Personalisation in Service-Oriented Systems Using Markov Chain Model and Bayesian Inference

Jakub M. Tomczak[1] , Jerzy Świątek[1]

[1] Institute of Computer Science, Faculty of Computer Science and Management,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{Jakub.Tomczak, Jerzy.Swiatek}@pwr.wroc.pl

Abstract. In the paper a personalization method using Markov model and Bayesian inference is presented. The idea is based on the hypothesis that user's choice of a new decision is influenced by the last made decision. Thus, the user's behaviour could be described by the Markov chain model. The extracted knowledge about users' behaviour is maintained in the transition matrice as probability distribution functions. An estimation of probabilities is made by applying incremental learning algorithm which allows to cope with evolving environments (e.g. preferences). At the end an empirical study is given. The proposed approach is presented on an example of students enrolling to courses. The dataset is partially based on real-life data taken from Wrocław University of Technology and includes evolving users' behaviour.

Keywords: modeling, Markov chain, Bayesian inference, incremental learning.

1 Introduction

Many modern computer networked systems (e.g. e-commerce, web services) are used to provide services to users. Moreover, if the services play a crucial role in the system, such systems could be called *service-oriented systems* (SOSs) [2], [5], [6]. [7], [8]. However, in SOSs there are several main problems that have to be concerned [2], [7]: i) user's demand formulation and contract negotiation; ii) user's demand matching with accessible services including aspects such as e.g. knowledge about users' behaviour; iii) service execution on physical machines concerning network quality of service (QoS).

To solve the mentioned problems a process consisting of negotiation, discovery, and execution stages could be proposed. The stage for demand translation and contract negotiation could be based on ontology knowledge representation [10]. Next, in the service discovery there are three main procedures: i) service matching with user's demand (contract), ii) personalisation of services, iii) new service composition if there is no accessible service fulfilling user's demand. Service matching could be made by applying rough set theory [2], and service composition – i.e. using ontologies [8]. Because the personalisation is the main topic of this work, thus it will be discussed in Section 3. The last stage (execution) has to handle final aspects of the whole process and be QoS-aware [5], [6], [7].

This paper consists of following sections. In Section 2 a contribution of proposed approach to the technological innovation is presented. Next, the general problem of personalisation is described and the solution using Markov chains is outlined. Moreover, a recursive expression for decision making using Bayesian inference is presented. At the end an empirical study is conducted. The presented approach is applied to the real-life problem of students enrolment to courses at Wrocław University of Technology (WTU) and could be used as a new functionality in the existing educational platform (so called *EdukacjaCL*).

2 Contribution to sustainability

Applying the personalisation method in service-oriented systems aims in increasing quality of user service. It is widely used in the e.g. e-commerce [1], but there is a constant need for new approaches and applications. In this paper an approach of a Markov model as a knowledge representation and Bayesian inference as a reasoning method is proposed. Furthermore, an interesting result of using Markov model and Bayesian inference is that the decision is made due to a recursive procedure (see Section 3.1). Moreover, using knowledge about users gives SOSs a new function of sustainability.

Moreover, an another novelty is using an *incremental learning paradigm* for probabilities estimation [14]. Learning about users' preferences, which evolve in time, is one of the crucial aspects in modern SOSs. And to solve it some adaptive approach has to be applied. Applying incremental learning affects in sustaining model accuracy.

Therefore, in this paper a compact framework using probabilistic model and inference for personalisation problem is proposed. The framework tries to maintain the system's sustainability by making optimal decisions about users' demands and keep an up-to-date knowledge about users.

3 Personalisation in service-oriented systems

Mining knowledge about users in computer systems could lead to increasing quality of user service and profits to service provider(-s), e.g. to overcome so called *information overload* [1], [11], [15]. Therefore, there are different aspects of personalisation, concerning e.g. recommendation, user tutoring, adaptation of layout and content, and so on [15].

However, in this paper the main concern is put on the recommendation task. In other words, on example of an educational platform, during student's enrolment to a course, a sorted list of courses *the best* suited to her/his demands is presented. In the literature there are different approaches to personalisation [1], e.g. individual or collaborative, user or item information, memory- or model-based.

In presented approach it is assumed that user's decision depends on previously made decisions. For example, a student enrols to a new course based on courses she/he has been enrolled in the past. Besides, the decision is rather uncertain because of e.g. lack of information, and that is why a new decision could be described by a

probability distribution function. Hence, the Markov chain model [3], [16] seems to be a proper knowledge representation to model users' behaviour because it allows to model a situation that previous decisions affects the current decision. Then, to reason about the maintained knowledge the Bayesian inference is used.

