Singapore Management University
Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems
School of Computing and Information Systems

5-2010

Learning user profiles for personalized information dissemination

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Christine TEO

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Databases and Information Systems Commons, and the Information Security Commons

Citation
TAN, Ah-hwee and TEO, Christine. Learning user profiles for personalized information dissemination. (2010). *Proceedings of the International Joint Conference on Neural Networks (IJCNN '98): Alaska, May 4-9*. 183-188.
Available at: https://ink.library.smu.edu.sg/sis_research/6452

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

Learning User Profiles for Personalized Information Dissemination

Ah-Hwee Tan, Christine Teo
Kent Ridge Digital Labs, 21 Heng Mui Keng Terrace, Singapore 119613
Email: ahhwee@krdl.org.sg, bkteo@krdl.org.sg

Abstract

Personalized information systems represent the recent effort of delivering information to users more effectively in the modern electronic age. This paper illustrates how a supervised Adaptive Resonance Theory (ART) system, known as fuzzy ARAM, can be used to learn user profiles for personalized information dissemination. ARAM learning is on-line, fast, and incremental. Acquisition of new knowledge does not require re-training on previously learned cases. ARAM integrates both user-defined and system-learned knowledge in a single framework. Therefore inconsistency between the two knowledge sources will not arise. ARAM has been used to develop a personalized news system known as PIN. Preliminary experiments have verified that PIN is able to provide personalized news by adapting to user's interests in an on-line manner and generalizing to new information on-the-fly.

1. Introduction

Information overflow is an inevitable problem in the modern life. Since the popularization of World Wide Web (WWW), even a casual computer user now has to deal with an uncontrolled flood of digital information. Methods are needed to handle the information overloading problem by filtering away those information that a user deems irrelevant. Information filtering allows the user to focus his/her energy on important information and therefore improves productivity.

Personalized information systems represent the recent effort of disseminating information to users more effectively. A typical personalized system builds a profile for each prospective user so that only the information most relevant to a user is identified and presented. Fishwrap [8], for example, adopted an information retrieval approach by indexing or pre-categorizing each piece of information. The indexing process was time consuming and reindexing was needed whenever a new piece of information became available.

Others explore more efficient methods in which each information piece can be processed individually. Most of the systems perform rule-based learning in order to obtain data correlations and parameters of a user model from the input data stored over a period of time. Many build a dynamic decision tree based on the constantly changing input data in order to deduce rules for agent behavior. Some also incorporate greedy attribute selection for the purpose of building the decision tree. For example, Mitchell [10] employed the ID-3 decision tree to learn the new rules overnight and merge with the old rules. Besides that on-line incremental learning was not possible, the simple merging of the new and old rules, as reported, had produced "disappointing" generalization performance. Maes [9] adopted a Nearest Neighbour approach which stored past cases as examples and past cases with wild card features as rules. Given a new case, the system acted according to the closest example/rule. There was however no attempt to generalize rules from examples.

This paper explores how a neural network model, known as the Adaptive Resonance Associative Map (ARAM) [13], can be used to learn user personal profile. ARAM organizes information into categories or clusters based on their semantic similarities. A user profile is modeled by associating each of the information categories, characterized by semantic features, with a relevance factor indicated by the user.

ARAM has been incorporated into a personalized news system known as *Personalized Information Network* (PIN). PIN retrieves on-line news articles from World Wide Web (WWW) and provides customized news to registered users. Preliminary experiments with PIN have indicated that it provides personalized news by adapting to user's interests in an on-line manner and generalizing to new information on-the-fly.

The remaining sections of this article are organized as follows. To make this article self-contained, section 2 gives a brief review of ARAM and its algorithm. Section 3 describes ARAM properties in the perspective of learning personal profile. Section 4 presents PIN, a personalized news system, and its three major components, namely the retrieval agent, the learning agent, and the news browser. The final section reports the experimental results.

