INCREMENTAL USER MODELING WITH HETEROGENEOUS USER BEHAVIORS

Rafael Alonso, Philip Bramsen and Hua Li
SET Corporation, a SAIC Company,1005 North Glebe Road, 4th Floor, Arlington, VA 22201, U.S.A.

Keywords: User Modeling, Machine Learning, Implicit Feedback, User.

Abstract: We describe an innovative approach for incrementally learning user interests from multiple types of user behaviours or events. User interests are reflected in the concepts and their relations contained in these events. The concepts and relations form the structural elements of a user interest model. The relevance of each structural element is signified by a weight. Our user modeling algorithm builds dynamic user interest model with two concurrent processes. One process grows the model by intelligently incorporating concepts and relationships extracted from user events. Another process adapts the weights of these model elements by applying a novel combination of two mechanisms: reinforcement and forgetting, both important in modulating user interests. Our modeling algorithm supports incremental and real time modeling, and readily extends to new types of user events. One interesting application of user interest models is to identify a virtual interest group (VIG), which is an ordered set of other system users who exhibit interests similar to those of the target user. As a result, we can evaluate our user modeling algorithm through a VIG identification task. In a formative NIST evaluation using intelligence analysts, we achieved 95% VIG identification precision and recall.

1 INTRODUCTION

In recent years, implicit user modeling has becoming popular for unobtrusively learning user interests and information needs from observed user behaviors (Kelly & Teevan, 2003; Hijikata, 2004; Shen et al. 2005; Teevan et al. 2005). There are many types of user behaviors when a user interacts with an information system. For example, a typical Web user may issue a query, view a page, cut and paste a section of the page, bookmark the page and print it. Clearly, all these actions indicate the user's interests. In addition, the evidence for an interest will be stronger if it is reflected in multiple actions than in only one. We also observe that different types of user actions reflect user interests to different degrees. We introduce the notion of relevance to denote the correspondence between the information in the event and the user's true interests. For example, a keyword appearing in a query reflects more of the user's thinking than if it happens to occur in a document the user read. Similarly, book marking a document highlights the user's interests, whereas deleting it is likely to imply the user's lack of interest.

Thus the primary motivation for our work is the need to model user's interests with heterogeneous user behaviors. Most studies focus on one or two types of user behaviors, namely user query and page view (Mostafa et al. 1997; Pazzani & Billsus, 1997; Kelly & Teevan, 2003; Teevan et al. 2005; Shen et al. 2005).

We also recognize that there are advantages for better user modeling by studying multiple behaviors concurrently. If a user is very interested in a particular topic, her interest will likely to be implicitly expressed in multiple behaviors. Thus one advantage of working with heterogeneous behaviors is different types of user behaviors tend to reinforce or corroborate each other to indicate a particular interest. For example, if a web page is really interesting, the user may read it, bookmark it, save it, and print it. Another advantage lies in the fact that a user's interests can be complex and might be expressed in multiple types of behaviors. In other words, only if we look at different behaviors, do we get a full picture of user's interests. For example, a user is interested in both politics and sports. He likes to read about politics on the Web. When it comes to sports, however, he likes to chat with his buddies. If we only pay attention to either Web page visits or

Alonso R., Bramsen P. and Li H..
INCREMENTAL USER MODELING WITH HETEROGENEOUS USER BEHAVIORS.
DOI: 10.5220/0003062801290135
In *Proceedings of the International Conference on Knowledge Management and Information Sharing* (KMIS-2010), pages 129-135
ISBN: 978-989-8425-30-0

chats, we end up with partial understanding of the user's interests.

Working with heterogeneous behaviors also offers the advantage to fine tune user's interests by integrating both positive and negative feedback from user behaviors. Positive feedback refers to information conveyed by user events that indicate user's interests. Such events include query, cut/paste, reading, saving, printing, positive rating, and selection from a result list. Most studies in the user modeling retrieval literature rely on positive feedback (Mostafa et al. 1997; Pazzani & Billsus, 1997; Kelly & Teevan, 2003; Hijikata, 2004). Negative feedback is information conveyed by user events such as deletion and non-selection that indicate user's lack of interest. Until recently research in user modeling and information retrieval has largely ignored negative feedback (Wang et al.2008; Li et al. 2009). Still, these research efforts have been studying negative feedback in isolation, i.e. not combined with positive feedback. We recognize that both positive and negative feedback are important in modeling user interests. In our work, we have developed an approach that incorporates both types of feedback in a uniform manner.

