A Recommender System from Semantic Traces Based on Bayes Classifier

Ning Wang, Marie-Hélène Abel, Jean-Paul A Barthès, Elsa Negre

A Recommender System from Semantic Traces Based on Bayes Classifier

Ning WANG[1], Marie-Hélène ABEL[1], Jean-Paul BARTHÈS[1], Elsa NEGRE[2]

1. UMR CNRS 7253 HEUDIASYC, University of Technology of Compiègne, Sorbonne Universities, Rue Roger Couttolenc, Compiègne, France
2. UMR CNRS 7243 LAMSADE, Paris-Dauphine University, Place du Maréchal de Lattre de Tassigny, Paris, France

ning.wang@utc.fr
marie-helene.abel@utc.fr
barthes@utc.fr
elsa.negre@dauphine.fr

ABSTRACT. *Collaboration allows integrating knowledge from every participant to achieve individual or collective goals. Thanks to informational environments, we can better organize, realize and record collaboration. Every activity produces a set of traces. With the help of a model of competency, traces contribute to evaluate the competency of users on a certain subject. In this article, we propose a semantic model of traces and analyze classified traces by means of a Bayes classifier. We exploit the results to offer users recommendations and decision aid.*

1.   Introduction

Collaboration is a source of power for human society development and progress [GRU 94]. As the complexity and difficulty of our projects increase, nowadays collaboration is more important. Thanks to information technology, collaboration is better organized in informational environment. In such an environment, users achieve purposes by taking different actions. We are interested in the results of actions as well as the actions themselves. A set of actions, step by step, is defined as a trace [ZAR 11]. Under careful modelling and analysis, traces could in return help indicating the competency of an individual [TOM 11]. Thus, with the information exploited from the traces, we can improve collaboration focusing on the reuse of traces for different purposes such as decision aid [GAR 11] or recommendation [CHA 13].

In this paper we propose a mechanism that models, records and analyzes users' traces. It allows evaluating competencies for recommending people with more expertise on a certain subject. The following tasks are needed to achieve this objective: (i) propose a semantic structure to record traces; (ii) propose a model of competency; (iii) evaluate traces using some classifier and in our case a Bayes classifier and semantic distance and (iv) propose recommendations accordingly.

In this paper we identify various limitations of the current studies on recommendation in Section 2. In Section 3 we propose a recommender system for exploiting traces. In Section 4 we illustrate our method by an example. Section 5 gives conclusions and points to directions for future works.

2.   Related works

The interest of a recommender system is justified by the need to manage the growing amount of information [ADO 05]. Recently various articles have been published about exploiting the traces with the help of semantics. [CHE 13] presents a mechanism for personalized knowledge search and recommendation adapting a suitable domain ontology according to the previous browsing and reading behavior of users. [VEN 10] proposed a novel conversational search and recommendation system that involves finding relevant information based on social interactions and feedback. [BRE 98], [CON 99] and [PEN 00] all tried to provide recommender using probabilistic modeling. But none of these studies focused on combining action model with probabilistic method for the recommendation of users' competency.

Our previous work tried to give a solution by TF-IDF [WAN 14], but it lacks capability when the number of features deciding a recommendation is large. Inspired by [GHA 10], [SCH 02] and [MEL 02], in the following we give recommendation based on Bayes classifier.