2. A Review of ARAM

Adaptive Resonance Associative Map (ARAM) belongs to a family of neural network systems known as the supervised Adaptive Resonance Theory (ART) networks [3, 4]. In contrast to slow learning gradient descent neural networks, supervised ART systems offer learning that is fast, stable, and incremental. The learning capability of such method has been evaluated on a number of well

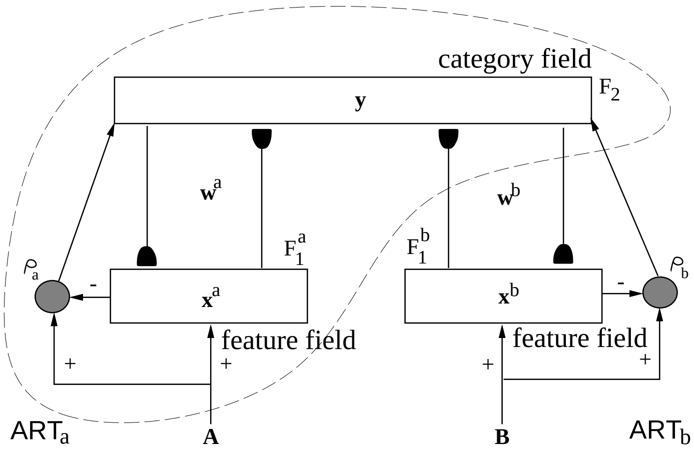

Fig. 1: The Adaptive Resonance Associative Map architecture.

known benchmark problems [7, 13, 14] in terms of predictive accuracy and learning efficiency.

Whereas a similar supervised Adaptive Resonance Theory (ART) system, ARTMAP [3], consists of two ART modules interconnected by an inter-ART associative map field, ARAM can be visualized as two overlapping ART modules sharing a single category field. In an ARAM network (Figure 1), the category field F_2 receives bottom up activities from the two feature fields F_1^a and F_1^b. Thus, an F_2 category node learns to encode a complete pattern pair.

Given a pair of patterns, the category field F_2 selects a winner that receives the largest overall input from the feature fields F_1^a and F_1^b. The winning node selected in F_2 then triggers a top-down priming on F_1^a and F_1^b, monitored by separate reset mechanisms. Code stabilization is ensured by restricting encoding to states where resonances are reached in both modules. By synchronizing the unsupervised categorization of two pattern sets, ARAM learns supervised mapping between the pattern sets. Due to the code stabilization mechanism, fast learning in a real-time environment is feasible. As the network structure and operations are symmetrical, associative recall can be performed in both directions.

The ART modules used in ARAM can be ART 1 [1], which categorizes binary patterns, or analog ART modules such as ART 2 [2], ART 2-A [5], and fuzzy ART [6], which categorize both binary and analog patterns. The fuzzy ARAM model, that is composed of two overlapping fuzzy ART modules (Figure 1), is described below.

Input vectors: Normalization of fuzzy ART inputs prevents category proliferation. The F_1^a and F_1^b input vectors are normalized by complement coding that preserves amplitude information. Complement coding represents both the on-response and the off-response to an input vector $\mathbf{a}$. The complement coded F_1^a input vector $\mathbf{A}$ is a 2M-dimensional vector

$$\mathbf{A} = (\mathbf{a}, \mathbf{a}^c) \equiv (a_1, \ldots, a_M, a_1^c, \ldots, a_M^c) \tag{1}$$

where $a_i^c \equiv 1 - a_i$. Similarly, the complement coded F_1^b input vector $\mathbf{B}$ is a 2N-dimensional vector

$$\mathbf{B} = (\mathbf{b}, \mathbf{b}^c) \equiv (b_1, \ldots, b_N, b_1^c, \ldots, b_N^c) \tag{2}$$

where $b_i^c \equiv 1 - b_i$.

Activity vectors: Let $\mathbf{x}^a$ and $\mathbf{x}^b$ denote the F_1^a and F_1^b activity vectors respectively. Let $\mathbf{y}$ denote the F_2 activity vector. Upon input presentation, $\mathbf{x}^a = \mathbf{A}$ and $\mathbf{x}^b = \mathbf{B}$.

Weight vectors: Each F_2 category node j is associated with two adaptive weight templates $\mathbf{w}_j^a$ and $\mathbf{w}_j^b$. Initially, all category nodes are uncommitted and all weights equal ones. After a category node is selected for encoding, it becomes *committed*.