2 MODELING USER INTERESTS WITH HETEROGENEOUS USER BEHAVIORS

Many types of user behaviors occur when a user interacts with an information system. Clearly, there are various ways to organize them and a couple of researchers have attempted to classify them (Kelly & Teevan 2003; Oard & Kim; 2001). For this work, we adopted the Analysis Log Event (ALE) specification developed by IARPA (Schroh et al. 2009). It is a comprehensive and systematic taxonomy for describing user events commonly occurring in information systems. It is completely defined in XML schema. The ALE specification describes four high-level event classes. The classes relevant to this paper include the following:

- Search – Generated when a user submits a query to a search engine.
- Access – Generated when a user views a URL.
- Retain – Generated when a user keeps information by actions such as copy/paste, bookmarking a web page, printing a document, or inserting evidence into her workspace.
- Focus – Generated when the user interacts with or 'focuses' on an entity.
- Assess – Generated when a user rates a document or when the user links a piece of evidence with a hypothesis.
- Create – Generated when a user creates information object such as a new report or a hypothesis.
- Discard – Generated when a user deletes a document.
- StartApplication – Generated when a user opens an application.

The ALE taxonomy represents the state of the art classification scheme for user behaviors. It encompasses both implicit and explicit feedback behaviors. In particular, the assess events allows the user to provide explicit ratings while the other event classes report various forms of implicit feedback information. In addition, the taxonomy captures both negative and positive feedback behaviors. For example, discard events provide negative feedback whereas search, access, and retain events convey positive feedback.

2.1 Reinforcement and Aging based Modeling Algorithm (RAMA)

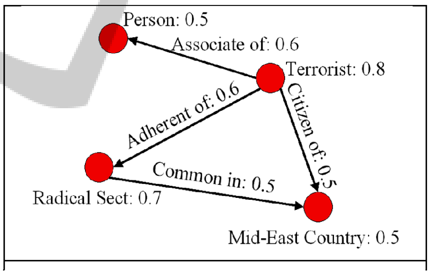

Figure 1: A concept map example.

The input to RAMA is a stream of heterogeneous user behaviors in the form of ALEs generated when a user interacts with an information system. The output are user models represented by a weighted concept map (Alonso & Li, 2005). In this paper, we define a concept map as consisting of concepts, relations, and their associated weights. For convenience, concepts and relations are also referred to as features of the user model. A concept denotes an idea, or a general notion regarding the task. A relation denotes the relationship between two concepts. Concepts and relations carry weights that indicate levels of user interest. In the example shown in Figure 1, the concept map consists of 4 concepts (red nodes) and 4 relations (black arrows). The associated weights for the features are also shown.

At runtime, RAMA algorithm ***incrementally*** processes individual user behavioral events as they arrive to the system. For each user event, we extract the textual content from the event, and perform preprocessing (e.g. removing HTML tags) if needed. A specialized natural language processing (NLP) tool is applied to the textual content to extract an "***information object model***", a data structure that represents the content in the form of a weighted concept map. Consequently the NLP tool that produces the data structure is referred to as "***information object modeler***."

The RAMA algorithm then ages the current user model by applying a forgetting function to all features in current user model, effectively causing their weights to decrease, i.e., user's existing interests will decrease over time. In RAMA, the forgetting function is defined as follows:

$$W' = (1 - A)W$$

The formula above states that every time it is applied, the new weight of a feature in the model will decrease as a linear function of current weight. The amount of decrease is determined by ***A***, which is the ***aging factor*** with value in the range of [0, 1].

The next step of the algorithm is selective ***reinforcement***. A feature of the user model is activated if it is also seen in the information object model. Activated features will have their interest weights positively or negatively reinforced, depending on the nature of the feedback provided by the user event. This step thus provides the mechanism for incorporating both positive and negative feedback in a consistent manner.

Positive reinforcement occurs when the ALE is a positive feedback. An activated feature increases its weight according to the following formula:

$$W' = Max(W + R_f I(1 - W), R_f W_m),$$

where W' is the new weight, W is the current weight, R_f is the relevance for feedback *f*, and ***I*** is ***the reinforcement factor***, which determines the size of the weight increase. The value of *I* falls in the range of [0, 1]. The larger the value for *I* is, the larger the increase. Note that the second term (R_f W_m) in the Max function is the default weight for the current feedback. The Max function is to guarantee the new weight be at least the size of default weight for the given feedback.

Negative reinforcement occurs when the polarity of the feedback is negative. The activated feature is reinforced via a decrease in its weight as follows:

$$W' = W - R_f IW$$

The symbols in the above formula have the same meanings as before. Note the larger the value of the reinforcement factor, the larger the decrease.