3.   Our approach

In the following we introduce how we orchestrate model of actions, model of competency and Bayes classifier to make a recommendation on users' competency. Figure 1 shows the structure of recommender system. Firstly users' actions are collected and modelled from an interactive platform. After being sifted by the filter of classification, we obtain classified traces, which allows a preliminary presentation back to the users. Alternatively, we apply an algorithm to calculate an index indicating the correlation between the classified traces of a certain user and a given subject. These values can lead to useful information that are presented as personalized recommendations, either to a group defined as a set of users of the platform, or to an individual user.
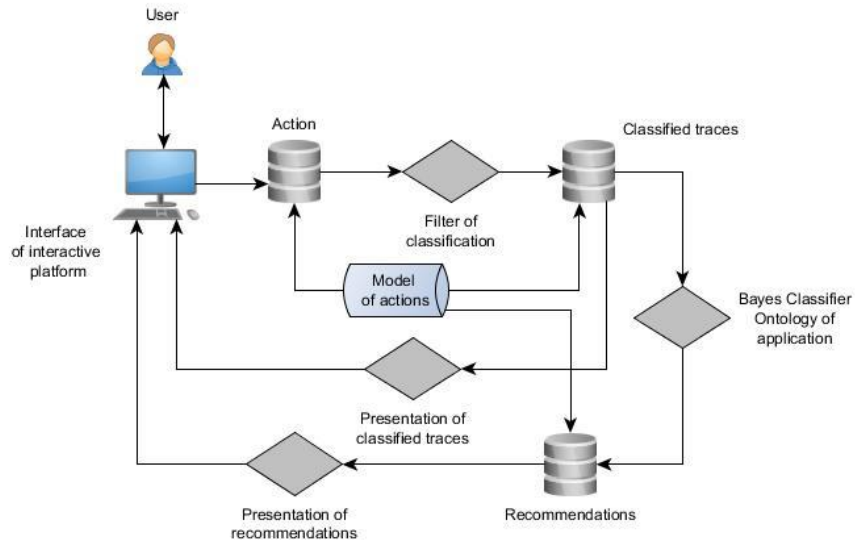

Figure 1. *The structure of our proposed recommender system for the exploitation of semantic traces*

3.1. *Model of Actions*

We define the principal concepts as follows:

− Action: an interaction or an act performed by a user in a collaborative environment, e.g. sending a document to other users.

− Classified trace: a set of actions that were performed by a user in the informational environment classified according to the model of traces [LI 12].

− Set of traces: an ordered set of classified traces.

According to our definition, action is the basic element forming a trace. Regarded as an important resource for our recommender system, we introduce the Resource Description Framework (RDF) to model actions [ANT 04]. RDF is used as a general formalism for conceptual description or modelling of information that is implemented in web resources. Figure 2 shows the basic structure in the RDF schema of our model. An ellipse represents a class of resources and a rectangle represents an object property. For example, a person has the object property "has_id_person" and the range of this property is a class called "id".
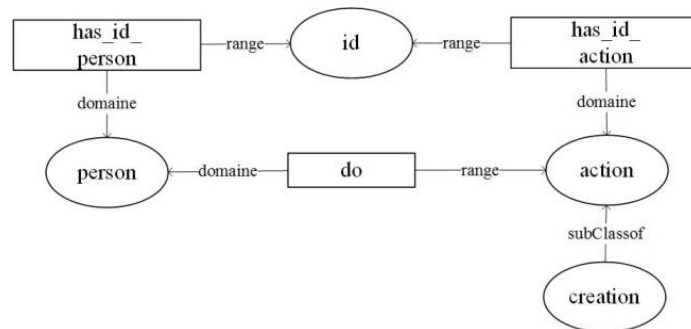
Figure 2. *Basic structure in RDFS graph presenting an action*

This model of actions has two main advantages compared to a traditional form of history or log of users:

− Actions are presented in a labeled, directed multi-graph. In our model actions are represented as resources in the RDF schema and they are connected by properties. This allows a better structure of storage and usage of actions. For example, a person "Ala" chats with "Ning". This action can be represented by an RDF instance showed in the lower part of Figure 3 where "Ala" and "Ning" are two

instances of the class of resources "person". "Chat_1" is an instance of the class of resources "conversation" which is linked to the action "creation".

− Normally different types of actions have different importance. For example creating elements of a Wiki is more important than consulting it. In our model actions are classified by three classes: creation, consultation and addition which enables to treat different types of actions more efficiently. We define the importance of creation, addition and consultation respectively as "high", "medium" and "low".