Parameters: The ARAM dynamics is determined by choice parameters $\alpha_a > 0$ and $\alpha_b > 0$; learning rate parameters $\beta_a \in [0,1]$ and $\beta_b \in [0,1]$; vigilance parameters $\rho_a \in [0,1]$ and $\rho_b \in [0,1]$; and a control parameter $\gamma \in [0,1]$.

Code activation: Given activity vectors $\mathbf{x}^a$ and $\mathbf{x}^b$, for each F_2 node j, the choice function T_j is computed by

$$T_j = \gamma \frac{|\mathbf{x}^a \wedge \mathbf{w}_j^a|}{\alpha_a + |\mathbf{w}_j^a|} + (1-\gamma) \frac{|\mathbf{x}^b \wedge \mathbf{w}_j^b|}{\alpha_b + |\mathbf{w}_j^b|}, \tag{3}$$

where the fuzzy AND operation $\wedge$ is defined by

$$(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i), \tag{4}$$

and the norm $|.|$ is defined by

$$|\mathbf{p}| \equiv \sum_i p_i \tag{5}$$

for vectors $\mathbf{p}$ and $\mathbf{q}$.

Code competition: All F_2 nodes undergo a code competition process. The winner is indexed at J where

$$T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}. \tag{6}$$

When a category choice is made at node J, $y_J = 1$; and $y_j = 0$ for all $j \neq J$.

Template matching: Resonance occurs if the *match functions* m_J^a and m_J^b meet the vigilance criteria in their respective fields:

$$m_J^a = \frac{|\mathbf{x}^a \wedge \mathbf{w}_J^a|}{|\mathbf{x}^a|} \geq \rho_a \qquad \text{and} \qquad m_J^b = \frac{|\mathbf{x}^b \wedge \mathbf{w}_J^b|}{|\mathbf{x}^b|} \geq \rho_b. \tag{7}$$

Learning then ensues, as defined below. If any of the vigilance constraints is violated, mismatch reset occurs in which the value of the choice function T_J is set to 0 for the duration of the input presentation. The search process repeats to select another F_2 node J until resonance is achieved.

Template learning: Once a node J is selected, the weight vectors $\mathbf{w}_J^a$ and $\mathbf{w}_J^b$ are modified by the learning rule

$$\mathbf{w}_J^{k(\text{new})} = (1-\beta_k)\mathbf{w}_J^{k(\text{old})} + \beta_k(\mathbf{x}^k \wedge \mathbf{w}_J^{k(\text{old})}) \tag{8}$$

for $k = a$ and b.

3. ARAM for Learning Personal Profiles

We describe the key properties of ARAM in the perspective of learning personal profile below.

3.1. Fast, Incremental, and On-line Learning

ARAM is designed to learn multi-dimensional mappings in a fast and incremental mode. Learning of new cases does not require re-learning of previously learned cases as in the state-of-the-art gradient descent learning systems [11, 12, 15]. In addition, ARAM learning is on-line as it does not need to go through the training examples many times to learn the knowledge. These learning properties have endowed ARAM with the capability of learning personal profile on-the-fly as and when new cases are presented.

3.2. Generalization

To operate in a real world environment, a learning personal agent must perform fuzzy or approximate reasoning. ARAM's feature matching and category competition properties allow fuzzy matching of multi-dimensional feature vectors. Learning of a user's interests in today's news allows generalization to the user's interests in tomorrow's news.

3.3. Integration of Rules and Examples

ARAM represents a user's interest profile by a set of recognition categories, each associating a set of conjunctive features to a relevance factor. As the knowledge structure is compatible with rule-based knowledge representation, user-defined interest keywords can be readily translated into the recognition categories of an ARAM system. Initializing an ARAM network with pre-existing knowledge has been proven to improve learning efficiency and predictive accuracy. Subsequent user feedback on individual pieces of information can be used to refine the ARAM network. Through the refinement process, ARAM learns interest terms that are not explicitly mentioned by the user. As both user-defined and system-learned knowledge are represented in a single system, any inherent conflict or inconsistency can be resolved through ARAM's code competition and template learning mechanisms.

