Nested State-Transition Graph Data Sequencing Model With Hierarchical Taxonomy through Radix Coding

JIA-SHENG HEH, SHEIN-YUN CHENG AND CHANG-KAI HSU
Department of Information and Computer Engineering
Chung Yuan Christian University
Chungli, 320 Taiwan
E-mail: jsheh@ice.cycu.edu.tw

The Internet enables all electronic information to be connected through communication networks. Users access these Internet resources with different behavior models. This paper proposes a systematic data mining approach to studying users' Internet resource access actions to find out behavior models as state-transition graphs. With this state-transition graph model, it is possible to predict the future behaviors of different user communities.

A series of Internet resource access actions are stored in a database of [user, resource-access-action, time] records, indicating that the user accesses the resource at the recorded time. Such access actions are treated as basic behavior elements and form an action hierarchy which possesses different levels of radix codes. For every user, the data sequence is divided into a series of transactions, and all the actions in a transaction constitute a special behavior pattern, called (inter-transaction) behavior. The behavior codes can be aggregated from their action codes and then form a behavior hierarchy. Users can be classified into communities and subgroups by their aggregated behavior codes and behavior transitions. Each community and subgroup has its own behavior models, formulated as a state-transition graph with behavior states and transition probability between behaviors. The overall mining process has been computerized and is validated here by two examples. The first example uses simulated sequential data to show how the AprioriAll algorithm and the proposed algorithm can be combined to construct a set of nested state-transition graphs. The second example applies this method to find the predictive models of a real distance education data set and checks the predictability of these models.

Keywords: Internet resource, user behavior, data sequencing, state-transition graph, hierarchical taxonomy

1. INTRODUCTION

1.1 Motivation

As local-area networks evolved into the Internet and then the WWW, electronized information, including text, data records, images, audio and video, came to be all connected into a hyperlinked structure. Users browse the embedded information from one resource to another, and the entire viewing process is recorded in a Web log. Different users browsing the web present different behavior models, the study of which is especially important for the investigation of learners' behaviors in distance education.

Received March 13, 2003; revised August 6, 2004; accepted November 1, 2004.
Communicated by Michael R. Lyu.

A standard Web log is in the form of [user, resource-access-action, time] records. Exploring knowledge in such a data store is called *Web mining*. Three data mining techniques, called *data clustering*, *data association*, and *data sequencing*, are typically used. Clustering techniques can be used to classify users and behaviors (actions) on the Web. Association provides quantitative measures between different users, between different behaviors (actions), or between users and behaviors. Finally, web browsing behaviors are sequential recorded, and then investigated using data sequencing techniques.

Users access Internet resources through DTE (Data Terminating Equipment), such as computers, telephones, PDAs, and so on. This use condition leads to a series of contiguous resource access actions, called *behaviors in a transaction*. On the other hand, all the access actions can be categorized into several classes and then into sub-classes. This phenomenon results in a level structure of access actions, called an *action hierarchy* or *taxonomy*.

With hierarchical codes of resource access actions, the behavior codes and behavior models must have some corresponding properties. By splitting a resource access log into transactions, we can describe the behavior transitions among transactions by using some models. In the following discussion, we will apply *radix coding* of *hierarchical taxonomy* and data mining techniques to propose an effective, complete algorithm, which we have tested using real data.

1.2 Related Works

The study of user behaviors in accessing Internet resources belongs to the research area of *Web mining*. The tasks of Web mining are divided into three categories: *Web content mining*, *Web structure mining*, and *Web usage mining*, and our problem falls within the last one [23]. With Web usage data, there are three processing steps: A) *preprocessing*, B) *pattern discovery*, and C) *pattern interpretation* [19], each of which corresponds to the five steps of knowledge discovery in databases, which are *selection*, *preprocessing* (corresponding to step A), *transformation*, *data mining* (corresponding to B), and *interpretation* (corresponding to C) [5].

Preprocessing in Web usage mining first divides the Web log into a series of sessions, which are sets of Internet access actions [23]. As for the use for distance education, a reasonable browsing time for access actions can be further determined from pedagogical psychology [13]. For pattern discovery, the traversal patterns (called behaviors in this paper) are always compressed in a suffix tree [14], k-sequences (large itemsets based on the Apriori algorithm) [2], or episodes [9, 11]. Several behavior models have also been studied: association rules, sequential patterns (large sequences based on the AprioriAll algorithm) [18], episodes, maximal forward references, and maximal frequent sequences [22]. All these behavior models not only study users in terms of their behaviors but also try to predict forthcoming user behaviors [20]. Besides these models, if the hierarchical property of taxonomy is put to full use [1], it is possible to greatly simplify the mining process and get hierarchical models, too. Furthermore, it is possible to use these behavior models to classify users.

As the temporal property of Internet behaviors is involved, the whole story is quite different. Although the investigation of *sequential patterns* [2] and their lattice structure [24] can greatly simplify the process, log mining seems to be more suitable for use with the finite-state machine [3] and hidden Markov model [17]. From traditional association, the inter-transaction rule [21] and episode rules [15] also play roles in behavior models.

It is important, for the purpose assessment, to examine the portfolio of learners, especially in the case of distance education. Distance learning includes three kinds of interaction: interaction between learners, interaction between teachers and learners, and interaction between learners and learning materials [15]. Corresponding to three styles of communication, these interactions consist of simplex, half-duplex, and full-duplex communication [7]. Such interactions can then be studied using the finite-state machine models of user behaviors [8, 16]. By continuing the mining patterns, such as large sequences [18], these behavior models provide more comprehensive means of predicting future behaviors.

1.3 Our Approach

This paper proposes a complete approach to finding the (state-transition graph) models of Internet behaviors based on the *radix coding* of *hierarchical taxonomy*. This state-transition graph provides a comprehensive model for the prediction of future behaviors in different user communities. The next section formulates Internet resources and the access actions of users, introduces the preprocessing steps of user behaviors, including sessionization and behavior filtering, and then describes *intra-transaction behaviors* and *inter-transaction behaviors*. The *radix coding of hierarchical taxonomy* is illustrated in section 3, including the radix codes of access actions, the aggregated codes of behaviors, and the distances between these radix codes. From these *hierarchical radix codes*, section 4 constructs *behavior hierarchy* and *nested state-transition graphs* for behavior transitions. In section 5, two examples are given to demonstrate the proposed algorithm: (1) the first example shows how to execute the AprioriAll algorithm to establish a set of *nested state-transition graphs*; (2) the second example applies this method to find the predictive models of a real data set for distance education. Finally, section 6 gives a conclusion.

2. THE PROBLEM OF DATA SEQUENCES WITH TRANSACTIONS

2.1 Data Sequences

Hypertexts, E-mails, and product items are all general Internet *resources* existing in a *resource set*. An atomic access taken action to access one resource is called a *resource access action*, symbolized as $\lambda \in \boldsymbol{\Lambda}$ (the *set of resource access actions*). A real human or a computer-simulated agent, who is able to access the above Internet resources in $\boldsymbol{\Lambda}\mathbf{0}$, is called a *user* or an *individual*, $\Psi_p \in \boldsymbol{\Psi}$. For example, $\boldsymbol{\Psi}_A = \{ \Psi_{A1}$ = A (Arthur), Ψ_{A2} = B (Buddy), Ψ_{A3} = C (Clement), Ψ_{A4} = D (Dian)$\}$.

An atomic access actions λ for user Ψ_p to access a resource at time t is defined as a *behavior element* $\boldsymbol{\beta}(p, \lambda, t) = [\Psi_p, \lambda, t]$. For example, [B, FG001, t] means user B (Buddy) gets text file 001 at time t. For a serial resource server, no matter whether it is an E-mail server or WWW server, all the behavior elements are often indexed by time t and recorded in a *log database of behavior sequences*:

$$\Theta = \{\boldsymbol{\beta}(p, \lambda, t) = [\Psi_p, \lambda, t] : \Psi_p \in \boldsymbol{\Psi}, \lambda \in \boldsymbol{\Lambda}, t \in T\}, \quad (1)$$

in which all the resource access actions λ's of Ψ_p's are mixed. For E-mail data [MailID, sender, receiver, *t*] recorded by the E-mail server, two transformed access actions can be obtained as [Ψ_p = sender, MSendMailID, *t*] and [Ψ_p = receiver, MGetMailID, *t*]. And the log file [IPaddress, T001, *t*] of the WWW server has to be combined with the login file [IPaddress, UserID] of the LOGIN server to obtain the form [Ψ_p = UserID, WT001, *t*].