Note that in both types of reinforcement, different types of user behaviors affect the weight change via the relevance factor R_f. The larger the value, the bigger the impact on the weight change.

The last step of RAMA grows the user model carefully. Only if the feedback from is positive, do we insert the top-weighted N new concepts and relations from the information object model into the current user model. Then the top N concepts with the highest values are selected. N is an algorithm parameter which can be adjusted depending on the application. The interest weights at insertion will be modulated by the type of their source event.

2.2 Extracting Model Features using Information Object Modelers

Several object modelers have been developed and tested in the NIST evaluation. The two modelers we will focus on in this paper are: 1) Term Frequency (TF); and 2) Text to Specialized Concept Maps (T2SCM).

2.2.1 TF Object Modeler

The document model built with the TF modeler is simply the normalized term frequencies of those terms in the document. To constrain the number of potential terms, a file of common words was used for the TF modeler. This file contains a single long list of words which is intended to represent the most common words in the corpus of training documents. This list of words constitutes the marker set for the TF modeler. This file contained a total of 6,503 words, which was selected from a set of 50,000 documents collected by Oculus using in-domain web queries. Note that only terms are extracted. The relations among them terms are not extracted.

2.2.2 T2SCM Object Modeler

The T2SCM (Text to Specialized Concept Map) tool extracts typed concepts and relations from English text and generates a specialized concept map represented as an XML document. It is based on the open source NLP software GATE (*http://gate.ac.uk/*). The input text is first processed by GATE to extract named entities or annotations. The entities are then processed to derive typed concepts and proximity-based relations (e.g. cooccurring in the same sentences). The weight for the concepts and relations are determined by the number of their occurrences normalized by the size of the document.

3 EVALUATION

The evaluation of user modeling algorithm is based on our Model-Guided Inquiry (MGI) framework, which uses a user model to shape the behavior of an information system (Alonso et al. 2003; Alonso & Li, 2005a, 2005b). Such systems are termed model-guided systems. In this framework, the value of user modeling is not in the models themselves, but rather in new capabilities they enable in an information system. The benefits of model-guided inquiry have been demonstrated in our prior work. In one study, it was used to personalize a search for obsolete replacement parts (Alonso et al. 2003). In another, encouraging results were obtained by using models to guide a text search mechanism based on swarm intelligence (Alonso & Li, 2005b). Furthermore, user models were also leveraged to avoid biases in analytic work (Alonso & Li, 2005a). Last but not the least, user models were used for automatic query generation and for augmenting or contextualizing user queries (Alonso & Li, 2005a).In this work, the user model is used to guide the identification of a virtual interest group (VIG), which refers to the virtual community of users who exhibit interests most similar to the target user. In the literature, there are many approaches for discovering VIGs (Guy et al. 2009; McDonald, 2003; Zhang et al. 2007). Here we derive the VIG by comparing the similarity of user models. It is reasonable to assume that the more accurate the user models, the more accurate the VIGs will be. Thus the problem of evaluating the performance of the user modeling translates into one of judging the accuracy of VIG.

3.1 VIG Identification

The insight for this algorithm comes from the observation that even in the course of a single analysis task, the user will explore different aspects of the problem overtime. In other words, the user will display slightly different interests at different times. To capture the shifting of interests under a task, it makes sense to build mini-models for different phases or time segments rather than one all-encompassing lifetime model. We refer the model for a specific time segment as a **segment model**. Along the same line of reasoning, it makes sense to compare all the segment models of two users in order to determine if they are similar to each other.

3.2 NIST Experiment Setup

This evaluation by NIST was referred to as the Recommender2 study. Seven Navy Reserve analysts were recruited by NIST for the study. Each analyst spent two 8-hour days at NIST (some spent additional time prior to the final experiment to help with system tuning) and worked four tasks. Each task was 3.5 hours, except the training task of the first half-day, which had nearly the same schedule as the other tasks. Tasks focused on collecting hypotheses in response to the task directive, based on evidence found with the information system. For each task, each analyst used a different user handle. In the analysis, a "*user*" is a user session with one handle. Altogether, we have 7 * 4 = 28 user sessions. We knew the number of tasks and the details of the training task, but did not know specifics of the other tasks.

During the experiment, ALEs were captured and stored in a web service-based repository called analysis log service (ALS), which was developed and hosted by Oculus. The ALE data was made available to us for VIG analysis. Although our algorithms run in real time, VIG analysis was run after the experiment, as our VIG predictions were to be tested against the tasking assignments and not provided to the analysts.