3.4. Real Time Adaptation

A user's interests often change over time. ARAM tracks the changes of a user's interests dynamically by creating new recognition categories and modifying weight templates. Recognition categories representing outdated knowledge gradually will become inactive. Inactive categories can be removed by a pruning algorithm [7].

3.5. Pruning and Knowledge Interpretation

Besides removing outdated categories, pruning also helps to keep a user profile within a reasonable size. Network pruning is important in a scaled-up system in which computation and memory efficiencies are major concerns.

As each recognition category or cluster learned by ARAM corresponds to a IF-THEN rule, an ARAM network can be translated readily to a set of IF-THEN rules. Pruning plays an important role here to reduce the size of the rule set for interpretation of learned knowledge and analysis of system decisions.

4. PIN: A Personalized News System

The *Personalized Information Network* (PIN) system aims to tailor personalized multimedia news information to the individual subscribers based on advanced information retrieval technologies. It assists one to mine and navigate through massive repositories of news available daily.

Personal profiles of users are created or updated over times to identify specific interests, e.g., news about particular companies and industries. Based on the learned user profiles, PIN's interactive news browser presents personalized news categories and ranked lists of news titles in decreasing order of relevance. One can further refine his personalized profile by selecting and rating news which are relevant or interesting over the interactive interface.

PIN (Figure 2) consists of three main subsystems. They include a retrieval agent, that performs news searching and retrieval across World-Wide Web; a personal learning agent, that performs profile learning and information filtering; and a personalized news browser. We describe each of these subsystems in the following sections.

4.1. The Retrieval Agent

The retrieval agent navigates through the WWW and downloads news source documents from news stands web sites. Currently, the retrieval agent downloads from a local news site, namely "The Straits Times Interactive" (http://www.asia1.com.sg/straitstimes).

To retrieve articles from a news stand web site, the news stand home page is first accessed via its URL. This index document often contains *links* that allow objects on one page to point to other document pages. The document pages can comprise of text (HTML or plain) and/or image content (GIF or JPEG). A news source URL acts as the root for a hypertext graph to be explored. Every links that are embedded in HTML <href> element tags in a page leads to more pages that need to be visited. A sequential recursive page-visiting algorithm is employed to extract the hypertext links and download the relevant HTML documents. Steps are taken to avoid pages that are visited already.

After downloading, HTML documents with text content are further parsed to separate text contents from the HTML elements. These text documents are required for content analysis by the learning personal agent. Other

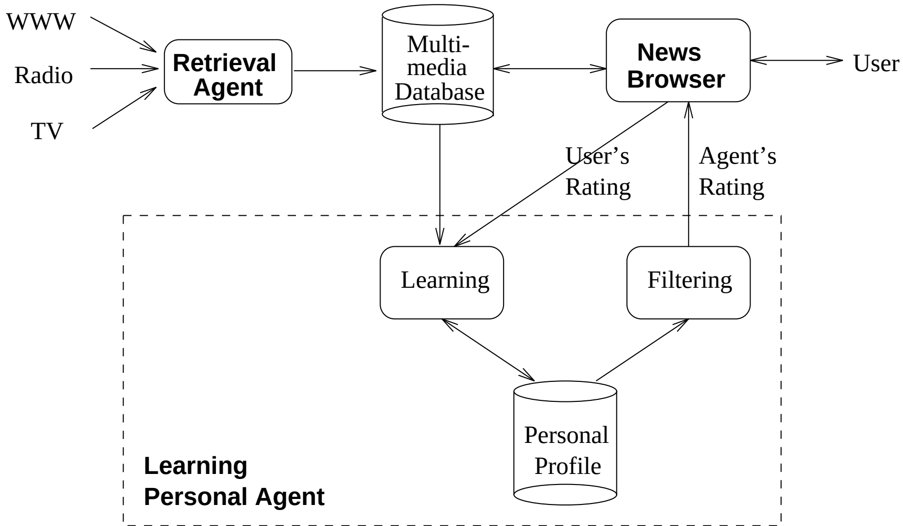

Fig. 2: The PIN's system configuration.

information, such as the document file name and the news title, are also captured for presentation on the news browser.