For the database in Eq. (1), a series of behavior elements of a specific user Ψ_p forms a *user-behavior sequence* $\Theta(\Psi_p)$. For example, in the case of a WWW user, $\Theta(\Psi_{A1}$ = Arthur) = {[Arthur, WT101 = Login, t_1], [Arthur, BR001 = ReadNews, t_2], [Arthur, WM201 = Video, t_3], [Arthur, FP111 = PutHomework, t_4]} describes the behaviors of Arthur when he is visiting the given website. Such a behavior sequence can be used to find the *user model* $\Xi(\Theta(\Psi_p))$ in some formulation, such as a *large sequence* like Ξ(Arthur) = (FB001, SW111) [1], a *regular expression* like Ξ(Arthur) = (FB001)(SW111)* [23], and so on. No matter which user model is selected, it can be applied to predict the forthcoming user behavior.

With a classification method, the engaged user set can be divided into several *user clusters*, *user groups*, or *communities*, $\Omega_q \subseteq \boldsymbol{\Psi}$. All users in Ω_q possess the same or similar behaviors. For example, Ω_2 = {Buddy, Clement, Eoda} means that these three persons have similar user behaviors. Therefore, besides user behaviors, we are going to study *community behavior* $\Xi(\{\Theta(\Omega_q)\})$.

2.2 Preprocessing Steps: Sessionize, Cleaning, Focused Actions

For an Internet service, the server cannot differentiate whether the client user keeps working on his/her DTE (Data Terminating Equipment) or not, but can only track the Internet requests from the client side. Therefore, information about the access durations of a user logging on to a DTE can be traced only with the time duration between two consecutive access actions, so the following assumption is needed.

Assumption 1 (Assumption for Access Duration) The user uses (stays at) the resource right from the time he/she gets (accesses) it until he/she fetches (accesses) the next resource.

With this assumption, the *access duration* $D(p, \lambda)$ is defined as the time interval needed for a user to access a resource. However, such a time duration can be extraordinarily unreasonable, say, $D(p, \lambda)$ = 102108 seconds. This is because these two access actions do not occur at the same time, but fall into two different use times. This phenomenon leads to the concept of transactions. A *transaction* is defined as a series of resource actions for one user performed to contiguously use the client program. This definition needs an assumption.

Assumption 2 (Assumption for Transaction) When the access duration is greater than a threshold, called *TransactionBoundary*, the next resource is treated as being used at another (the next) transaction.

Note that this assumption makes the access time of the last resource access action indeterminable. The order of transactions is defined as an integer, called the *transaction index* ($TIND(\Psi_p) = m(\Psi_p) = m(p)$), whereas the order of access actions in a transaction is also defined as an *action index* $AIND(\Psi_p, TIND(\Psi_p)) = n(m(p))$. For example, the first action when a user uses the client program is indexed as $TIND(\Psi_p) = m(\Psi_p) = 1$ and $AIND(\Psi_p, 1) = n(m(p)) = 1$. Then, the *access duration* can be redefined formally: $D(p, \lambda, m, n) = t' - t$, where $\beta(p, \lambda, t)$ and $\beta_{next}(p, \lambda', t')$ are two contiguous access actions, $m = m(p) = TIND(\Psi_p)$, and $n = n(m(p)) = AIND(\Psi_p, m)$. All these calculations are collected in the following algorithm.

Algorithm 1 (Algorithm for Determining Transaction Indices, Action Indices and Access Durations) Given a series of behavior elements in a user-resource log database as shown in Eq. (1) with all the records of the same user collected in the increasing order of time, find the access durations $D(p, \lambda, m, n)$, transaction indices $TIND(\Psi_p)$, and action indices $AIND(\Psi_p, m')$.

1. $m(p) = 1$ for each Ψ_p.
2. For two successive behavior elements $\beta(p, \lambda, t) = [\Psi_p, \lambda, t]$ and $\beta(p', \lambda', t') = [\Psi_{p'}, \lambda', t']$, carry out the following checks.
3. If $\Psi_{p'} = \Psi_p$ and $t' - t \leq TransactionBoundary$, then
   $D(p, \lambda, m, n) = t' - t$, $TIND(\Psi_p) = m(p)' = m(p)$, and
   $AIND(\Psi_p, m') = n(m(p))' = n(m(p)) + 1$.
4. If ($\Psi_{p'} = \Psi_p$ and $t' - t > TransactionBoundary$) or ($\Psi_{p'} \neq \Psi_p$),
   then $D(p, \lambda, m, n)$ = indeterminable and $TIND(\Psi_p) = m(p') = m(p) + 1$.

With this algorithm, the access time, transaction index, and action index of Eq. (1) can be found; then, the *augmented behavior sequence*

$$\Theta_a(\Psi_p) = \{[\Psi_p, \lambda, D(p, \lambda, m, n), m(p), n(m(p))]\} \tag{2}$$

can be obtained. Before actions are transformed into behaviors, the above behaviors have to be selected carefully. The selected behaviors are called *focused actions* or, briefly, *focuses*. The selection of focuses is based on four *selection criteria*:

1. **Time windowing:** to select behaviors for $t \in Twindow = [t_i, t_f]$, which are always selected for studying those behaviors with the time from the initial time t_i to the final time t_f; e.g., behaviors for $t \in [0, 1500000(\text{sec})]$ represent those behaviors in the first two weeks.
2. **Duration thresholding:** to select behaviors for $D(p, \lambda, m, n) \geq DurationThreshold$, which are always selected for those behaviors with longer stays; e.g., behaviors for $D(p, \lambda, m, n) \geq 25(\text{sec})$ represent those behaviors with stay times longer than or equal to 25 seconds.
3. **RAcode centering:** to select behaviors for $\lambda \in RAcenter = \{\lambda_1, \ldots, \lambda_i, \ldots\}$, which are always selected for those behaviors with designated codes (to do some specific kinds of actions or access some specific kinds of resources); e.g., actions for $a_1a_2(\lambda) \in RA$-

center = {WM, BP} represent the specific actions of a WWW server and BBS server.

4. **RAcode leveling:** to select behaviors for $\lambda_1 \ldots \lambda_{level}(\lambda)$, which are always selected for those behaviors with designated *code levels* (neglecting deeper codes, which are usually resource codes); e.g., $\lambda_{level=1}(\lambda)$ causes action codes in examples to become {W, B}, which then can be used to study the behaviors from the server (Web, BBS) viewpoint.

All the four selection criteria can be formulated together as a *focus filter*:

$$\Theta_f(\Psi_p) = \{\boldsymbol{\beta_f}(p, \lambda, D, m, n)\} = FocusFilter(\Theta_a(\Psi_p), Twindow, DurationThreshold, RAcenter, level), \quad (3)$$

which concentrates the focus of augmented behaviors $\Theta_a(\cdot)$ upon the focused behaviors $\Theta_f(\cdot)$, with a set of the above four selection criteria.

Example 1. Let *TransactionBoundary* = 7200 (secs = 2 hours). Algorithm 1 gives the access durations $\boldsymbol{D}$, transaction indices $\boldsymbol{m}$, and action indices $\boldsymbol{n}$, and the filtering conditions are $t \in Twindow = [0, 1500000]$, $Aduration \geq 25$, $c_1c_2c_3(RAcode) \in \{WT, WM, BP, BR\}$, and $level = 3$.

Table 1. Example of transaction and duration determination.

Ψ_p	λ	t	$D(p, \lambda, m, n)$	$m(p)$	$n(m(p))$	β_f
A	WT101	12	43	1	1	WT
	WT121	55	10		2	–
	WT202	65	indeterminable		3	WM
	WM210	10215	1297	2	1	WM
	WM201	11412	indeterminable		2	WM
B	WM212	980	30	1	1	WM
	WM221	1010	4002		2	WM
	BP710	5012	2209		3	BP
	BR720	7221	indeterminable		4	BR
C	WT111	1200	1050	1	1	WT
	WM212	2250	24		2	–
	WT111	2274	indeterminable		3	WT
	BP710	18221	indeterminable	2	1	BP
D	WM211	512	171	1	1	WM
	WM212	783	indeterminable		2	WM
	BP710	19421	1014	2	1	BP
	BR720	20435	indeterminable		2	BR

2.3 Intra-Transaction/Inter-Transaction Behaviors

When the behaviors in a log database are divided into transactions, coded, and selected as focuses, the log database in Eq. (1) becomes a series of focus symbols as in Eq. (3). Through the collection of these behavior according to their (user, transaction) indices, the above-defined user behavior $\Theta_f(\boldsymbol{\Psi}, \boldsymbol{\Lambda}, T)$ can be re-formulated as a set of *transaction-indexed user behaviors*:

$$\Theta_f(\Psi_p, m) \equiv \{[n, \lambda, t] : \lambda \in \boldsymbol{\Lambda}, \forall n(m(\Psi_p)), t \in T\}. \quad (4)$$

Take Example 1 as an example. The corresponding transaction-indexed user behaviors are

$\Theta_f(A, 1)$ = [(WT, 12), (WM, 65)],
$\Theta_f(A, 2)$ = [(WM, 10215), (WM, 11412)],
$\Theta_f(B, 1)$ = [(WM, 980), (WM, 1010), (BP, 5012), (BR, 7221)],
$\Theta_f(C, 1)$ = [(WT, 1200), (WT, 2274)],
$\Theta_f(C, 2)$ = [(BP, 18221)],
$\Theta_f(D, 1)$ = [(WM, 512), (WM, 783)] and
$\Theta_f(D, 2)$ = [(BP, 19421), (BR, 20435)].

