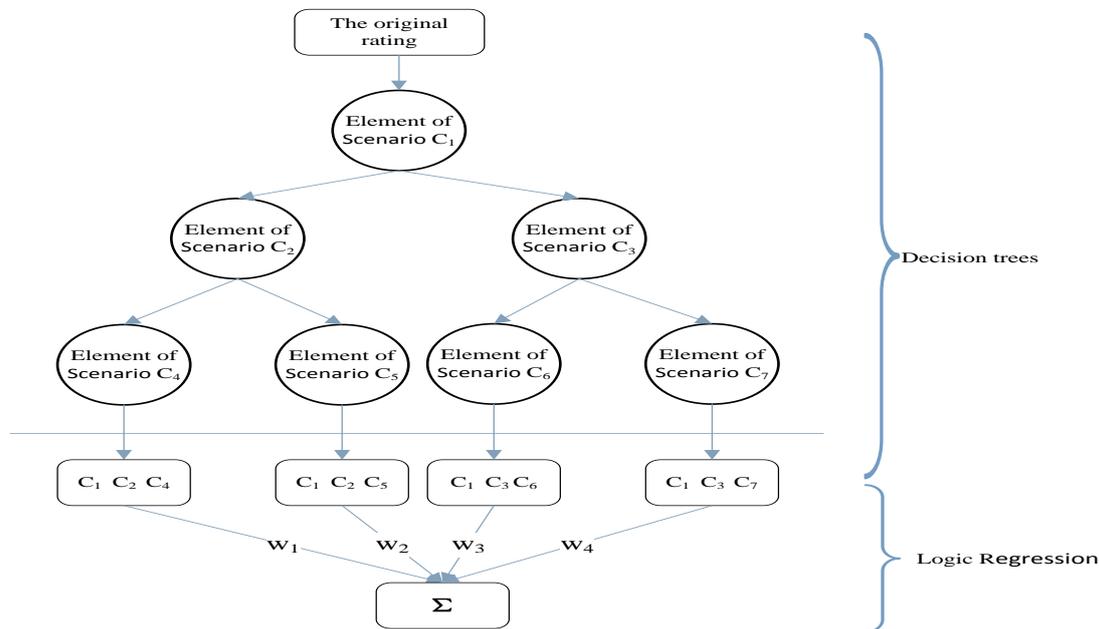


Figure 2. The process of GBDT and LRE

Search the most Similar Scenario. After we confirm the important scenario elements, we find the similar historical scenario by computing similarity between the historical scenario and the current scenario of user:

$$sim(C_1, C_2) = \sum_{t \in C} W_t \times sim(t_1, t_2)$$

Where $t \in C$ represent t is one of the set of the important scenario elements C , W_t is weight value of t . $sim(t_1, t_2)$ is the similarity function when $t=t_1$ and $t=t_2$.

When t is vector type:

$$sim(t_1, t_2) = \frac{t_1 \bullet t_2}{t_1 \times t_2} = \frac{\sum_{i=1}^{|t_1|} \sum_{j=1}^{|t_2|} t_{1i} t_{2j}}{\sum_{i=1}^{|t_1|} t_1^2 \sum_{j=1}^{|t_2|} t_2^2} \tag{5}$$

When t is collection type:

$$sim(t_1, t_2) = \frac{|t_1 \cap t_2|}{|t_1 \cup t_2|} \tag{6}$$

Experiment Results and Analyses

Experiment Dataset. In order to validate the performance of this algorithm, the author get the log file from the state-owned enterprise’s Knowledge Management System as the experiment dataset. After the data cleaning, extract 14590 effective recordings from the dataset. The dataset include 400 users and 1654 knowledge items.

Evaluation on the Experiment Method. The recommended way in the article is Top-N, that is recommending a list of knowledge item for user. So we verify the accuracy of the prediction by precision and recall:

$$Precision = \frac{\sum_{k \in K} |R(k) \cap T(k)|}{\sum_{k \in K} |R(k)|}$$

$$Recall = \frac{\sum_{k \in K} |R(k) \cap T(k)|}{\sum_{k \in K} |T(k)|}$$

Where k is knowledge item, K is test set of knowledge item, $R(k)$ is the count of the recommended item, $T(k)$ is the count of the item that user interested in.

Experiment and Analysis. At first, we select 15 elements possible related to the scenario from the log data, and convert the user’s browsing behavior into the user’s rating of the knowledge item. And then we extract 80% of recordings is as training set and the rest is as test set.

According to the method above (See 3.2.2), we set the height of the tree is 5 and the number of decision trees is 3. We get the different combination of scenario elements and calculate the weight value of each combination as Table 1

Table 1 Weight value of each combination

Combination of scenario	Weight value
Domain, Goal, Time, Resource, Person	0.532
Domain, Goal, Time, Organization, Person	0.334
Domain, Goal, Time, Organization, Last Click	0.134

We can choose combination of scenario elements $C = \{Domain, Goal, Resource, Person, Time\}$, the meaning is as Table 2

Table 2 The meaning of the element

Elements	Description
Domain	Areas of knowledge
Goal	What problem the knowledge item can solve
Time	The create time of knowledge item
Resource	Where the knowledge item is from
Person	The author

And then we can see which one has better prediction accuracy in Collaborative Filtering, Item Rank and Scenario Similarity by calculate the precision and recall. The result is shown as Fig. 3 and Fig. 4

In the Fig. 3 and Fig. 4, the bottom axis N reflects the count of recommended knowledge item. We can see the precision of Scenario Similarity recommendation in this article is improved obviously. However, the recall of Scenario Similarity recommendation hasn't improve much. It illustrates that the accuracy of recommendation is improved, but the coverage is still difficult to promote.