3.2.1 Data for VIG Identification

In our first analysis, we performed user modeling and VIG identification using the 4,463 ALEs generated from the Recommender2 study alone. In a second analysis, we incorporated 28,249 additional archived ALEs from an earlier research program and performed VIG analysis with the combined set of ALEs. The results for the second analysis were similar to the first, and were not shown for reasons of space limitation. The Recommender2 study generated the following ALEs (the number indicates the number of ALEs for that type): 178 Search, 1180 Access, 599 Retain, 43 Focus, 1392 Assess, 928 Create, 103 Discard, and 40 StartApplication.

For the VIG analysis, we used two data segmentation intervals: 30 and 10 minutes. As mentioned in the VIG identification algorithm, segmentation refers to the slice of time on which one subject was compared to the other subjects. On average, subjects generated 2 ALEs (analytic logging events) every minute. Therefore, 10 minutes

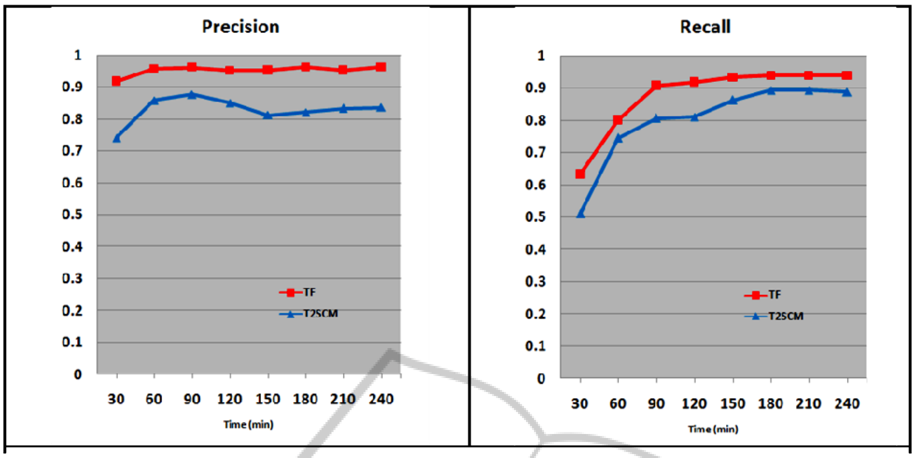

Figure 2: VIG metrics -- 30 minute segments for 28 sessions.

corresponds roughly to 20 ALEs, while 30 minutes represents 60 ALEs worth of activity. In the case of 30 minute segmentation, we built up to 8 half-hour segment models per user. As a result, we had a total of 168 segment models for task identification.

3.2.2 Gold Standards and Performance Metrics

The gold standards (i.e. the ground truth) were created by NIST for the Recommender2 data. They were unknown to us until after the evaluation was complete. The gold standards contained the true VIGs for each user. They were used by NIST to compute performance metrics.

Precision and Recall and equal-weighted F-values were used as the main outcome measures. Precision is defined as hits/(hits + false alarms), i.e. (#correct answers)/(# total answers). Recall is defined as hits/(hits + misses), i.e. (# correct answers)/(# correct answers).

3.3 Results

Figure 2 shows the results at segmentation sizes of 30-minute. The first thing to observe in this figure is that different object modelers (TF and T2SCM) had clearly different abilities to detect VIG's. Secondly, TF, the putatively simplest of the methods, was best in terms of all three metrics. Third, a comparison between segment sizes of 30-minute and 10-minute (results not shown) leads to the conclusion that 10-minute time slices are not as effective as 30-minute slices under the conditions of this study. It is intriguing, however, to speculate that 10-minute segments appear to be nearly as good as 30-minute segments in the early part of the time series. Thirty minute segments are clearly better as the user model grows over time. Finally, the absolute values for precision and recall for TF, and therefore for F-value, are quite extraordinary.

In summary, in this NIST evaluation, VIG detection was tested against gold standards. VIG's were identified well in the set of 28 sessions generated by our current 7 subjects. The best object modeler (TF) achieved 95% precision and nearly 95% recall.

4 CONCLUSIONS AND FUTURE WORK

We describe an innovative approach for incrementally learning user interests from multiple types of user behaviors or events. User interests are reflected in the concepts and their relations contained in these events. The concepts and relations form the structural elements of a user interest model. The relevance of each structural element is signified by a weight. Our user modeling algorithm builds dynamic user interest model with two concurrent processes. One process grows the model by intelligently incorporating concepts and relationships extracted from user events. Another process adapts the weights of these model elements by applying a novel combination of two mechanisms: reinforcement and forgetting, both important in modulating user interests. Our modeling algorithm supports incremental and real time modeling, and readily extends to new types of user events. One interesting application of user interest models is to identify a VIG, which is an ordered set of other system users who exhibit interests similar to those of the target user. As a result, we can evaluate our user modeling algorithm through a VIG identification task. In a formative NIST evaluation using intelligence analysts, we achieved 95% VIG identification precision and recall.