3.1 Problem statement

As it was mentioned in previous section, a users' behaviour is modelled using Markov chains. Thus, the problem of personalisation could be stated as a classification task.

Let assume that an input in the N^{th} moment is described by a vector of features, $\mathbf{u}_N \in \mathbf{U}=\mathbf{U}_1\times\mathbf{U}_2\times...\times\mathbf{U}_S$ (card$\{\mathbf{U}_s\}$ = $K_s < \aleph_0$). The sequence of inputs is denoted by $\overline{\mathbf{u}}_N = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N)$. A decision in the N^{th} time step is denoted by $j_N \in \mathbf{M}=\{1,2,...,M\}$. Furthermore, we assume that $\mathbf{u}_n$, and j_n, for each n = 1, 2, ..., N are realizations of stochastic processes drawn with distributions $P_n(j_n)$ i $P_n(\overline{\mathbf{u}}_n|j_n)$. However, in further considerations it is assumed that decisions j_1, j_2, …, j_N forms a Markov chain [3] described by an initial (*a priori*) vector of probabilities, p_1, $p_1^{(i)} = P_1(j_1 = i)$, i=1,2,…,M, and a sequence of transition matrices, P_N = $[p_{N,i,j}]$, i,j=1...M, $p_{N,i,j} = P_N(j_{N+1} = j \mid j_N = i)$. Besides, it is assumed that the observations of users are independent, $P_N(\overline{\mathbf{u}}_N \mid j_N) = \prod_{n=1}^{N} P_n(\mathbf{u}_n \mid j_n)$.

Thus, the problem could be stated as follows. Having a new user *the best* suited decision should be made. It could be made by minimizing following risk functional

$$R_N(\overline{\Psi}_N) = \sum_{n=1}^{N} \mathbf{E}\left[L\left(J_n, \Psi_n(\mathbf{X}_n)\right)\right], \tag{1}$$

where $\overline{\Psi}_N$ = [Ψ_1 Ψ_2 … Ψ_N] is a vector of decisions, $i_n = \Psi_n(\mathbf{u}_n)$, $\mathbf{E}[\cdot]$ is an expected value and $L(\cdot,\cdot)$ is a chosen loss function. It is easy to notice [3] that to minimize risk functional (1) it is enough to consider

$$R_n(\Psi_n) = \mathbf{E}\left[L\left(J_n, \Psi_n(\mathbf{X}_n)\right)\right] = \int \sum_{m=1}^{M} L(m, \Psi_n(\overline{\mathbf{u}}_n)) \cdot p_n^{(m)} \cdot P_n(\overline{\mathbf{u}}_n \mid m) \mathbf{d}\overline{\mathbf{u}}_n, \tag{2}$$

where *a priori* distribution in N^{th} moment is $p_n^{(m)} = \sum_{l=1}^{M} p_{n,m,l} \cdot p_{n-1}^{(l)}$.

Thus, we can propose an optimal decision making algorithm (*Bayesian algorithm*)

$$i_n = \arg\min_{l=1,2,...M} \left\{r(l, \overline{\mathbf{u}}_n)\right\}, \tag{3}$$

where: $r(l, \overline{\mathbf{u}}_n) = \sum_{m=1}^{M} L(m,l) \cdot p_n^{(m)} \cdot P_n(\overline{\mathbf{u}}_n \mid m)$, and for 0-1 loss function:

$$\delta(l, \overline{\mathbf{u}}_n) = p_n^{(l)} \cdot P_n(\overline{\mathbf{u}}_n \mid l) \stackrel{\Delta}{=} \delta_n^{(l)}. \tag{4}$$

It could be noticed [3] that using assumption about independence of input observations and that *a priori* distributions depends on recent *a priori* distribution and current transition matrix, the expression for dependent risk functional (5) could be calculated using following recursive procedure:

$$\delta_{N+1} = D_{N+1}(\mathbf{u}_{N+1}) \cdot P_N \cdot \delta_N, \tag{5}$$

$$\delta_1 = D_1(\mathbf{u}_1) \cdot p_1$$

where $\delta_N = [\delta_N^{(1)} \ \delta_N^{(2)} \ ... \ \delta_N^{(M)}]^T$ and

$$D_N(\mathbf{u}_N) = \begin{bmatrix} P_N(\mathbf{u}_N \mid 1) & 0 & \cdots & 0 \\ 0 & P_N(\mathbf{u}_N \mid 2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_N(\mathbf{u}_N \mid M) \end{bmatrix}. \tag{6}$$

Hence, to make a decision for a given input in N^{th} time step it is enough to have matrices $D_N(\mathbf{u}_N)$, P_N, and the vector δ_{N-1}. It is important due to the learning stage.