4.2. The Learning Personal Agent

4.2.1. Profile Initialization

PIN allows a user to define his/her profile by one or more interest terms, each of which consists of one or more keywords. Some examples of the interest terms include "Microsoft Products", "Java Programming", and "Properties". The learning agent maintains a keyword list as the features of news articles. If the user defines a new keyword, it will be added to the keyword list automatically.

During profile initialization, each user-defined interest term is converted into a M-dimensional vector $\mathbf{a}$ and a 1-dimensional vector $\mathbf{b}$, where M is the number of entries in the keyword list. Specifically, given an interest term,

$$x_1, x_2, \ldots, x_m, \tag{9}$$

the algorithm derives a relevance vector $\mathbf{b} = (1)$ and a feature vector $\mathbf{a}$, such that for each index $j = 1, \ldots, M$,

$$a_j = \begin{cases} 1 & \text{if } e_j = x_i \text{ for some } i \in \{1, \ldots, m\} \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where e_j is the jth attribute in the keyword list. Note that the relevance value is always 1 as the user-defined interest terms should, of course, be highly relevant to the user.

The vector pairs $\mathbf{a}$ and $\mathbf{b}$ are then complement coded to form the ART$_a$ input vector $\mathbf{A}$ and ART$_b$ input vector $\mathbf{B}$ respectively (section 2). During profile initialization, the ARAM vigilance parameters are $\rho_a = \rho_b = 1$ so that each distinct pair of vectors $\mathbf{A}$ and $\mathbf{B}$ is encoded by a recognition category.

Table: 1: Five levels of user feedback and their corresponding relevance factors.

Feedback	Relevance Factor
Highly relevant	1.00
Relevant	0.75
Don't Know	0.50
Irrelevant	0.25
Highly Irrelevant	0.00

4.2.2. Feature Extraction

The text documents downloaded by the retrieval agent are subsequently parsed for feature extraction. We use a morphological analyzer to identify the part-of-speech of each word. To reduce complexity, only noun terms are extracted for further processing. For each article, the root word or *stem* of each noun term is matched against a *keyword list* maintained by the learning agent. Whenever a match is found in the keyword list, the frequency count of the corresponding keyword (c_j) is incremented by 1. The M-dimensional feature vector $\mathbf{a}$ is then computed by normalizing the keyword frequency count

$$a_j = c_j / c_J \quad \text{where} \quad c_J > c_j \;\; \forall j \neq J. \tag{11}$$

4.2.3. Profile Learning

Users provide feedback by rating the articles that they read. To make the system user-friendly, users only need to give their feedback by choosing one of the five feedback icons. The feedback is converted to a real-valued relevance factor ($b_1 \in [0, 1]$) for internal processing. Table 1 lists the five types of user feedback and their corresponding relevance factors.

To learn a user's feedback on an article, the feature vector $\mathbf{a}$ and the relevance vector $\mathbf{b}$ are complement coded to form the ART$_a$ input vector **A** and ART$_b$ input vector **B** respectively, before they are presented to the user's ARAM profile network.

4.2.4. Information Filtering

Before preparing a news session for a user, the learning personal agent goes through all the downloaded documents to evaluate their relevance. For each article, the complement coded feature vector **A** is presented to the user's ARAM profile network as the ART$_a$ input vector. The ART$_b$ input vector **B** is set to (0.5,0.5). During prediction, the ARAM profile network performs code activation and competition as in learning. However, template matching only checks for the matching resonance in the ART$_a$ module. If resonance occurs, the ART$_b$ activity vector, which indicates the system generated relevance factor, is given by

$$\mathbf{x}^b = \mathbf{w}_J^b. \tag{12}$$

4.3. The Personalized News Browser

4.3.1. The Profile Manager

PIN maintains a personal profile for each of its users. First time user will need to register with his login name, password, and profile to subscribe to this news service. The user information are stored in the PIN server. Subsequent logon by the user with his login name and password enables it to load the correct user profile upon successful logon verification.