3.2. *Model of Competency*

Recently, many researchers have focused on modelling competency. [MOL 99] defines a "Core Competency" in the manufacturing clusters, including 4 generic and comprehensive components, namely: products, processes, skills, and task service. [MOL 06] proposes "Competence cells" for the competency cell-based networks in which the main components are "resources" and "fulfilled task or executed function". [BOU 05] presents a "s-a-r-C model" of competency consisting of "Professional Situations", "Actor" and "Resource". These models share two components, namely "resources" (including "human resources" or "physical resources") and "activity" (also called "process", "production skill" or "task").

The success of an activity requires actions on relevant concepts. For example, if we want to create a website, collaborators should put their knowledge about different concepts like "PHP", "Javascript" and "HTML" into effect. During the activity, their actions are recorded, e.g., creating a manual or sharing a technical article. Our system aims to analyze these actions and evaluate collaborators' competencies on different subjects, so that when the next time arrives, and a certain activity needs an expert on a certain concept, we can recommend a collaborator. As we are interested in the management of knowledge, resource is represented mainly in the form of knowledge. Thus we propose an "action-knowledge model" that integrates the merits in the models above and covers the strength of our system as showed in Figure 3.
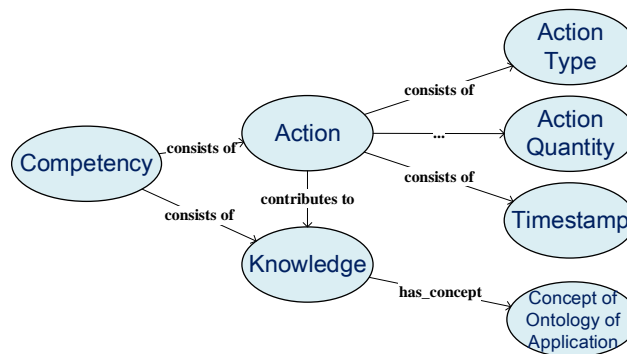

Figure 3. *Generic a-k model of competency*

The detailed definition of these components is as following:

− Action is how a user applies the knowledge. Action also helps accumulating the knowledge of a user. For example, if a user consults many files about "Java", it's reasonable to assume that his/her knowledge grows.

− Action Type describes different types of actions. Some types of actions directly contribute to competency, for example answering questions of other users or creating a Wiki about this concept. Such actions indicate that the user tends to be more competent about what he/she applies. Meanwhile other actions only contribute to the knowledge of the concept such as reading a paper about it. In 3.1, action types are described by semantic model of actions.

− Action Quantity records both a user and his collaborators' intensity of efforts on this activity.

− Timestamp records the time when an action took place.

− Knowledge is what a user applies during an action.

− Concept of Ontology of Application describes the nature of a user's action. It is the semantic description of knowledge.

With this model of competency we merge our methods in this section to evaluate users' competencies.

3.3. *Application of Bayes Classifier*

Previously, we focused on analyzing traces using TF-IDF [WAN 14]. As a trace is composed of actions on a set of concepts, we need a method that better handles multi-dimension factors. The Naïve Bayes classifier is based on the Bayes theorem with strong (Naive) independence assumption, and is suitable for the cases having high input dimensions [GHA 10]. In the following, we elaborate how to adapt the method of Bayes Classifier to our case.

Naïve Bayes is a conditional probability model. Given a problem instance to be classified, represented by a vector of n features $F = (F_1, ..., F_n)$, we tend to calculate the probability that it belongs to class C. Using Bayes' classic theorem, we have:

$$p(C \mid F_1, ..., F_n) = \frac{p(C)\, p(F_1, ..., F_n \mid C)}{p(F_1, ..., F_n)}$$

[1]

To simplify, we use the naïve Bayes classifier so that features $F_1,...,F_n$ are independent. Here we adapt the classic bag-of-words theory [MOO 00] and regard a trace as an independent bag of actions neglecting the logical relationship among the actions. Based on this assumption we have:

$$p(F_1,...,F_n \mid C) = p(F_1 \mid C),...,P(F_n \mid C)$$
$$p(F_1,...,F_n) = p(F_1),...,p(F_n) \qquad [2]$$