Note that the indices, that is, the n's, are neglected because they are replaced by the positions of elements $[\lambda, n]$ in vectors $\Theta_f(\Psi_p, m)$'s. It can be seen that the time scale between transactions (more than 7200 seconds, usually more than several days) is much greater than that within actions in a transaction (usually within hundreds of seconds). This brings us to the following assumption.

Assumption 3 (Assumption for Action Indices) Given the transaction-indexed user behavior in Eq. (4), as the time durations between transactions grow longer (always 10 times longer the time durations between actions in the same transaction), the action indices n's can be neglected.

As this assumption is satisfied, the above two kinds of behaviors gain new definitions: *intra-transaction user-behavior* $\Theta_{intra} = \{\lambda^n\} \equiv \Theta$ and *inter-transaction user-behavior*, *that is*,

$$\Theta_{inter}(\Psi_p) = \{\boldsymbol{\beta}(p, m) = \{\lambda^n\}\} \equiv \Theta(\Psi_p), \quad (5)$$

where $\lambda^n = [\lambda, n]$. The former behaviors describe the switching of actions within a transaction, whereas the latter represent the changes of (sets of) actions from one transaction to the next. Note that these two behaviors are automatically differentiated by the presence of argument Ψ_p. Continuing with the above example, we have the following user-behaviors:

Θ = {[WT], [WT], [WM], [WM], [WM], [WM], [BP], [BR], [WT], [WT], [BP], [WM], [WM], [BP], [BR]} = {WT, WM, BP, BR},

and

$\Theta(A) = [\boldsymbol{\beta}(A, 1) = \{WT\}, \boldsymbol{\beta}(A, 2) = \{WM\}]$,
$\Theta(B) = [\boldsymbol{\beta}(B, 1) = \{WM, BP, BR\}]$,
$\Theta(C) = [\boldsymbol{\beta}(C, 1) = \{WT\}, \boldsymbol{\beta}(C, 2) = \{BP\}]$, and
$\Theta(D) = [\boldsymbol{\beta}(D, 1) = \{WM\}, \boldsymbol{\beta}(D, 2) = \{BP, BR\}]$.

Each element of $\Theta(\cdot)$ represents an un-ordered set of resource access actions, i.e., an inter-transaction behavior $\boldsymbol{\beta}(p, m)$, which will be used in data mining. Before data mining techniques are used in the modeling of these behaviors, radix-coding of hierarchical taxonomy can be employed to make these data much simpler.

3. RADIX CODING OF HIERARCHICAL TAXONOMY

3.1 Hierarchical Taxonomy

As in the item taxonomy of Agrawal and Srikant [1], these accessed resources will generally form a *hierarchical taxonomy*. Fig. 1 shows an example. As Fig. 1 shows, the above access actions in Eq. (5) can be represented in a vector form: $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_l, \ldots, \lambda_L]$, where the integer $\lambda_l \in \{1, 2, \ldots, L\}$ indicates that the access action $\boldsymbol{\lambda}$ belongs to λ_l^{th} branch at the l^{th} level of the taxonomy and L is the depth of the taxonomy. Each access action corresponds to a terminal node in this taxonomy. Note that such vector formulation can be applied to terminal nodes as well as internal nodes. Furthermore, a general form of a *partial action* can be denoted as $\boldsymbol{\lambda}_{i:j} = [\lambda_i, \ldots, \lambda_j]$, where $1 \le i < j \le L$. Then, as shown in Fig. 1, the post actions of a BBS message 002 can be generally represented as BP002 (which means [B, P, 001] = [2, 2, 002]).

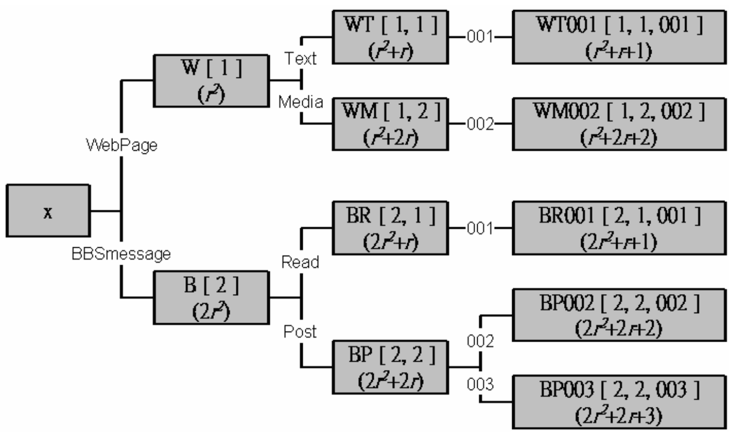

Fig. 1. Example showing the coding of resource access actions.

3.2 Radix Codes of Access Actions

To simplify the action expression and the following derivation, the nodes in the hierarchical taxonomy are coded in a radix representation:

$$a(\boldsymbol{\lambda}) = code(\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_l, \ldots, \lambda_L]) = \sum_{l=1,\ldots,L} \lambda_l r^{L-l} = \boldsymbol{\lambda} \cdot \boldsymbol{r}, \tag{6}$$

where r is the *radix* and $\boldsymbol{r} = [r^{L-1}, r^{L-2}, \ldots, r, 1]$ is its *radix vector*. In Eq. (6), r is an integer greater than one, indicating that those codes in lower levels gain more weight. Similarly, $a_{i:j} = a(\boldsymbol{\lambda}_{i:j})$ is the radix code of a partial action. For example, BP(post) and BR(read) are two action codes for BBS messages, and if 001, 002, and 003 are three message (resource) codes for messages, then $code(\text{BP00}x = [2, 2, x]) = [2, 2, x] \cdot [r^2, r, 1] = 2r^2 + 2r + x$ and $code(\text{BR001} = [2, 1, 1]) = 2r^2 + r + 1$. Also the internal node BP can use $code(\text{BP} = [2, 2]) = 2r^2 + 2r$ to represent all the post actions.

This radix coding of access actions is performed under the constraint that all the access actions with higher codes (λ_l's) will be larger than those with lower codes (λ_{l+1}'s). Satisfaction of this constraint depends on careful selection of the value of r, as the following lemma indicates.

Lemma 1 **(A)** Let the radix in Eq. (6) be selected as $r > \max\{\lambda_l\} + 1$; then, $a_k^i \in (r^{L-k}, r^{L-1} - r^{L-k})$. **(B)** As two access actions $a^i = code(\boldsymbol{\lambda}^i)$ and $a^i = code(\boldsymbol{\lambda}^j)$ with $a_l^i = a_l^j$ for $l = 1, \ldots, k-1$ and $a_l^i > a_l^j$, the resultant combined code of $a_{1:k+1}^i > a_{1:k+1}^j$.

Proof: **(A)** Eq. (6) causes $a_k^i = \sum_{l=1,\ldots,k} \lambda_l r^{L-l} \le \sum_{l=1,\ldots,k} (r-1) r^{L-l} = (r-1)r^{L-k}(r^{k+1} - 1)/(r-1) = r^{L-1} - r^{L-k}$ and $a_k^i = \sum_{l=1,\ldots,k} \lambda_l r^{L-l} \ge r^{L-k}$. **(B)** By definition, the l level of access action $\lambda_{1:l}$ is

$$a_{1:k+1}^i = \sum_{l=1,\ldots,k+1} \lambda_l \cdot r^{L-l} = \sum_{l=1,\ldots,k} \lambda_l \cdot r^{L-l} + \lambda_{k+1} \cdot r^{L-k-1} = a_{1:k}^i + \lambda_{k+1} \cdot r^{L-k-1}. \tag{7}$$

Then the conclusion of this lemma can be drawn obviously. ❑

Note that Eq. (7) in the above proof is called the *nesting formula of an access action*. In general, $r > \max\{\lambda_l\} + 1$ for all levels. For the above example, as $r > \max\{1, 2, 001, 002, 003\}$, the radix can be chosen as $r = 5$. Then, the above coding becomes $code(\text{BR001} = [2, 1, 1]) = 2 \cdot 5^2 + 5 + 1 = 56$ and $code(\text{BP002} = [2, 2, 2]) = 2 \cdot 5^2 + 2 \cdot 5 + 2 = 62$.