In the evaluation, we have explored a couple of other information object modelers especially those based on topic distributions such as Latent Dirichlet Allocation (LDA) (Wang et al. 2007). The results were not conclusive due to insufficient time and resource for tuning. In the future we would like to study them further. It would also be interesting to acquire the relevance assignments for different types of ALEs automatically by restricting inputs to VIG algorithm to single ALE types and compare the impacts on VIG identification performance. Lastly, we would like to apply clustering algorithms to the task identification problem using the generated segment models.

ACKNOWLEDGEMENTS

This study was supported and monitored by the Intelligence Advanced Research Projects Activity (IARPA) under the CASE MASTER project with contract number FA8750-06-C-0193. The views, opinions, and findings contained in this report are those of the authors and should not be construed as an official IARPA position, policy, or decision, unless so designated by other official documentation.

We would like to thank the NIST team headed by Emile Morse for conducting the formative evaluation of our work. We are also indebted to New Vectors and Oculus teams for their collaboration and software support. We also owe our gratitude to FairIsaac and BAE teams for providing the object modeler support and software.

REFERENCES

Alonso, R., Bloom, J.A., Li, H. and Basu, C., 2003. An adaptive nearest neighbor search for a parts acquisition ePortal *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, Washington, D.C.

Alonso, R. and Li, H., 2005a. Combating Cognitive Biases in Information Retrieval. in *First International Conference on Intelligence Analysis Methods and Tools*, McLean, VA, USA.

Alonso, R. and Li, H., 2005b. Model-guided information discovery for intelligence analysis *Proceedings of the 14th ACM international conference on Information and knowledge management*, ACM, Bremen, Germany.

Hijikata, Y., 2004. Implicit user profiling for on demand relevance feedback Proceedings of the 9th international conference on Intelligent user interfaces, ACM, Funchal, Madeira, Portugal.

Guy, I., Ronen, I. and Wilcox, E., 2009. Do you know?: recommending people to invite into your social network Proceedings of the 13th international conference on Intelligent user interfaces, ACM, Sanibel Island, Florida, USA.

Kelly, D. and Teevan, J., 2003. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, *37* (2). 18-28.

Li, Y., Algarni, A., Wu, S.-T. and Xue, Y., 2009. Mining Negative Relevance Feedback for Information Filtering *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, IEEE Computer Society.

McDonald, D.W., 2003. Recommending collaboration with social networks: a comparative evaluation *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, Ft. Lauderdale, Florida, USA.

Mostafa, J., Mukhopadhyay, S., Palakal, M. and Lam, W., 1997. A multilevel approach to intelligent information filtering: model, system, and evaluation. *ACM Trans. Inf. Syst.*, *15* (4). 368-399.

Oard, D.W. and Kim, J., 2001. Modeling information content using observable behavior. in *Proceedings of the 64th Annual Meeting of the American Society for Information Science and Technology*, 38-45.

Pazzani, M.J. and Billsus, D., 1997. Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, *27*. 313-331.

Schroh, D., Bozowsky, N., Savigny, M. and Wright, W., 2009. nCompass Service Oriented Architecture for Tacit Collaboration Services *Proceedings of the 2009 13th International Conference Information Visualisation - Volume 00*, IEEE Computer Society.

Shen, X., Tan, B. and Zhai, C., 2005. Implicit user modeling for personalized search *Proceedings of the 14th ACM international conference on Information and knowledge management*, ACM, Bremen, Germany.

Teevan, J., Dumais, S.T. and Horvitz, E., 2005. Personalizing search via automated analysis of interests and activities *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, Salvador, Brazil.

Wang, X., Fang, H. and Zhai, C., 2008. A study of methods for negative relevance feedback *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, Singapore, Singapore.

Wang, X., McCallum, A. and Wei, X., 2007. Topical N-grams: Phrase and Topic Discovery, with an Application to Information Retrieval. in *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*.

Zhang, J., Ackerman, M.S., Adamic, L. and Nam, K.K., 2007. QuME: a mechanism to support expertise finding in online help-seeking communities *Proceedings of the 20th annual ACM symposium on User interface software and technology*, ACM, Newport, Rhode Island, USA.