History dependent Recommender Systems based on Partial Matching

Armelle Brun, Geoffray Bonnin, Anne Boyer

History dependent Recommender Systems based on Partial Matching

Armelle Brun, Geoffray Bonnin, and Anne Boyer

LORIA Laboratory, University Nancy 2, BP 239 54506 Vandoeuvre les Nancy
{armelle.brun,geoffray.bonnin,anne.boyer}@loria.fr,
http://kiwi.loria.fr

Abstract. This paper focuses on the utilization of the history of navigation within recommender systems. It aims at designing a collaborative recommender based on Markov models relying on partial matching in order to ensure high accuracy, coverage, robustness, low complexity while being anytime.
Indeed, contrary to state of the art, this model does not simply match the context of the active user to the context of other users but partial matching is performed: the history of navigation is divided into several sub-histories on which matching is performed, allowing the matching constraints to be weakened. The resulting model leads to an improvement in terms of accuracy compared to state of the art models.

1 Introduction

Due to the increase of the size of the web and the Internet traffic, users are overwhelmed by the quantity of information available. Personalization and recommendation systems, that predict user attempts and propose resources linked to their tastes, are thus becoming more and more popular.

Several types of recommender systems have been studied, as content-based recommenders, collaborative filtering, etc. In the frame of collaborative web recommender systems, not only the set of resources consulted by all the users has to be used, but the order of consultation of these resources is of major importance and has to be exploited to perform accurate recommendations. State of the art approaches use datamining techniques to perform recommendations, and the web usage mining can be defined as "the automatic discovery and analysis of patterns and clickstream collected as a result of interactions with Web resources" [1]. To discover navigational patterns, sequential association rules (SAR), Markov models (MM), etc. are classically used, among which MM are the most popular due to their accuracy.

In this article, we design a model that takes advantage of all the previous models, in terms of accuracy, robustness, space complexity and coverage. Furthermore, it is anytime, allowing its use in all real-time applications.

Section 2 presents some datamining models used in recommender systems and put forward that partial matching allows high coverage and robustness. Section 3 defines the proposed model. The next section is dedicated to the evaluation of this model. Conclusion and perspectives are put forward in the last section.

2 Datamining models for recommender systems

This section is dedicated to state of the art of datamining models used to perform history dependent collaborative recommendations. All these models assume that the consultation of a resource depends on the resources that the active user has previously consulted.

2.1 Sequential Association Rules

In the frame of web navigation [2], Sequential association rules (SAR) are used to capture dependences between resources. SAR are of the form $X => Y$ where X (the antecedent) is a sequence of items. Y, called the consequence, is a single resource.

In the recommendation step, if the antecedent of a SAR matches the history of navigation of the active user, we can deduce that the corresponding consequence resource is highly probable and may thus be predicted.

The advantage of SAR is that they are robust to noise: the SAR learned are not necessarily contiguous. Thus the matching step is more permissive than in the case of Markov models described in the next section. The use of SAR leads to a model with a low space requirements. The main drawback of SAR is the time required to learn them and filter out the most relevant ones. Such models also result in a low coverage.

2.2 K-order and All kth-order Markov Models

A k-order Markov model (KMM) assumes that the consultation of a resource depends only on the k previously accessed resources, the resources consulted before these k resources are considered as non-informative. Thus, a KMM computes the probability of accessing a resource given the sequence of the k previously accessed resources. Let $S_a = r_{a1}, \ldots, r_{al}$ be the active session, made up of the sequence of resources consulted by the active user u_a. A KMM estimates the probability $p(r_{al+1}|r_{al-k}, \ldots, r_{al})$ for each candidate resource. The resources that are recommended are those that have the highest probability.
Obviously, the higher the value of k is, the most accurate the probabilities are (in the case of a sufficiently large training dataset), and it has been shown [3] that the accuracy of KMM increases with the value of k. However, the higher the value of k is, the larger the number of states to be stored is and the lower the coverage is (as the probability that the history of size k perfectly matches one state of the model decreases).

To cope with the coverage problem, a All-kth-order Markov model (AKMM) has been proposed [4]. In this model, various KMM of different order k are trained and used to make predictions. Predictions are first computed by using a k-order MM, if no prediction can be performed, a $k-1$ order MM is used, etc. until a recommendation can be made. Such models provide a high coverage, but the number of states is dramatically increased.

KMM and AKMM are not robust to noise, as the history of navigation has to perfectly match a state of the model used. KMM and AKMM quickly reach their limits when the order of the models grows: both performance and accuracy decrease due to the size of the training data (probabilities are no more reliable).