3.2 General methodology

However, from the computational point of view it is important to have as small number of features as possible with smallest loss of information. Therefore, the number of inputs could be decreased to minimum. For example, a student could be described by a mean value of grades, number of courses he attended. Then students could be clustered into groups and each group contains students who have similar descriptions. One group is so called *context* [4], [9].

Furthermore, in real situations the probability distributions are unknown. Therefore, a learning stage is needed. During the learning process the distributions are estimated based on training sequence, $(\mathbf{u}_n, j_n)$, $n = 1,2, ..., N$. It is worth to notice, that at each n there could be more than one observation, $(\mathbf{u}_{n,k}, j_{n,k})$, $k=1,2,...,K$. However, in real environments the non-stationarity occurs which means that the estimation cannot be made using all observations, e.g. users' preferences which evolve in time [15].

Thus, following general methodology could be propose:

1. Divide users descriptions (demands) into clusters (each cluster is called a context) using some chosen clustering algorithm (e.g. *k-Means*).
2. *Learning Stage*:
   i.) At the beginning estimate p_1, $D_1(\cdot)$
   ii.) At each learning step $N>1$ estimate $D_N(\cdot)$, P_N, and δ_{N-1}.
3. *Classification stage*:
   i.) For given user's demand find an according context (e.g. using 1-*Nearest Neighbour* classifier).
   ii.) For given user's context make a decision using (5).

3.3 Incremental learning algorithm

As it was mentioned, in many real-life situation the non-stationarity occurs. Therefore, an adaptive estimation methods have to be proposed. Such adaptation to changes could be made by applying *incremental* learning algorithms [14].

In presented approach a following method of estimation is proposed. Let us assume that at each learning step there are K observations which come in a data stream (sequence). It means that k^{th} observation occurs before $(k+1)^{\text{th}}$ observation. Then, the probability distribution is estimated using a frequency matrices (matrices of the same sizes as $D_N(\cdot)$, P_N, and δ_{N-1}) but consisting frequency of occurrence of responding input or decision. Let denote such matrix as $V_{D,N}$, $V_{P,N}$, $V_{\delta,N-1}$. Then, the learning algorithm could be as follows[1]:

1. Take a new observation and initiate all matrices.
2. Update the appropriate frequency matrix by a new k^{th} observation,
$$V_N^{(k)} = a_N \cdot V_{N-1} + V_N^{(k-1)} + W_k$$
where W_k is a matrix with ones in proper places for observation's decision and context (determining row and column), and zeros – otherwise, $a_N \in [0,1]$ is a forgetting factor.
3. Estimate probability distribution using appropriate frequency matrix. If there is a new observation, then go to 1.

The key issue is to fix a proper value of the forgetting factor because it affects the estimation. It can have a different value at each learning step or be constant.

4 Experimental study

Presented approach is checked on the example of the educational platform at WTU. The platform is dedicated to students service and enables e.g. sending applications, signing up for courses, checking past grades, and so on. Therefore, the educational platform could be seen as a SOS in which different services could be distinguished.

However, in this work the proposed approach is used in the enrolment service. At WTU there are different services for enrolment: faculty enrolment, specialisation enrolment, sport enrolment, and foreign languages enrolment. Each of mentioned enrolment needs reading about tens or even hundreds of course descriptions which is a big waste of time both for student and system. In the experiment on the example of a enrolment for foreign language it is to shown how proposed approach could be used.

4.1 Experiment details and results

In the experiment the state in the Markov chain is associated with the language and its level: *English A1, English A2, English B1, English B2, English C, German A1, German A2, German B1, German B2, German C, nothing*. At WTU there are more

[1] Because the algorithm is the same for all matrices, therefore indexes by V are skipped.

languages available (e.g. Korean, Japanese, Czech, Italian, Swedish) but for transparency of the experiment it was limited to 11. The *nothing* means that student is not signed up for any course.