The hierarchical tree in Fig. 1 is an example of a *code hierarchy*. The formal description of a code hierarchy can be written as $\boldsymbol{CH}(\Lambda = \{\boldsymbol{\lambda}_{1:l}\}, E_\Lambda)$, where $\Lambda = \{\boldsymbol{\lambda}_{1:l}\}$ is the set of all $1 - l^{\text{th}}$ codes and E_Λ is the edge set of all edges between $\boldsymbol{\lambda}_{1:l}$ and $\boldsymbol{\lambda}_{1:l+1}$. Such a hierarchy can be quantified as a *radix code hierarchy*: $\boldsymbol{RCH}(\{a(\boldsymbol{\lambda}_{1:l})\}, RE_\Lambda)$, where RE_Λ is the edge set of all edges between $a(\boldsymbol{\lambda}_{1:l})$ and $a(\boldsymbol{\lambda}_{1:l+1})$. Note that Eq. (7) in Lemma 1 indicates that $a(\boldsymbol{\lambda}_{1:l+1}) = a(\boldsymbol{\lambda}_{1:l}) + \lambda_l \cdot r^l$, then $a(\boldsymbol{\lambda}_{1:l+1}) > a(\boldsymbol{\lambda}_{1:l})$.

3.3 Aggregated Radix Codes of Behaviors

With this kind of coding, the scattering of behavior codes causes behaviors with similar codes to cluster together. Through elaborate design of the weights and radices, such coding can be extended to inter-transaction behaviors. Since an inter-transaction behavior $\boldsymbol{\beta}(p, m)$ is the union of its elements (access actions, $\boldsymbol{\lambda}$'s), its coding $b(p, m)$ can be naturally performed by adding up or maximizing their corresponding codes, a technique called *code summation* or *code maximization*.

When the *code summation* strategy is adopted, all the codes in intra-transaction behaviors, $\alpha(\boldsymbol{\lambda}^n)$'s, can be replaced and the inter-transaction behavior code then becomes

$$b_{SUM}(p, m) = code(\boldsymbol{\beta}(p, m) = \{\boldsymbol{\lambda}^n\}) = \sum_n a^n = \sum_{l=1,L} (\sum_n \lambda_l^n) r^{L-l}. \quad (8)$$

When $code$(WT001 = [1, 1, 1]) = $r^2 + r + 1$ and $code$(WM002 = [1, 2, 2]) = $r^2 + 2r + 2$, we have $code$({WT001, WM002}) = $2r^2 + 3r + 3$, which may generate ambiguous $code$(BP00x = [2, 2, x]) = $2r^2 + 2r + x$.

To avoid the above radix-like coding of access code $\boldsymbol{\lambda}$, inter-transaction behaviors can use the *code maximization* strategy to aggregate their access actions:

$$b_{MAX}(p, m) = code(\boldsymbol{\beta}(p, m) = \{\boldsymbol{\lambda}^n\}) = \max_n\{a^n\} = \sum_{l=1,L} (\max_n \{\lambda_l^n\}) r^{L-l}. \quad (9)$$

When $code$(WT001) = $r^2 + r + 1$ and $code$(WM002) = $r^2 + 2r + 2$, we have $code$({WT001, WM002}) = $r^2 + 2r + 2$, which can be easily differentiated from $code$(BP00x) = $2r^2 + 2r + x$. Note that this coding strategy pays no attention to the enumeration of the same behavior in one transaction.

Besides the code summation strategy Eq. (8) and code maximization strategy Eq. (9), an inter-transaction behavior can employ more code aggregation strategies to assemble its intra-transaction action codes. In general, all aggregation operations can be applied [10].

3.4 Distances between Radix Codes

A set of inter-transaction behaviors, $\Theta(\Psi_p)$, in Eq. (5) is a collection of access actions; its corresponding distance is, naturally, the difference between them. Radix code provides these behaviors corresponding to aggregation codes. Accordingly, the numeric difference between two radix action codes can be defined as

$$d(a^i, a^j) = \sum_l \lambda_l^j r^l - \sum_l \lambda_l^i r^l = \sum_{l=1,L} (\lambda_l^j - \lambda_l^i) \cdot r^{L-l} = (\boldsymbol{\lambda}^j - \boldsymbol{\lambda}^i) \cdot \boldsymbol{r}. \quad (10)$$

Then, the distance between two radix codes of inter-transaction behaviors has the following relationships:

$$\begin{aligned} d(b_{SUM}^i, b_{SUM}^j) &= code(\{\boldsymbol{\lambda}^j - \boldsymbol{\lambda}^i\}) = \sum (\boldsymbol{\lambda}^j - \boldsymbol{\lambda}^i) \cdot \boldsymbol{r} = \sum \boldsymbol{\lambda}^j \cdot \boldsymbol{r} - \sum \boldsymbol{\lambda}^i \cdot \boldsymbol{r} \\ &= b_{SUM}^j - b_{SUM}^i, \end{aligned} \quad (11)$$

and

$$\begin{aligned} d(b^i_{\text{MAX}},\ b^j_{\text{MAX}}) &= code(\{\boldsymbol{\lambda}^j - \boldsymbol{\lambda}^i\}) = \textstyle\sum_{l=1,L}(\max\{\lambda^j_l - \lambda^i_l\})r^l \\ &= \textstyle\sum_{l=1,L}(\max\{\lambda^j_l\})r^l - \sum_{l=1,L}(\max\{\lambda^i_l\})r^l \\ &= \boldsymbol{\lambda}^j \cdot \boldsymbol{r} - \boldsymbol{\lambda}^i \cdot \boldsymbol{r} = b^j_{\text{MAX}} - b^i_{\text{MAX}}. \end{aligned} \tag{12}$$

For the inter-transaction behaviors in Example 2, the code maximization strategy causes the behavior codes to be $b(B, 1) = code(\boldsymbol{\beta}(B, 1) = \{\text{WM002, BP002, BR001}\}) = \max\{r^2 + 2r + 2,\ 2r^2 + 2r + 2,\ 2r^2 + r + 1\} = 2r^2 + 2r + 2$, $b(C, 1) = code(\boldsymbol{\beta}(C, 1) = \{\text{WT001}\}) = r^2 + r + 1$, and $b(D, 2) = code(\boldsymbol{\beta}(D, 2) = \{\text{BP002, BR001}\}) = \max\{2r^2 + 2r + 2,\ 2r^2 + r + 1\} = 2r^2 + 2r + 2$. In addition, we can get $d(b(B, 1), b(C, 1)) = r^2 + r + 1$ and $d(b(B, 1), b(D, 2)) = 0$. The overall possibilities of these behavior codes in Eq. (9) and their in-between distances in Eq. (12) are listed in Table 2. Table 3 shows the codes in Eq. (8) and distances in Eq. (11) obtained using the code summation strategy, where the radix r is always chosen to be greater than the weights w and x, e.g., $r = 10$ and $w = 5$, as shown in Table 4. Note that all the additions and subtractions are performed in a radix-wise manner.

From the nesting formula of an access action Eq. (7), the distance between two inter-transaction behavior-codes possesses the form shown in Eq. (13) and described in Lemma 2. The *nesting formula of distance* in Eq. (13) plays a key role in the derivation of our cluster in the next section.