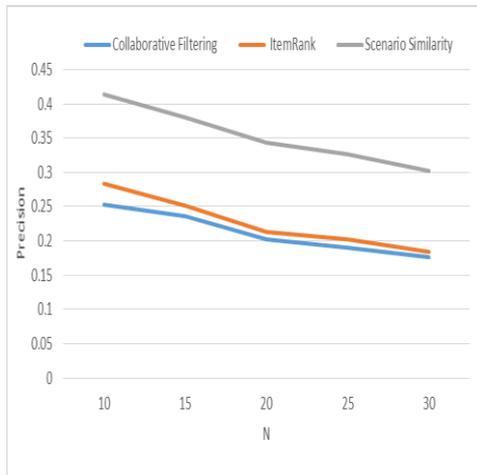


Figure 3. The precision comparison

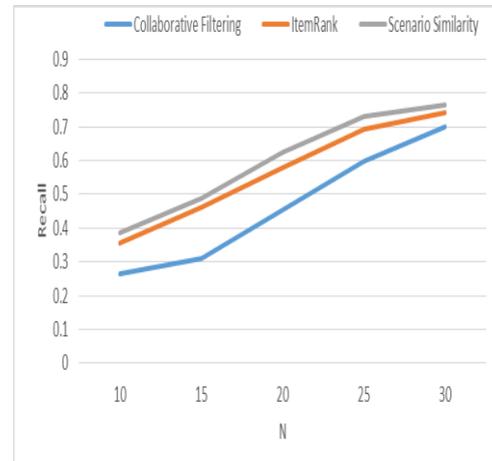


Figure 4. The recall comparison

Conclusion

With the development of the information-society, it is a very important task how to acquire the knowledge you need quickly and accurately. The paper introduces the knowledge scenario into personalized knowledge recommendation. At first, we confirm the elements of scenario by GBDT (gradient boosted decision trees) and LRE (logistic regression equation). And then we calculate the similarity of the knowledge scenario and recommend the knowledge items associated with the similar knowledge scenario. At last, we draw the conclusion that the personalized knowledge recommendation based on the knowledge scenario can improve the accuracy of recommendation through the experiment. However, there are still many deficiencies in the paper, such as the weight value of the browsing behavior and individual element of scenario is set according to the experience and they should be improved in the further study.

Acknowledgements

The National Natural Science Foundation of China (31201121).
Provincial Teaching Research Project of Hubei Province (2012215).

References

- [1] Verbert K, Manouselis N, Ochoa X, et al. Context-aware recommender systems for learning: a survey and future challenges [J]. Learning Technologies, IEEE Transactions on, 2012, 5(4): 318-335.
- [2] Ma H, Yang HX, Lyu MR, King I. SoRec: Social recommendation using probabilistic matrix factorization. In: Proc. of the ACM Int'l Conf. on Information and Knowledge Management. ACM Press, 2008: 978-991.
- [3] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In SIGIR 2009, pages 203 - 210.
- [4] Guo L, Ma J, Chen ZM, Jiang HR. Learning to recommend with social relation ensemble. In: Proc. the ACM Int'l Conf. on Information and Knowledge Management. ACM Press, 2012:2599-2602.

- [5] Jamali M, Ester M. Trust Walker: A random walk model for combining trust-based and item-based recommendation. In: Proc. of the ACM SIGKDD Conf. on Knowledge Discovery and Data Mining. ACM Press, 2009. :397-405.
- [6] Zhou TC, Ma H, King I, Lyu MR. UserRec: A user recommendation framework in social tagging systems. In: Proc. of the 24th AAAI Conf. on Artificial Intelligence. AAAI Press, 2010:1486-1491.
- [7] Wu L, Chen EH, Liu Q, Xu LL, Bao TF, Zhang L. Leveraging tagging for neighborhood-aware probabilistic matrix factorization. In: Proc. of the ACM Int'l Conf. on Information and Knowledge Management. ACM Press, 2012:1854-1858.
- [8] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation, in Social Networks Proceedings of the 2010 ACM conference on Recommender systems (RecSys), 2010
- [9] S. Wasserman and K. Faust. Social Network Analysis. Cambridge Univ. Press, 1994.
- [10] Koren Y, Bell R, and Volinsky C. Matrix factorization techniques for recommender system. IEEE Computer, 2009, 42(1): 30-37.
- [11] Liu Q, Wang C, Xu C. A Modified PMF Model Incorporating Implicit Item Associations. Processing of 2008 International Conference On Machine Learning Vancouver, Canada, ,2008:880-887
- [12] Massa P, Avesani P. Trust-aware Bootstrapping of Recommender Systems .In Proc. of E CAI Workshop on Recommender Systems, 2006.
- [13] NADOVEZA D, KIRITSIS D. Ontology-based approach for con-text modeling in enterprise applications [J]. Computers in Industry, 2014, 65 (9) : 1218-1231.