Moreover, it is assumed that each student is described by a following vector: *number of all courses, number of exercises, number of laboratories, number of lectures, mean of grades, variance of grades, mean of grades from lectures, mean of grades from exercises, mean of grades from laboratories, number of fails (grade F), number of excellents (grade A).* To generate students a real logs from the educational platform was used. This dataset was used also in the previous works [12], [13].

Then the problem is as follows: ***Propose a student the most appropriate language and the level if recently she/he was enrolled to a course X***.

The experiment was conducted for 2 semester (one semester – one learning stage). In the semester one student signs up for a course after another.

The methodology of the experiment was following:

1. Using the real dataset a context was established using *k-Means* clustering algorithm. (There are 3 clusters).
2. Student is generated due to the appropriate probability distribution[2] and her/his description is mapped with the context id by using 1-*NN* classifier.
3. Initial state and next states are generated due to (5).
4. Incremental learning algorithm with a forgetting factor a is applied.

It was assumed that at the first learning stage (first semester) $a_1 = 0$.

Following methods were compared (for 100 students during a semester, 2000 students, and 5000 students):

- Markov chain (MC) model with Bayesian inference (5) with $a_2 \in \{0, 0.1, 0.25, 0.5\}$.
- Bayesian model (no assumption about Markov chain, *Bayes*) with Bayesian inference with $a_2 \in \{0, 0.1, 0.25, 0.5\}$.
- Random – list of states is given randomly.

Additionally, beside classification accuracy another performance index was used, called *position index*. This criterion calculates the difference of real state in the sorted due the probabilities list of decisions from the first position.

Table 1. Classification accuracy (mean value and standard deviation) for compared models.

	Mean			Std		
	100	**2000**	**5000**	**100**	**2000**	**5000**
MC 0.0	0,23	0,283	0,285	0,043	0,005	0,004
MC 0.1	0,229	0,283	0,285	0,042	0,005	0,004
MC 0.25	0,229	0,283	0,285	0,042	0,005	0,004
MC 0.5	0,228	0,282	0,285	0,04	0,005	0,004
Bayes 0.0	0,245	0,281	0,281	0,046	0,009	0,0055
Bayes 0.1	0,245	0,281	0,281	0,047	0,009	0,0055
Bayes 0.25	0,244	0,281	0,281	0,047	0,009	0,0055
Bayes 0.5	0,244	0,281	0,281	0,047	0,009	0,0055
Random	0,095	0,093	0,090	0,015	0,004	0,001

The experiment was conducted on 10 generated datasets (decisions about enrolments) and the results are a mean values. Classification accuracy and position

[2] Features *number of fails* and *number of excellents* were drawn from discrete distributions and all other – Gaussian.

index are calculated as a mean of two semesters. The results are shown in the tables 1 (*classification accuracy*) and 2 (*position index* – the lower the value the better).

Table 2. Position index (mean value and mean value of standard deviation) for compared models.

	Mean value of position index			Mean value of std of position index		
	100	**2000**	**5000**	**100**	**2000**	**5000**
MC 0.0	2,91	2,46	2,42	0,713	0,597	0,583
MC 0.1	2,85	2,45	2,42	0,662	0,592	0,581
MC 0.25	2,86	2,45	2,42	0,660	0,592	0,581
MC 0.5	2,87	2,43	2,42	0,665	0,591	0,581
Bayes 0.0	2,81	2,53	2,53	0,703	0,745	0,756
Bayes 0.1	2,78	2,53	2,53	0,636	0,741	0,754
Bayes 0.25	2,78	2,53	2,53	0,636	0,741	0,754
Bayes 0.5	2,76	2,53	2,53	0,639	0,740	0,754
Random	4,95	4,97	6,02	0,230	0,277	0,356

4.2 Discussion

First of all it has to be said that analysing classification accuracy for Markov model and Bayesian model no statistical difference could be noticed. In the situation with 100 students in semester one and two the Markov model performs worst because of its bigger size of matrices. However, in case of 2000 and 5000 students Markov model gave slightly better results. On the other hand, analysing position index Markov model was a little bit better than Bayesian model and the mean value of std shows that Markov model behaves in more stable way (difference of about 0.15).

Furthermore, applying Bayesian inference allows to propose around 1/3 of students proper decision or, in average, proper decision at 2^{nd} or 3^{rd} position in the list. In comparison to random method it is clear that Bayesian approach outperforms *random* personalisation. Moreover, the usage of the incremental algorithm gives slight improvement in the quality of classification accuracy and position index.

Concluding, presented experiment is conducted only on two semesters. Probably, carrying out the experiment on 10 semesters should show that applying Markov chain model is more appropriate.