Lemma 2 Given two inter-transaction behavior-codes $b^1(p, m) = code(\boldsymbol{\beta}^1(p, m) = \{\lambda^1_n\})$ or $b^2(p, m) = code(\boldsymbol{\beta}^2(p, m) = \{\lambda^2_n\})$, their distance has the following nesting property:

$$d(b^1_{1:k+1},\ b^2_{1:k+1}) = d(b^1_{1:k},\ b^2_{1:k}) + r^{L\text{-}k\text{-}1} \cdot d(b^1_{k+1},\ b^2_{k+1}). \tag{13}$$

Proof: From Eq. (7), $code(\lambda^i_{1:k+1}) = a^i_{1:k+1} = \sum_{l=1,k+1} a^i_l \cdot r^{L\text{-}l} = \sum_{l=1,k} a^i_l \cdot r^{L\text{-}l} + a^i_{k+1} \cdot r^{L\text{-}k\text{-}1}$. Then, the distance between two action codes becomes

$$\begin{aligned} d(a^1_{1:k+1},\ a^2_{1:k+1}) &= \textstyle\sum_{l=1:k+1}(a^2_l - a^1_l) \cdot r^{L\text{-}l} \\ &= \textstyle\sum_{l=1,k}(a^2_l - a^1_l) \cdot r^{L\text{-}l} + r^{L\text{-}k\text{-}1} \cdot (a^2_{k+1} - a^1_{k+1}) \\ &= d(a^1_{1:L},\ a^2_{1:L}) + r^{L\text{-}k\text{-}1} \cdot d(a^1_{k+1},\ a^2_{k+1}). \end{aligned}$$

When inter-transaction behaviors use the code summation strategy shown in Eq. (8), the distance between two behaviors is

$$\begin{aligned} d(b^1_{1:k+1},\ b^2_{1:k+1}) &= d(\textstyle\sum a^1_{1:k+1},\ \sum a^2_{1:k+1}) = \sum d(a^1_{1:k+1},\ a^2_{1:k+1}) \\ &= d(\textstyle\sum a^1_{1:k},\ \sum a^2_{1:k}) + r^{L\text{-}k\text{-}1} \cdot d(\sum a^1_{k+1},\ \sum a^2_{k+1}) \\ &= d(b^1_{1:k},\ b^2_{1:k}) + r^{L\text{-}k\text{-}1} \cdot d(b^1_{k+1},\ b^2_{k+1}). \end{aligned}$$

Similarly, Eq. (13) can be easily proven for code maximization strategy. ❑

Table 2. Example of inter-transaction behavior codes and their distances obtained with the code maximization strategy.

distance	*behavior*	{WT001}	{WM002}	{WM002, BP002, BR001}	{BR001}	{BP002, BR001}
behavior	*code*	$r^2 + r + 1$	$r^2 + 2r + 2$	$2r^2 + 2r + 2$	$2r^2 + r + 1$	$2r^2 + 2r + 2$
{WT001}	$r^2 + r + 1$	0	$r + 1$	$r^2 + r + 1$	r^2	$r^2 + r + 1$
{WM002}	$r^2 + 2r + 2$	$r + 1$	0	r^2	$r^2 - r - 1$	r^2
{WM002, BP002, BR001}	$2r^2 + 2r + 2$	$r^2 + r + 1$	r^2	0	$r + 1$	0
{BR001}	$2r^2 + r + 1$	r^2	$r^2 - r - 1$	$r + 1$	0	$r + 1$
{BP002, BR001}	$2r^2 + 2r + 2$	$r^2 + r + 1$	r^2	0	$r + 1$	0

Table 3. Example of inter-transaction behavior codes and their distances obtained with the code summation strategy.

distance	*behavior*	{WT001}	{WM002}	{WM002, BP002, BR001}	{BR001}	{BP002, BR001}
behavior	*code*	$r^2 + r + 1$	$r^2 + 2r + 2$	$(2w + 1)r^2 + 5r + 5$	$wr^2 + r + 1$	$2wr^2 + 3r + 3$
{WT001}	$r^2 + r + 1$	0	$r + 1$	$2wr^2 + 4r + 4$	$(w - 1)r^2$	$(2w - 1)r^2 + 2r + 2$
{WM002}	$r^2 + 2r + 2$	$r + 1$	0	$2wr^2 + 3r + 3$	$(w - 1)r - r - 1$	$(2w - 1)r^2 + r + 1$
{WM002, BP002, BR001}	$(2w + 1)r^2 + 5r + 5$	$2wr^2 + 4r + 4$	$2wr^2 + 3r + 3$	0	$(w + 1)r + 4r + 4$	$r^2 + 2r + 2$
{BR001}	$wr^2 + r + 1$	$(w - 1)r^2$	$(w - 1)r - r - 1$	$(w + 1)r + 4r + 4$	0	$wr^2 + 2r + 2$
{BP002, BR001}	$2wr^2 + 3r + 3$	$(2w - 1)r^2 + 2r + 2$	$(2w - 1)r^2 + r + 1$	$r^2 + 2r + 2$	$wr^2 + 2r + 2$	0

Table 4. Example of inter-transaction behavior codes and their distances obtained with the code summation strategy and $r = 10$, $w = 5$.

distance	*behavior*	{WT001}	{WM002}	{WM002, BP002, BR001}	{BR001}	{BP002, BR001}
behavior	*code*	111	122	1155	511	1033
{WT001}	111	0	11	1044	400	922
{WM002}	122	11	0	1033	389	911
{WM002, BP002, BR001}	1155	1044	1033	0	644	122
{BR001}	511	400	389	644	0	522
{BP002, BR001}	1033	922	911	122	522	0

4. MODEL OF DATA SEQUENCING

4.1 Subsequence and STG Models

In the formulation of Eq. (5), inter-transaction behavior $\Theta(\Psi_p)$ possesses a standard format of a data sequence in the research area of data sequencing [4], where an access action is called an *item*, a series of items in a transaction is called an *itemset*, and the aim of these researches is to find the *large sequence* of itemsets. Apparently, these kinds of behaviors can be represented in a more explicit form: $\Theta(\Psi_p)$ = "$\boldsymbol{\beta}(p, 1) \rightarrow \boldsymbol{\beta}(p, 2) \rightarrow \ldots \rightarrow \boldsymbol{\beta}(p, m) \rightarrow \ldots$," which is a sequence of behaviors $\boldsymbol{\beta}(p, m)$'s. For example, $\Theta(A)$ = "{WT} → {WM}," $\Theta(B)$ = "{WM, BP, BR}," $\Theta(C)$ = "{WT} → {BP}," and $\Theta(D)$ = "{WM} → {BP, BR}."

There are many data mining methods for modeling these data sequences, such as the above-mentioned subsequence [1], sequence lattice [24] and regular expression [6] methods, Amongst these methods, the most popular formulation is the *subsequence technique*, which can be simplified into two stages for our problem:

1. Find large itemsets with frequencies equal to or more than a designated *support count* and then to endow them with other codes;
2. Search large sequences of 1 large itemset with enough support, then those of 2 large itemsets with enough support, and so on.

The first stage for large itemset and their codes corresponds to inter-transaction behaviors with aggregated access actions and radix codes that can simplify the search work for hierarchical taxonomy. Let $count(b_l^i = code(\boldsymbol{\beta}^i)_l)$ be the *support count* of inter-transaction behaviors $\boldsymbol{\beta}^i$ with level l in a data sequence. The second stage for large sequences with more than one large itemset can then be performed by searching these radix codes.

As mentioned in many researches [1], the supports for k-element large sequences decrease with the value of k. Then, the most effective large sequences are 1-element and 2-element ones. The count of a 1-element large sequence shows the frequency of this element in data sequences. Furthermore, 2-element large sequences indicate how many consecutive behaviors will occur. From the quantitative viewpoint, $count\,([b_l^{i2} = code(\boldsymbol{\beta}^{i2})_l,\ b_l^{i2} = code(\boldsymbol{\beta}^{i2})_l])$ is the *support count* for a 2-element behavior sequence $[\boldsymbol{\beta}^{i1}, \boldsymbol{\beta}^{i2}]$. For the behaviors in Example 1, the support counts can be *count*([{WT}, {WM}]) = 1, *count*([{WT}, {BP}]) = 1, and *count*([{WM}, {BP, BR}]) = 1.

Furthermore, the probability of a large sequence indicates the portion for this element to occupy in data sequences. Let *Ndata* be the total number of all behaviors (transactions) in a data sequence. For the probability density functions of behavior (itemset) sequences, we have *P*({WT}) = *count*({WT})/*Ndata* = 2/7 and *P*([{WT}, {WM}]) = *count*([{WT}, {WM}])/*Ndata* = 1/7. These probability density functions of 2-element item sequences can be re-formulated according to the Bayesian rule:

$$P(\boldsymbol{\beta}^{i2} \mid \boldsymbol{\beta}^{i1}) = P([\boldsymbol{\beta}^{i1}, \boldsymbol{\beta}^{i2}])/P(\boldsymbol{\beta}^{i1}) = count(b^{i1}, b^{i2})/count(b^{i1}). \qquad (14)$$

For example, *P*([{WM} | {WT}]) = 1/2 = *P*([{WT}, {WM}])/*P*([{WT}]) = (1/7)/(2/7).