[1] is reformulated as:

$$p(C \mid F_1,...,F_n) = \frac{p(C)p(F_1 \mid C),...,P(F_n \mid C)}{p(F_1),...,p(F_n)} \qquad [3]$$

In our case, we aim at evaluating a user's competency on a certain concept with a trace he/she left on a set of concepts. So we adapt [1] as [4]:

$$p(C_j \mid T_i) = \frac{p(C_j)p(T_i \mid C_j)}{p(T_i)} \qquad [4]$$

where $p(C_j)$ is defined as the a priori probability that a random user has the highest competency on concept j of total J concepts. $p(C_j \mid T_i)$ represents the probability that a user i leaving trace T_i in the platform has the most competency on concept j among all the N users. $p(T_i)$ is the probability that a user leaves a trace like T_i. As previously described, the trace of a user is a combination of actions on a variety of concepts. We define $p(T_i)$ in [5]:

$$p(T_i) = p(A_{i,1}) \times \cdots \times p(A_{i,n}) = \prod_{k=1}^{n} p(A_{i,k}) \qquad [5]$$

where $p(A_{i,k})$ represents the probability that actions of trace i on concept k happen. T_i is composed of actions on n concepts respectively. So [4] becomes [6]:

$$p(C_j \mid T_i) = \frac{p(C_j)p(T_i \mid C_j)}{\prod_{k=1}^{n} p(A_{i,k})} \qquad [6]$$

$p(C_j)$ is a constant because with no other constraints, all the users have the same probability to perform the best at a certain concept. An estimate $\hat{p}(C_j)$ for $p(C_j)$ is calculated as:

$$\hat{p}(C_j) = \frac{1}{N} \qquad [7]$$

where N is the total number of users. We measure users' performances by the frequency of actions. We define $p(A_{i,k})$ as the top percentage of rank of frequency among all users. Thus the higher the frequency of actions that user i takes on concept j ranks, the smaller $p(A_{i,k})$ is. For example, user No. 9 has applied actions which ranks 2 out of a group of 10 users, then $p(A_{9,j}) = 0.2$. It means if we randomly choose a user x from this set of users, the probability that x performed at least equally as user No.9 is 0.2.

$p(T_i \mid C_j)$ represents the probability of user i having a trace T_i if user i has the most competency on concept j. Two factors influence this value. Firstly, if a user has the most competency on j, it is highly probable that this user has much competency on concepts semantically nearby. As T_i is composed of a set of actions $\{A_{i,k} \mid A_{i,k} \in T_i\}$, we evaluate the semantic distance between k and j. We use $\omega_{k,j}$ to represent the weight of concept k on j. Figure 4 shows a part of ontology of a use case for developing a semantic website. In view of complexity of calculation, we consider only the concepts semantically 2 edges away from j. Suppose j is the concept "Ontologic_request". Obviously, "Language" and "SQL" are two edges from j and we put their weight of influence to j as ω. "Request" and "SPARQL" are given 2ω and finally for the concept j itself we give 4ω. The sum of weight of concepts is 10ω which is equal to 1. Secondly, given the weight between concept k and j, the higher user i ranks on concept k, the larger $p(T_i \mid C_j)$ is. We define:

$$p(T_i \mid C_j) = \frac{1}{Z} \sum_{\{k \mid A_{i,k} \in T_i\}} [1 - p(A_{i,k})] \times \omega_{k,j} \qquad [8]$$