2.3 Skipping Based Markov Models

The probabilistic model k-order Skipping Markov Model (KSMM) presented in [5], uses a KMM that allows skipping between the elements of the $k+1$-tuple, both during training and recommending step. The distance is limited to a predefined value D. Such a model has a low space complexity (similar to a KMM) while using resources at a distance higher than k. A weighting scheme is applied to these $k+1$-tuples, according to the distance between the resources. The frequency of a $k+1$-tuple in the training corpus is equal to the weighted sum of all the occurences of this $k+1$-tuple (within a distance lower than D). The corresponding conditional probabilities are then computed. Let for example the sequence (x, y, t, s, x, y, z). The triplet (x, y, z), occurs twice and the two weighted occurrences are added to the frequency of the triplet.

During prediction step, given the sequence of navigation of the active user $(r_{a1}, \ldots, r_{al})$, the probability of each resource r_{al+1} is computed as follows:

$$P(r_{al+1}|H) = P(r_{al+1}|r_{al-D}, \ldots, r_{al}) = 1 - \prod_{h \in H} 1 - P(r_{al+1}|h) * w(h, H) \quad (1)$$

where h sums over all the sub-histories of size k within the window of size D and $w(h, H)$ is the weight of history h in the whole history H. The probability of a resource r_{al+1} is based on the probability that none of the histories h predicts r_{al+1} as following resource.

This model has the advantage to use long-distance resources while being a low order MM and low complex. This model has been proved to be more accurate with a higher coverage than the corresponding KMM, but has the drawback of not reaching a total coverage (it is however higher than the one of KMM).

2.4 The advantage of Partial Matching

We have seen that SAR enable the use of long-distance resources in the history as they enable distance between the elements of the history that match the rule; only a sub-part of the history is used: they perform partial matching. This partial matching makes the model robust to noise and enables to perform recommendations even when the whole sequence of navigation of the active user does not perfectly match training data.

The KSMM model also divides the history into sub-histories and performs recommendations by using these sub-histories. Once more, this partial matching is robust to noise: the consultation of additional resources slightly influences the recommendation process, whereas it highly influences the accuracy of KMM and AKMM.

The KSMM model is robust and accurate, due to partial matching, its main drawback is its coverage as matching is performed only on sequences of size k. Let us remember that a 100% coverage is reached by AKMM due to the use of several KMM models of different order, but is not robust.
We propose here a model that exploits the characteristics of both preceding models (partial matching and several values ok k), resulting in a model having a high coverage and a high accuracy while being robust.

3 A all-kth-Order Skipping Markov Model

To prevent from the coverage problem of the KSMM, while keeping its high accuracy, we decide to create a all-kth order Skipping Markov Model (AKSMM). On the same principle than AKMM, k KSMM models are developed (from order 1 to order k). The KSMM of order k is first used to perform recommendations; if no resource can be recommended (the history of the active user does not match any conditional probability of the model), then order of the model is iteratively decreased until a resource can be recommended.

As the KSMM is more accurate than a KMM, we assume that the AKSMM will be more accurate than a AKMM while having a 100% coverage.

The resulting recommender is robust to noise as it relies on skipping, it has a 100% coverage rate and a low state-space complexity.

4 Experimental Evaluation

4.1 Corpus and Protocol

The dataset used for the evaluation is provided by the Crédit Agricole SA French banking group. It is made up of the logs collected on $3,391$ distinct Web pages (of an intranet of the group) browsed by 815 bank clerks, corresponding to a corpus of $123,470$ anonymous consultations. The corpus has been divided into training and test sets of 90% and 10% respectively.

To assess our models, we use the top-m score. This metric evaluates the average pertinence of recommendation lists. For each history of the test corpus, a recommendation list of size m is built, containing the m most probable resources according to the model. If the resource actually consulted by the user is in the recommendation list, the recommendation is considered as a success. This metric represents the percentage of pertinent recommendations.

We also evaluate the models in term of coverage, *i.e.* the percentage of cases where the model can recommend a resource.

4.2 Experimental Results

Before evaluating the AKSMM in terms of accuracy and coverage, the left part of Table 1 presents performance of KMM and AKMM on our corpus for comparison purpose. The size of the recommendation list is set to the usual value of 10.