Here, the conditional probability $P(\boldsymbol{\beta}^{i2} \mid \boldsymbol{\beta}^{i1})$ can be interpreted as the *transition probability* that a user with the behavior $\boldsymbol{\beta}^{i1}$ will present the behavior $\boldsymbol{\beta}^{i2}$ in succession. Of course, such a transition only holds under the Markovian assumption.

Assumption 4 (Assumption for the 1st order Markovian of Inter-transaction Behaviors) Given the intra-transaction behaviors $\Theta(\Psi_p) = \{\boldsymbol{\beta}(p, m), m = 1, 2, \ldots\}$ of a set of users Ψ_p's, where m is the transaction index, the inter-transaction behaviors satisfy the Markov property; i.e., his/her m^{th} behavior $\boldsymbol{\beta}(p, m)$ is dependent on the previous behavior, $\boldsymbol{\beta}(p, m-1)$.

As the intra-transaction behaviors in Eq. (5) are defined based on a finite number of states, the transition probability Eq. (14) will describe the switching between these states quantitatively. Therefore, the results of data sequences can be used to define *the state-transition graphs of inter-transaction behaviors*, as the following definition indicates.

Definition 1 (State-Transition Graph of Inter-Transaction Behaviors) For a user Ψ_p or a user cluster Ψ_p, $p = 1, \ldots, N_{user}$, *the state-transition graph of inter-transaction behaviors* is defined as **STG**(Θ, $\boldsymbol{P}$), where $\Theta = \{\boldsymbol{\beta}(p, m), m = 1, 2, \ldots, p = 1, \ldots, N_{user}\}$ is the set of intra-transaction behaviors and $\boldsymbol{P} = [P([\boldsymbol{\beta}^{i2} \mid \boldsymbol{\beta}^{i1}]), \boldsymbol{\beta}^{i1}, \boldsymbol{\beta}^{i2} \in \Theta] = [count(\boldsymbol{\beta}^{i1}, \boldsymbol{\beta}^{i2})]$ ./ $[count(\boldsymbol{\beta}^{i1})] = [count(b^{i1}, b^{i2})]$ ./ $[count(b^{i1})]$ is the *transition probability matrix*, where ./ stands for element-wise division.

For Example 1, Θ = {{WT}, {WM}, {WM, BP, BR}, {BP}, {BP, BR}} and the transition probability matrices of all users are

$$\boldsymbol{P} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} ./ \begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{15}$$

Such a stochastic finite-state machine gives a preliminary prediction of forthcoming states; for example, the present state of {WT} makes the next states {WM} and {BP}, each of which has a probability of 1/2. However, the items (access actions, behaviors) of our problems have item (access action) taxonomy with a hierarchical structure as shown in Fig. 1. Combined with the proposed behavior automata, the hierarchical taxonomy proposes a behavior hierarchy.

4.2 Radix Codes for a Behavior Hierarchy

No matter whether it is K-mean or K-NN, a clustering algorithm divides the given data set $X = \{x_i\}$ into several clusters $X^n = cluster_n(X, D)$, $n = 1, 2, \ldots, N$, where N is the number of clusters and D is the cutoff distance for different clusters. Each cluster usually has an (explicit or implicit) *cluster center*, $center(X^{\boldsymbol{n}}) \equiv c_n$. As clustering algorithms are generally described, the granularity of the resultant clusters is related to the number of clusters, the distances between cluster centers, or the accumulated errors in clusters [12]. With the definition of distance $d(\cdot, \cdot)$ from radix code, the clustering of behavior elements is much simplified, and its time complexity is then greatly reduced. The center of a cluster and its data members can be formally defined:

$$d(c_n, x_i) \leq D \leq d(c_k, x_i),\ k = 1, 2, \ldots, N,\ k \neq n,\ x_i \in X^{\boldsymbol{n}}, \tag{16}$$

which means that cluster center c_n has the minimal distance (but greater than D) to its data members, e.g., $c_n = arg\min_{ck} d(c_k, x_i)$. Here, the distance, no matter whether it is Manhattan or Hamming, is negligible.

For code hierarchy $\boldsymbol{RCH}(\{a(\boldsymbol{\lambda}_{1:k})\}, RE_{\Lambda})$, Eq. (7) leads to for all k,

$$d(a_{k+1}^{i},\ a_{k+1}^{i}) \geq r^{L\text{-}k+1} > d(a_{k}^{i},\ a_{k}^{i}), \tag{17}$$

which creates a nested clustering of access actions $\{a(\boldsymbol{\lambda})\}$. As the cutoff distance is chosen as $D \in (r^{L\text{-}k+1}, r^{L\text{-}k}]$, the clustering results match the original hierarchy $\boldsymbol{RCH}(\cdot, \cdot)$. For the example shown in Fig. 1, $d(a(\text{BP002})_1, a(\text{BR001})_1 = (2r^2) - (2r^2) = 0$, $d(a(\text{BP002})_2, a(\text{BR001})_2 = (2r^2 + 2r) - (2r^2 + r) = r$, and $d(a(\text{BP002})_3, a(\text{BR001})_3) = (2r^2 + 2r + 2) - (2r^2 + r + 1) = r + 1$. This calculation proves Eq. (10). For the entire codes shown in Fig. 1, when $D \in [r^2, r^3)$, there are two clusters (W, B); when $D \in [r, r^2)$, there are four clusters (WT, WM, BR, BP); when $D \in [1, r)$, there are five clusters (WT001, WM002, BR001, BP002, BP003).

This technique can be extended to cluster inter-transaction behaviors, $\boldsymbol{\beta}$'s. With the above-mentioned distance measure selected, these inter-transaction user behaviors can be classified into user-behavior classes: $\Theta^{q^*} = \cup_q \Theta^q = cluster(\{b(p, m) = code(\boldsymbol{\beta}(p, m)), \boldsymbol{\beta}(p, m) \in \Theta\}, D)$, $q = 1, 2, \ldots, N_q$. Note that in such clustering, the user index p and transaction index m can both be ignored since all users and transaction data are classified together. With the distances listed in Table 3, a cutoff distance $D = 320 \geq (w - 2)r^2$ will give three behavior clusters: Θ^{q1} = {{WT001}, {WM002}}, Θ^{q2} = {{BR001}}, and Θ^{q3} = {{WM002, BP002, BR001}, {BP002, BP003}}. Apparently, when the cut-off distance is relaxed to $D' = 110 \in (r, r^2]$, the number of clusters increases: $\Theta^{q'1}$ = {{WT001}, {WM002}}, $\Theta^{q'2}$ = {{WM002, BP002, BR001}}, $\Theta^{q'3}$ = {{BR001}}, and $\Theta^{q'4}$ = {{BP002, BP003}}. In general, the nesting formula for distance in Eq. (3) leads to such nesting relationship, written as

$$\Theta^q \to \cup_{q'} \Theta^{q'}. \tag{18}$$

The above example shows this condition: when $D' \leq 110$,

Θ^{q3} = {{WM002, BP002, BR001}, {BP002, BP003}}
$\to \Theta^{q'2}$ = {{WM002, BP002, BR001}} $\cup$ $\Theta^{q'4}$ = {{BP002, BP003}}.

Such evolution is unique, as indicated in the following theorem.

Theorem 1 (Evolution of Behavior Clusters) Given $\Theta^{q*} = \cup_{q=1,..,Q}\Theta^q = cluster(\{b(p, m)\}, D)$, $\Theta^{q'*} = \cup_{q'=1,..,Q'}\Theta^{q'} = cluster(\{b(p, m)\}, D')$ and $D' < D$ and $Q < Q'$. Then, for all $\Theta^{q'}$, there exists a unique Θ^q, $\Theta^q \rightarrow \Theta^{q'}$.