PIN assists a user to filter massive news information based on his/her specified personal profile. A user profile consists of background, interests, and preferences. All three sub-profiles are reconfigurable.

(a) *Background:* A user provides background information, such as his/her home town, country of residence, nationality, business, occupation, and company name etc. The background information contributes to the implicit interest terms of the users.

(b) *Interests:* A user can specify his/her interests by supplying one or more interest terms. An interest term is a conjunction of one or more keywords.

(c) *Preferences:* A user can specify his/her preferences of using PIN for news reading. Some of the news browser's environment parameters are

- News sources (*Paper*, *Radio*, or *Video*),
- News categories,
- View mode (*All*, *Normal*, or *Brief*)

4.3.2. The Personalized News Browser

The news browser gives users full control to hop across the available news categories to select news of their choices. Within each category, a catalog of news titles is presented. The entries are sorted by the relevance factors derived by the learning personal agent. To read an article, users simply click on its title. The actual

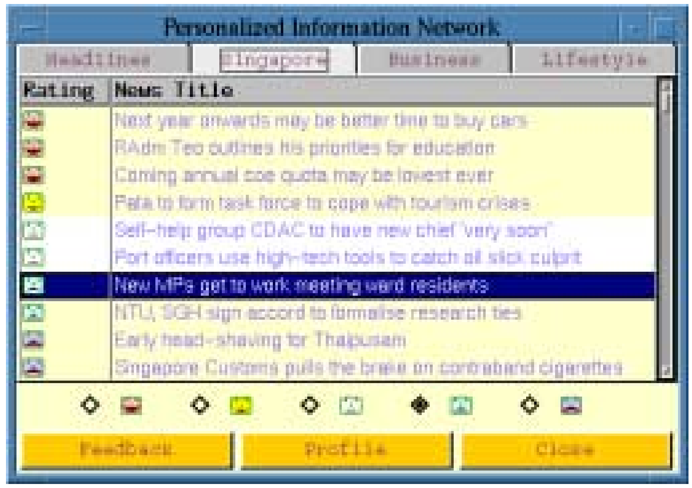

Fig. 3: The personalized news browser.

article at the original sites is then shown on a separate window. To help a user in navigation, articles already read by the user will be highlighted.

Each title in the catalog is attached with a feedback icon which indicates the system rating of the article based on the modeled user profile. A user can provide feedback by changing the face icon to indicate his/her own rating of the article. For user-friendliness, we have adopted a five level rating scheme. In addition to explicit feedback, we also plan to incorporate implicit feedback mechanism which observes the reader's viewing habits to determine his/her untold interests. Specifically, the order that a user browses through a electronic newspaper and the time that he/she spents on a news article (relative to the length of the article) give indications of his/her interests. Based on these implicit feedback, the system can estimate the relevance of a news article to a user, which can then be confirmed or overridden by the user explicitly. To take care of the situations where a user leaves the computer for some other tasks while reading an article, explicit confirmation from the user is always required for those articles with especially long reading time. For machines with face detection or recognition functions, the collection of implicit feedback can be done more gracefully, without the need of explicit confirmation.

5. Experiments

PIN was evaluated, for the first instance, on its capability to learn the profile of an artificial subject *SG* modeled after a set of eleven rules. Some examples of the rules were listed below.

IF Housing or Properties THEN highly relevant
IF Law or Politics THEN highly irrelevant

Twenty-two days of news from Straits Times Interactive (with an average of 80 articles per day) were used in the experiment. Based on the artificial subject *SG*, we assign a relevance factor to each of the news articles.

These relevance factors with their corresponding articles serve as both the testing and training data for PIN.

The performance index used in the experiments is the daily *Mean squared error*

$$MSE = \frac{\sum_j (t_j - p_j)^2}{S} \geq 0, \qquad (13)$$

where t_j and p_j is the target and predicted relevant factors for article j and S is the number of articles in the day.

For each day, PIN was first asked to provide relevance factors for the articles of the day. The predicted relevance factors together with the relevance factors provided by the subject *SG* are used to compute a MSE. The MSE is called *Test MSE* as it reflects the system's actual performance in real time.