5 Final remarks

In this paper the general approach using Markov chain model and Bayesian inference for personalisation was proposed. The presented method was evaluated on the partially real-life data take from the educational platform. The problem was formally stated and the general methodology was proposed.

In the future more attention should be paid in developing the whole system with specified, co-working modules. Moreover, given methodology should be applied in the existing educational platform as the additional feature.

Furthermore, the Markov model should be compared with other methods, e.g. using rules-based knowledge representation, or ensemble classifiers. And the higher order Markov chains are supposed to be considered.

Besides, the research ought to be conducted on bigger amount of data, including logs of users containing whole history, e.g. history of all enrolments.

Acknowledgments. The research presented in this paper has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.03.01-00-008/08.

References

1. Anand S.S., Mobasher B.: Intelligent Techniques for Web Personalization, in: "Intelligent Techniques for Web Personalization", LNAI, Vol 3169, 1-36 (2005)
2. Brzostowski K., Tomczak J.M., Nikliborc M.: A Rough Set-Based Algorithm to Match Services Described by Functional and Non-Functional Features, in: "Networks and Networks' Services", eds. A. Grzech, et al., , Ofic. Wyd. PWr, Wrocław, 27-38 (2010)
3. Bubnicki Z.: Pattern Recognition Algorithms for Simple Markov Chains, in: "Problemy informacji i sterowania", Ofic. Wyd. PWr, Wrocław, 3-18 (1972) (in Polish)
4. Chen N., Chen A.: Integrating Context-Aware Computing in Decision Support System, Proc. of the Int. ME&CS 2010, Vol. I, March 17-19, Hong Kong (2010)
5. Grzech A., Rygielski P.: Translations of Service Level Agreement in Systems Based on Service Oriented Architecture, in: KES2010, eds. Rossi Setchi et al., KES 2010, Part II, LNAI 6277, 523–532 (2010)
6. Grzech A., Rygielski P., Świątek P.: QoS-aware infrastructure resources allocation in systems based on service-oriented architecture paradigm, HET - NETs, 35-47 (2010)
7. Grzech A., Świątek P.: Modeling and Optimization of Complex Services in Service-Based Systems, Cybernetics and Systems: An International Journal, Vol. 40, 706–723 (2009)
8. Juszczyszyn K., Stelmach P., Grzelak T.: A Method for the Composition of Semantically Described Web Services, in: "Networks and Networks' Services", eds. A. Grzech, et al., Ofic. Wyd. PWr, Wrocław, 27-38 (2010)
9. Palmisano C., Tuzhilin A., Gorgoglione M.: Using Context to Improve Predictive Modeling of Customers in Personalization Applications, IEEE Trans. on Knowledge and Data Engineering, Vol. 20, No. 11, 1535-1549 (2008)
10. Pastuszko M., Kryza B., Słota R., Kitowski J.: Processing and negotiation of natural language based contracts for Virtual Organizations, Proc. of Cracow'09 Grid Workshop, ACC CYFRONET AGH, Kraków, 104–111 (2010)
11. Pierrakos D., Paliouras G., Papatheodorou Ch., Spyropoulos C.D.: Web Usage Mining as a Tool for Personalization: A Survey, User Mod. & User-Ad. Inter. Vol. 13, 311-372, (2003)
12. Prusiewicz A., Zięba M.: Services Recommendation in Systems Based on Service Oriented Architecture by Applying Modified ROCK Algorithm, in: Proc. of NDT 2010, eds. F. Zavoral et al., NDT 2010, Prague, Czech Republic, July 7-9, 2010, 226-238 (2010)
13. Sobecki J, Tomczak J.M.: Student courses recommendation using Ant Colony Optimization, LNAI, Vol. 5991, 124-133 (2009)
14. Tomczak J.M., Świątek J., Brzostowski K.: Bayesian Classifiers with Incremental Learning for Nonstationary Datastreams, in: "Advances in Systems Science", eds. A. Grzech, P. Świątek, J. Drapała, EXIT, Warszawa, 251-260 (2010)
15. Webb G.I., Pazzani M.J., Billsus D.: Machine Learning for User Modeling, User Modeling and User-Adapted Interaction, Vol. 11, 19-29 (2001)
16. Zhu J., Hong J., Hughes J.G.: Using Markov Chains for Link Prediction in Adaptive Web Sites, LNCS, Vol. 2311, 55-66 (2002)