Proof: By definition, for a behavior $b^i \in \Theta^q$, its cluster center c^q satisfies $d(c^q, b^i) \leq D \leq d(c^k, b^i)$, $k = 1, 2, \ldots, N_q$, $k \neq q$, where $d(\cdot, \cdot)$ satisfies Lemma 2. When $d(c^q, c^{q'}) \leq D'$, for all $b^j \in \Theta^{q'}$, the fact that $d(c^{q'}, b^j) \leq D' < D \leq d(c^k, b^j)$, $k = 1, 2, \ldots, N_{q'}$, $k \neq q'$, ensures that $d(c^q, b^j) \leq D' < D \leq d(c^k, b^j)$, which means $b^i \in \Theta^q$. Then, $\Theta^{q'}$ evolvs from Θ^q. ❑

The collection all evolutions of Θ^q's in Θ^{q*} leads to a shorthand notation, $\Theta^{q*} \rightarrow \Theta^{q'*}$. With Theorem 1, the clusters of inter-transaction behaviors appear as a *behavior hierarchy*, $\boldsymbol{BH}(\boldsymbol{B}, \boldsymbol{R})$. Here, the behavior-cluster set $\boldsymbol{B} = \cup_{q=1,..,L}\Theta^{q*}$ is formed by behavior clusters Θ^q's with different granularities, and the cluster relationship set $\boldsymbol{R} = \{(\Theta^q, \Theta^{q'}) : \Theta^q \rightarrow \Theta^{q'}\}$ represents the cluster evolution. Fig. 2 shows the behavior hierarchy of the previous example. For convenience, each level of behavior hierarchy and its clusters can be referred to Θ^{q*} and Θ^{qs}, e.g., Θ^{0*} = {xxxx}, Θ^{1*} = {Θ^{11} = Wxxxx, Θ^{12} = Bxxxx}, Θ^{2*} = {Θ^{21} = Wxxxx, Θ^{22} = BPxxx_BRxxx, Θ^{23} = BRxxx}, Θ^{3*} = {Θ^{31} = WT001, Θ^{32} = WM002, Θ^{33} = {WM002, BP002, BR001}, Θ^{34} = {BR001, BP002}, Θ^{35} = BR001}. Then, we have $\Theta^{0*} \rightarrow \Theta^{1*} \rightarrow \Theta^{2*} \rightarrow \Theta^{3*}$.

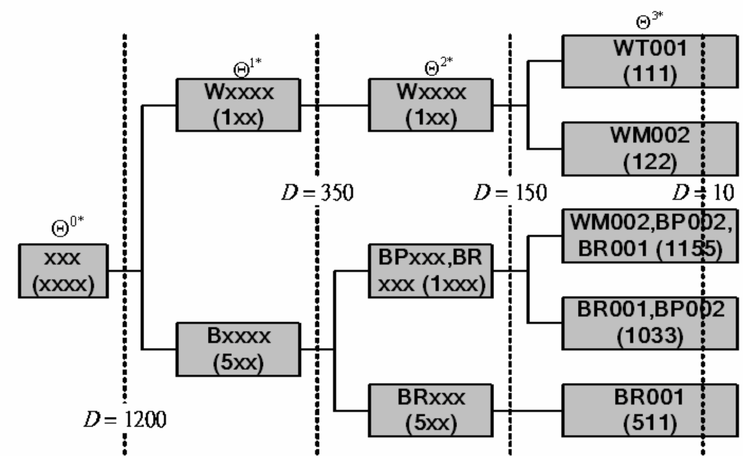

Fig. 2. Example behavior hierarchy.

4.3 Hierarchical State-Transition Graph

The straightforward method for conducting user clustering is to cluster users according to their behaviors. That is, given a cluster level q of user behaviors as the basis Θ^{q*}, the *transition probability matrix* can be extended to inter-transaction behaviors: $\boldsymbol{P}(\Theta^{q*}) = [P([\boldsymbol{\beta}^{i2} \mid \boldsymbol{\beta}^{i1}]), \boldsymbol{\beta}^{i1}, \boldsymbol{\beta}^{i2} \in \Theta^{q*}]$, where the conditional probability $P([\boldsymbol{\beta}^{i2} \mid \boldsymbol{\beta}^{i1}])$ is defined in Eq. (14). From the evolution of Θ^{q*}, the transition probability matrix also has the following property.

Corollary 1 (Evolution of Transition Probability Matrix) Given the inter-transaction behaviors $\Theta^{q^*} \rightarrow \Theta^{q'^*}$ and $\Theta^q = \{\boldsymbol{\beta}^i\} \rightarrow \Theta^{q'} = \{\boldsymbol{\beta}'^j\}$ as one of the in-between evolutions, it follows that

$$count(\boldsymbol{\beta}^{i1}, \boldsymbol{\beta}^{i2}) = \sum_{j1,j2} count(\boldsymbol{\beta}'^{j1}, \boldsymbol{\beta}'^{j2}) \tag{19}$$

and

$$count(\boldsymbol{\beta}^i) = \sum_j count(\boldsymbol{\beta}'^j). \tag{20}$$

Proof: This corollary is a direct derivation from Theorem 1 and the definition of user-clusters. ❑

Combined with the behavior evolution in Eqs. (18), (19) and (20) result in the evolution of transition probability matrices, denoted as $\boldsymbol{P}(\Theta^{q^*}) \rightarrow \boldsymbol{P}(\Theta^{q'^*})$. For the previous example, the most detailed behavior set Θ^{4^*} makes Eq. (15), renamed as $\boldsymbol{P}(\Theta^{3^*})$. For the behavior set $\Theta^{2^*} = \{\Theta^{\mathbf{21}}$ = Wxxxx, $\Theta^{\mathbf{22}}$ = BPxxxx_BRxxx, $\Theta^{\mathbf{23}}$ = BRxxx}, Eqs. (19) and (20) cause $\boldsymbol{P}(\Theta^{3^*})$ to be transformed into

$$\boldsymbol{P}(\Theta^{2^*}) = \sum \left[\begin{array}{cc|cc|c} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \end{array}\right] ./ \sum \left[\begin{array}{cc|cc|c} 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 \\ \hline 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \end{array}\right]$$
$$= \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} ./ \begin{bmatrix} 4 & 4 & 4 \\ 2 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

For the more concise form $\Theta^{2^*} = \{\Theta^{\mathbf{21}}$ = Wxxxx, $\Theta^{\mathbf{22}}$ = Bxxxx}, we have

$$\boldsymbol{P}(\Theta^{1^*}) = \sum \left[\begin{array}{c|cc} 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right] ./ \sum \left[\begin{array}{c|cc} 4 & 4 & 4 \\ \hline 2 & 2 & 2 \\ 1 & 1 & 1 \end{array}\right] = \begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix} ./ \begin{bmatrix} 4 & 4 \\ 3 & 3 \end{bmatrix} = \begin{bmatrix} 1/4 & 2/4 \\ 0 & 0 \end{bmatrix}.$$

Finally, $\boldsymbol{P}(\Theta^{0^*})$ = 3/7 = number(transitions)/number(behaviors). This gives $\boldsymbol{P}(\Theta^{0^*}) \rightarrow \boldsymbol{P}(\Theta^{1^*}) \rightarrow \boldsymbol{P}(\Theta^{2^*}) \rightarrow \boldsymbol{P}(\Theta^{3^*})$.

With the evolution of behaviors and the evolution of transition probability matrices, Definition 1 can be naturally extended to a *nested state-transition graph*. For our ongoing example, the representations are

$$\mathbf{STG}(\Theta^{0^*}, \boldsymbol{P}(\Theta^{0^*})) \rightarrow \mathbf{STG}(\Theta^{1^*}, \boldsymbol{P}(\Theta^{1^*})) \rightarrow \mathbf{STG}(\Theta^{2^*}, \boldsymbol{P}(\Theta^{2^*})) \rightarrow \mathbf{STG}(\Theta^{3^*}, \boldsymbol{P}(\Theta^{3^*})). \tag{21}$$

Such nesting and the in-between transitions are illustrated in Fig. 3.

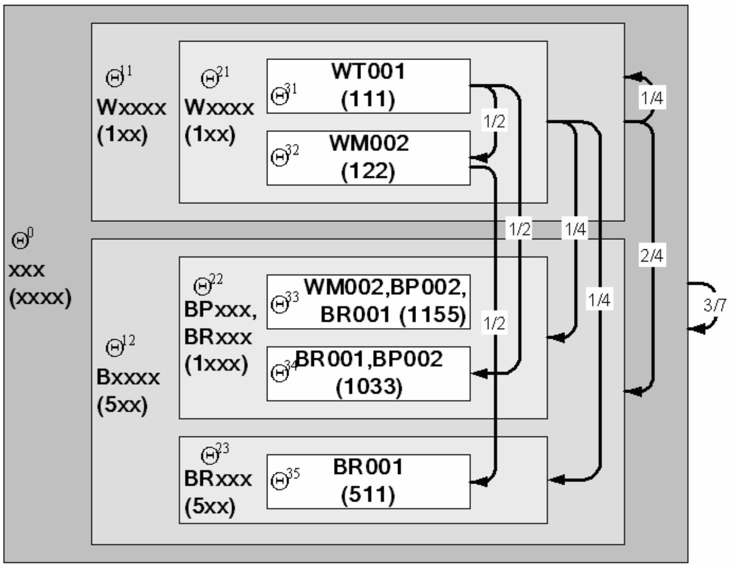

Fig. 3. Example of nested state-transition graphs.