where Z is a scaling normalizing factor dependent only on $\{A_{i,k} \mid A_{i,k} \in T_i\}$, that is, a constant if the values of the feature variables are known.
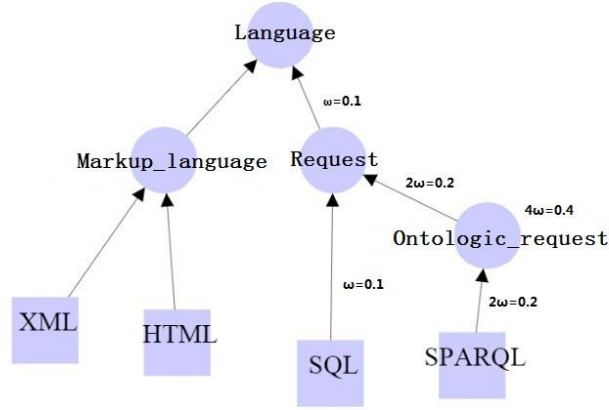
● **Figure 4.** *a part of ontology of a use case for developing a semantic website*

We have:

$$p(C_j \mid T_i) = \frac{\displaystyle\sum_{\{k \mid A_{i,k} \in T_i\}} [1 - p(A_{i,k})] \times \omega_{k,j}}{N \times Z \times \displaystyle\prod_{k=1}^{n} p(A_{i,k})}$$ [9]

Finally, we obtain $p(C_j \mid T_i)$ and by comparing the probability of all users on the same concept, we can finally give a recommendation about who is most probably the "best" at a concept given his/her traces.

4.   Example

We continue collecting and analyzing traces on the ontology of application shown in Figure 4. Table 1 shows the frequencies of all types of actions (addition, modification, etc.) and ranks of these frequencies among all users on concerned concepts respectively.

Table 1. *Traces of users on concepts from ontology of use case*

No.	Name	Frequency of Actions/Rank				
		Language	SQL	Request	SPARQL	Ontoligic_request
1	Ala	15/3	8/3	22/2	7/4	1/4
2	Ning	5/4	52/1	7/4	8/2	34/1
3	Xuan	22/2	4/4	63/1	12/1	9/2
4	Narjes	31/1	14/2	16/3	7/4	9/2

To calculate the probability of "being most competent" on "Ontologic_request" of "Ala", we have:

$$\hat{p}(C_{\text{Ontologic\_request}}) = \frac{1}{4} = 0.25$$

$$p(T_1) = \frac{3}{4} \times \frac{3}{4} \times \frac{2}{4} \times \frac{4}{4} \times \frac{4}{4} = 0.28125$$

$$p(T_1 \mid C_{\text{Ontologic\_request}}) = \frac{1}{Z} \sum_{\{k \mid A_{1,k} \in T_1\}} [1 - p(A_{1,k})] \times \omega_{k,\text{Ontologic\_request}} = \frac{0.15}{Z} \qquad [8]$$

$$p(C_{\text{Ontologic\_request}} \mid T_1) = \frac{p(C_{\text{Ontologic\_request}}) p(T_1 \mid C_{\text{Ontologic\_request}})}{\prod_{k=1}^{n} p(A_{1,k})} = \frac{0.133}{Z}$$

Table 2. *Probability of "being most competent"on "Ontologic_request"*

User Name	Action frequency on concept j
Ala	0.9%
Ning	25.8%
Xuan	59.7%
Narjes	13.6%

Results of calculation are shown in Table 2 in which "Xuan" earns the highest probability for the most competent so that he is recommended as expert on "Ontologic_request". We observe that although "Ning" has the most frequency on "Ontologic_request", he only ranks second for this calculated probability. On the contrary, "Xuan" is most probable not only because he ranks second for the frequency on "Ontologic_request", but he acted most frequently on semantically related concepts. From this view, the approach values especially the importance of semantic relations between concepts.

5.   Conclusion and future work

A full exploitation of traces helps us organizing and improving collaboration. In this article we proposed a model of competency and a semantic structure to record traces. Secondly we proposed recommendations based on the evaluation of traces using Bayes Classifier. Finally, we orchestrated these methods to evaluate the competencies of users. We illustrated our method by an example. Results meet our expectation showing that this approach takes good care of semantic relations between concepts

Future works include implementing our proposal of recommender system. Our previous work includes solving similar problem with TF-IDF [WAN 14]. Testing methodology is needed to compare this method with our previous work and other models.