Then PIN is trained with the SG's relevance factors and evaluated again. Note that we only allow the system to go through the training data once to simulate the on-line training environment. The MSE obtained after training is called *Train MSE* as it indicates how well the system learns on-the-fly.

Figure 4 summarizes the experimental results over a period of twenty two days of learning and predicting. Within any single day, the *Train MSE* is always smaller than the *Test MSE* and is often zero. This shows that the system is able to perform on-line learning effectively. Furthermore, compared with the baseline MSE generated by a naive system (which predicts *don't know* all the times), the *Test MSE* become very small almost immediately after the second day. This indicates that the system has learned *SG*'s personal profile in a very short time and generalized to provide recommendation close to what *SG* prefers.

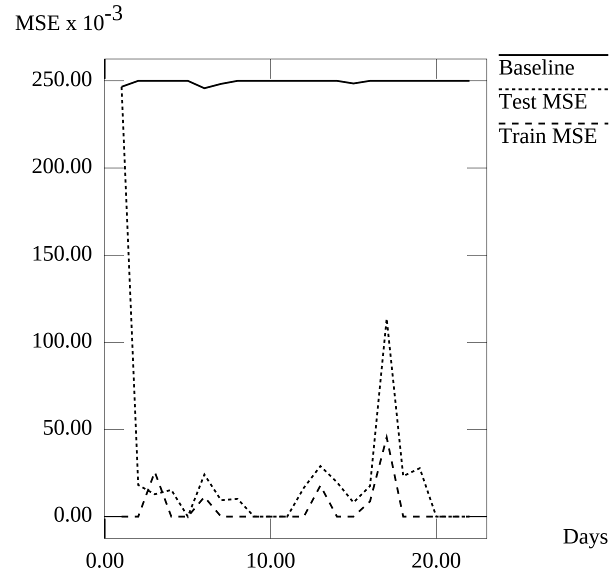

Fig. 4: PIN's learning and predicting performance comparing with the baseline performance by a naive system which performs no personalization.

References

[1] G. A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115, 1987.

[2] G. A. Carpenter and S. Grossberg. ART 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26:4919–4930, 1987.

[3] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3:698–713, 1992.

[4] G. A. Carpenter, S. Grossberg, and J. H. Reynolds. ARTMAP: Supervised real time learning and classification by a self-organizing neural network. *Neural Networks*, 4:565–588, 1991.

[5] G. A. Carpenter, S. Grossberg, and D. B. Rosen. ART 2-A: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:493–504, 1991.

[6] G. A. Carpenter, S. Grossberg, and D. B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[7] G. A. Carpenter and A. H. Tan. Rule extraction: From neural architecture to symbolic representation. *Connection Science*, 7(1):3–27, 1995.

[8] P. R. Chesnais, M. J. Mucklo, and J. A. Sheena. The Fishwrap personalized news system. In *Proceedings, 1995 IEEE 2nd International Workshop on Community Networking Integrating Multimedia Srvcs to the Home*, pages 1–7, 1995.

[9] P. Maes. Agents that reduce work and information overload. *Communications of ACM*, 37(7):48–53, 1994.

[10] T. Mitchell, R. Caruana, D. Freitag, J. Mcdermoot, and D. Zabowski. Learning personal assistants. *Communications of ACM*, 37(7):80–91, 1994.

[11] D. B. Parker. Learning logic, 1982. Invention report, S81-64, File 1, Office of Technology Licensing, Stanford University.

[12] D. E. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the microstructures of Cognitions*, pages 318–362. Cambridge, MA: MIT Press, 1986.

[13] A. H. Tan. Adaptive Resonance Associative Map. *Neural Networks*, 8(3):437–446, 1995.

[14] A. H. Tan and L. N. Teow. Learning by supervised clustering and matching. In *Proceedings, 1995 IEEE International Conference on Neural Networks, Perth*, volume I, pages 242–246, 1995.

[15] P. Werbos. *Beyond Regression: New tools for prediction and analysis in the behavioral sciences.* PhD thesis, Department of Computer Science. Harvard University, Cambridge, MA, 1974.