Table 1. Accuracy and coverage of KMM and AKMM according to the value of k

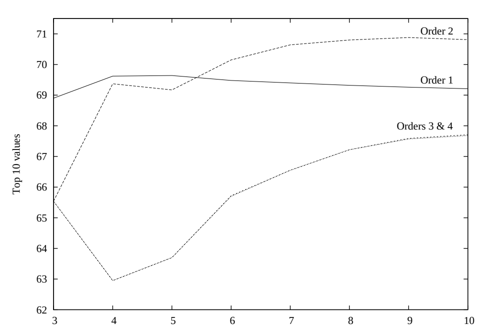
	KMM		**AKMM**		**KSMM**		**AKSMM**	
k	Acc.	Cov.	Acc.	Cov.	Acc.	Cov.	Acc.	Cov.
0	31.88	100	31.88	100.0	31.88	100	31.88	100
1	67.38	96.5	64.83	100.0	69.23	99.9	69.21	100
2	**68.14**	**84.4**	**65.16**	**100.0**	**71.21**	**98.8**	**70.81**	**100**
3	61.82	51.0	61.34	100.0	64.98	77.4	67.71	100
4	60.66	27.8	60.51	100.0	53.69	43.7	67.68	100

We can first notice that, the optimal value of k for KMM is 2 (the recommendation list is computed based on the two previous resources consulted by the active user), which leads to the highest accuracy (68.14). The corresponding coverage value is relatively high (84.4%), but is lower than for KMM of lower order. The accuracy of AKMM (that has a constant coverage) increases according to the value of k, until a value of $k = 2$, then accuracy decreases (as does KMM). Let us notice that the accuracy of AKMM with $k = 2$ is lower than KMM with a similar value of k. That is explained by the fact that 15.6% of recommendations have been computed with KMM of order $k < 2$, that have a lower accuracy.

The right part of Table 1 presents the accuracy and coverage of the KSMM and AKSMM with the maximum distance value set to $D = 10$. The value of D has been fixed to 10 as [6] showed that, on the same corpus accuracy of KSMM increased with the value of D and convergence was reached with a value of $D = 10$. As for KMM and AKMM the optimal value of k is 2 that reaches the best accuracy for both models.

In the whole table, we can notice that low order models are not evolved enough to obtain high accuracy values. At the opposite, high order models are too specific for this corpus and do not lead to high accuracy, we face the data sparsity problem. A value of $k = 2$ seems to be the best tradeoff on this corpus.

Fig. 1. Accuracy of AKSMM according to the distance D and the order value k



In order to study the characteristics of AKSMM, Figure 1 presents the accuracy of AKSMM, according to the distance D and the value of k. We can first notice that the accuracy of several models increase according to the size of the window. In the case of AKSMM of order 2, it increases by more than 5 points when the size of the window grows from 3 to 7, a larger window has no influence of the performance. Convergence is reached with a value of D lower than for KSMM. However, the AKSMM of order 1 reaches its optimal accuracy with an even smaller window size. AKSMM of order 3 and 4 also improve their accuracy according to the distance, but their accuracy is lower than the the AKSMM of order 2, once more due to the data sparsity problem.

So, we can conclude that the AKSMM we propose is promising due to its accuracy, coverage, robustness and complexity.

5 Conclusion and Perspectives

In this paper, we focus on context dependent recommender systems. The AKSMM model we design takes advantage of several state of the art models, expecially of skipped-based Markov models and all-kth-order Markov models. It results in a low-order Markov model that has a high coverage. This model has moreover a low space complexity. Experimentations show that the accuracy of this model outperforms those of the other models, while having a 100% coverage. We show that the accuracy increases according to the size of the history used to perform recommendations.

In a future work, we will test this model on larger corpora, to study the model proposed. Moreover, we envisage to design an alternative to the AKMM by using, when necessary, at the same time several models of different order.

References

1. Mobasher, B.: 3. In: Data Mining for Web Personalization. LNCS 4321 - Brusilovsky, P. and Kobsa, A. and Nejdl, W. (2007) 90–135
2. Fu, X., Budzik, J., Hammond, K.: Mining navigation history for recommendation. In: Proc. of the Intelligent User Interfaces Conference, ACM Press (2000) 106–112
3. Deshpande, M., Karypis, G.: Selective markov models for predicting web-page accesses. In: First SIAM International Conference on Data Mining. (2001)
4. Pitkow, J., Pirolli, P.: Mining longest repeating subsequences to predict world wide web surfing. In: USITS'99: Proceedings of the 2nd conference on USENIX Symposium on Internet Technologies and Systems. (1999)
5. Bonnin, G., Brun, A., Boyer, A.: Using skipping for sequence-based collaborative filtering. In: Proceedings of the International Conference on Web Intelligence. (2008)
6. Bonnin, G., Brun, A., Boyer, A.: A low-order markov model integrating long-distance histories for collaborative recommender systems. In: Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI'09), Sanibel Islands, USA (february 2009) 57–66