6.   References

[ANT 04] Antoniou, G., & Van Harmelen, F. (2004). *A semantic web primer* 2004, pp. 253-260. MIT press.

[BOU 05] Boucher, X., Peillon, S., & Burlat, P. Towards a decision support for a collaborative increase of competencies within networks of firms. Ecole de Mines de Saint-Etienne, Research Report G21-EMSE, 2005, 600-009.

[BRE 98] Breese, J. S., Heckerman, D., & Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. *In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence* (pp. 43-52). Morgan Kaufmann Publishers Inc, 1998.

[CHA 13] Chang, N., Irvan, M., & Terano, T. A TV Program Recommender Framework. Procedia Computer Science, 22, 561-570, 2013.

[CHE 13] Chen, Y. J., Chu, H. C., Chen, Y. M., & Chao, C. Y. Adapting domain ontology for personalized knowledge search and recommendation. Information & Management, 50(6), 2013, 285-303.

[CON 99] Condliff, M. K., Lewis, D. D., Madigan, D., & Posse, C. Bayesian mixed-effects models for recommender systems. In ACM SIGIR (Vol. 99, pp. 23-30), 1999.

[GAR 11] García-Crespo, Á., López-Cuadrado, J. L., Colomo-Palacios, R., González-Carrasco, I., & Ruiz-Mezcua, B. Sem-Fit: A semantic based expert system to provide recommendations in the tourism domain. *Expert systems with applications*, 38(10), 13310-13319.

[GHA 10] Ghazanfar, M., and Adam P., "An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering." 2010.

[GRU 94] Grudin, J. "Groupware and social dynamics: Eight challenges for developers." Communications of the ACM 37.1 (1994): 92-105.

[LI 12] Li, Q., Abel, M. H., & Barthès, J. P. Facilitating Experience Sharing Groups-Collaborative Trace Reuse and Exploitation. *In Proceeding of International Conference on Knowledge Management and Information Sharing*, 2012, pp. 21-30.

[MEL 02] Melville, P., Mooney, R. J., & Nagarajan, R. Content-boosted collaborative filtering for improved recommendations. In AAAI/IAAI (pp. 187-192), 2002.

[MOL 99] Molina, A., & Flores, M. A virtual enterprise in Mexico: From concepts to practice. *Journal of Intelligent and Robotic Systems*, 26(3-4), 1999, 289-302.

[MOO 00] Mooney, R. J., & Roy, L. Content-based book recommending using learning for text categorization. *In Proceedings of the fifth ACM conference on Digital libraries* (pp. 195-204). ACM, 2000.

[MUL 06] Müller, E. Production planning and operation in competence-cell-based networks. *Production Planning & Control*, 17(2), 2006, 99-112.

[PEN 00] Pennock, D. M., Horvitz, E., Lawrence, S., & Giles, C. L. Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. *In Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence* (pp. 473-480). Morgan Kaufmann Publishers Inc., 2000.

[TOM 11] Tomaz, L. F., Nt, J. A. R., Souza, J. M., & Xexéo, G. B. Bringing knowledge into recommendation systems. *In Computer Supported Cooperative Work in Design (CSCWD)*, 2011 15th International Conference on (pp. 246-252). IEEE, 2011.

[SCH 02] Schein, Andrew I., Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. "Methods and metrics for cold-start recommendations." *In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 253-260. ACM, 2002.

[VEN 10] Venkatesh, A., Sahay, S., & Ram, A. (2010). Cobot: Real Time Multi User Conversational Search and Recommendations. *Recommender Systems and the Social Web at ACM RecSys. 2010.*

[WAN 14] Wang N., Abel M.H., Barthes J.P. and Negre E., "Towards a Recommender System from Semantic Traces for Decision Aid", *KMIS, 2014.*

[ZAR 11] Zarka, R., Cordier, A., Egyed-Zsigmond, E., & Mille, A. Trace replay with change propagation impact in client/server applications. Actes de IC2011, 607-622.