5. EXPERIMENTS AND DISCUSSION

To prove our idea, two examples will be used to illustrate the above-mentioned algorithm. With simulated sequential data, the first example will show how AprioriAll algorithm can construct a set of nested state-transition graphs. The second example will show how this method can find the predictive models of a real data set of distance education and check the predictability of these models.

5.1 Example 1: From the Apriori Algorithm to State-Transition Graphs

In our first example, the standard simulation method of sequential data [2] is in the form show in Eq. (5). The hierarchical radix code of item taxonomy is [111, 112, 113, 121, 122, 131, 211, 212, 221, 222, 231, 232]. The simulation parameters are $N_I = 25$, $N_S = 50$, $|D| = 10$, $|I| = 1.25$, $|S| = 4$, $|T| = 2.5$, $|C| = 5$. For the simulated itemsets, the correlation level is 0.25, the weight exponential probability has a mean of 1, and an itemset has corruption_level with normal (0.95, 0.1).

In one simulation example, the resultant data sequence has 52 transactions for 10 customers. Applying the AprioriAll algorithm [18] with a minimal support value of 5, 36 maximal large sequences are found, including {222, 231} → 231 → 221, 113 → 231 → 131, {222, 231} → {222, 231}, 122 → 131 and {131, 211, 212}. The behaviors in these transactions can be coded in accumulated radix codes, as shown in Table 5.

In Table 5, there are 19 behaviors with minimal support values greater than or equal to 5, collected in a behavior set Θ^{3*} (since the radix level is 3). Of course, the item taxonomy creates a behavior hierarchy through leveled radix codes. According to the

Table 5. Behaviors from AprioriAll and their accumulated radix codes.

	Behaviors	acc. radix codes
one-action behaviors	112, 113, 122, 131, 211, 221, 222, 231	as left
two-action behaviors	{113, 131}	244
	{113, 231}	344
	{122, 131}	253
	{122, 231}	353
	{131, 211}	342
	{131, 212}	343
	{131, 231}	362
	{211, 231}	442
	{222, 231}	453
three-action behaviors	{131, 211, 212}	554
	{113, 221, 231}	565

k-element maximal large sequences across two transactions ($k \geq 2$), the transition matrix Θ^{l*} (l = 3) of 2-element behavior sequences $[\beta_l^{i1}, \beta_l^{i2}]$ with support values ≥ 5 can be obtained as shown in Table 6. It can be seen that the support counts of the transitions of all levels (l = 1, 2, 3) satisfy Eq. (19). Table 6 shows the corresponding transition probability matrices $\boldsymbol{P}(\Theta^{l*})$. The nested state-transition graphs can then be established.

5.2 Prediction Accuracy

The performance of the obtained STG models can be measured based on their prediction ability. Several indices can be defined for such measurement. As the finite-state machine model $\mathbf{STG}(\Theta^q, \boldsymbol{P}(\Theta^q))$ is used to explain (predict the future states correctly) these inter-transaction behaviors, Θ, $|\Theta_E(\mathbf{STG}(\Theta, \boldsymbol{P}(\Theta)))|$ is the total number of behaviors in $\mathbf{STG}(\Theta^q, \boldsymbol{P}(\Theta^q))$. Then, the *explaining capability* of $\mathbf{STG}(\Theta^q, \boldsymbol{P}(\Theta^q))$ can be defined as $\kappa(\mathbf{STG}(\Theta^q, \boldsymbol{P}(\Theta^q))) = |\Theta_E(\mathbf{STG}(\Theta^q, \boldsymbol{P}(\Theta^q)))| / |\boldsymbol{P}(\Theta^q)|$, which indicates how many behaviors in data sequences can be explained by each element in a transition model. Furthermore, the *explaining range* of the model is indicated by $\chi(\mathbf{STG}(\Theta^q, \boldsymbol{P}(\Theta^q))) = |\Theta_E(\mathbf{STG}(\Theta^q, \boldsymbol{P}(\Theta^q)))| / |\Theta^q|$.

For the overall system, including different STG models, the overall performance can be defined using two parameters: the overall explaining capacity and the overall explaining range. The *overall explaining capability* of $\mathbf{STG}(\Theta, \boldsymbol{P}(\Theta))$ can be defined as $\gamma_{ALL} = \sum_q |\Theta_E(\mathbf{STG}(\Theta^q, \boldsymbol{P}(\Theta^q)))| / \sum_q |\boldsymbol{P}(\Theta^q)|$, which means that the average number of behaviors in all data sequences can be explained by each element in the overall transition model. Furthermore, the *overall explaining range* of the model is indicated by $\kappa_{ALL} = \sum_q |\Theta_E(\mathbf{STG}(\Theta^q, \boldsymbol{P}(\Theta^q)))| / \sum_q |\boldsymbol{P}(\Theta^q)|$.

Table 6. Elements, count, and probabilities of nested automata.

β_1^{i1}	β_1^{i2}	*count*	$P(\Theta^{1*})$	β_2^{i1}	β_2^{i2}	*count*	$P(\Theta^{2*})$	β_3^{i1}	β_3^{i2}	*count*	$P(\Theta^{3*})$
1	1	16	0.1618	11	13	5	0.0725	113	131	5	0.0725
				12	13	5	0.0725	122	131	5	0.0725
				13	13	6	0.0870	131	131	6	0.0870
	2	44	0.1857	11	23	5	0.0323	113	231	5	0.0323
				12	22	5	0.1064	122	221	5	0.1351
				13	21	5	0.2000	131	211	5	0.2000
					22	6	0.1277	131	221	6	0.1622
				13	23	23	0.1484	131	231	23	0.1484
2	1	61	0.6162	21	13	5	0.0725	211	131	5	0.0725
				22	11	5	0.2500	222	113	5	0.3333
					13	10	0.1449	222	131	5	0.0725
								221	131	5	0.0725
				23	11	10	0.5000	231	113	10	0.6667
					13	31	0.4493	231	131	31	0.4493
	2	114	0.4810	21	22	5	0.1064	211	221	5	0.1351
					23	15	0.0968	211	231	15	0.0968
				22	23	22	0.1419	221	231	22	0.1419
				23	21	15	0.5000	231	211	15	0.6000
					22	10	0.2128	231	221	10	0.2703
					23	47	0.3032	231	231	47	0.3032
	4	25	0.3731	22	45	5	0.0806	222	453	5	0.0806
				23	45	20	0.3226	231	453	20	0.3226
3	1	5	0.0505	34	13	5	0.0725	344	131	5	0.0725
	2	20	0.0844	34	23	10	0.0645	343	231	5	0.0323
								344	231	5	0.0323
				35	23	5	0.0323	353	231	5	0.0323
				36	23	5	0.0323	362	231	5	0.0323
	4	5	0.0746	34	45	5	0.0806	342	453	5	0.0806
4	1	20	0.2020	45	11	5	0.2500	453	112	5	1.0000
					13	15	0.2174	453	131	15	0.2174
	2	30	0.1266		22	5	0.1064	453	221	5	0.1351
					23	25	0.1613	453	231	25	0.1613
	4	11	0.1642	44	45	5	0.0806	442	453	5	0.0806
				45	45	6	0.0968	453	453	6	0.0968

5.3 Real Experiment on Distance Education

A complete set of data on distance education was collected to verify the effectiveness of the above algorithms. The data came from the experiment a course "Special Education" at Chung Yuan Christian University, held in the spring of 2001. Thirty-nine (= N_{user}) students took this course. Besides attending class every week, the students could view learning materials, hand-in assigned homework, and take examinations on the Internet. A website with the address http://SpecialEducation.cycu.edu.tw was established to provide different services to students. Such classification makes 95 pages ($\Lambda = \{\lambda\}$) in totals be coded hierarchically, as shown in Fig. 4. The corresponding level-1 and level-2 radix action codes were also parenthesized. There were a total of 7964 visiting records in the form of [UserID, RelativeTime, PageID] in the Web log, which was the database ***DB*** $= \{\boldsymbol{\beta}(\Psi_p, \lambda, t) : \Psi_p \in \boldsymbol{\Psi}, \lambda \in \boldsymbol{\Lambda}, t \in T\}$ to be mined.

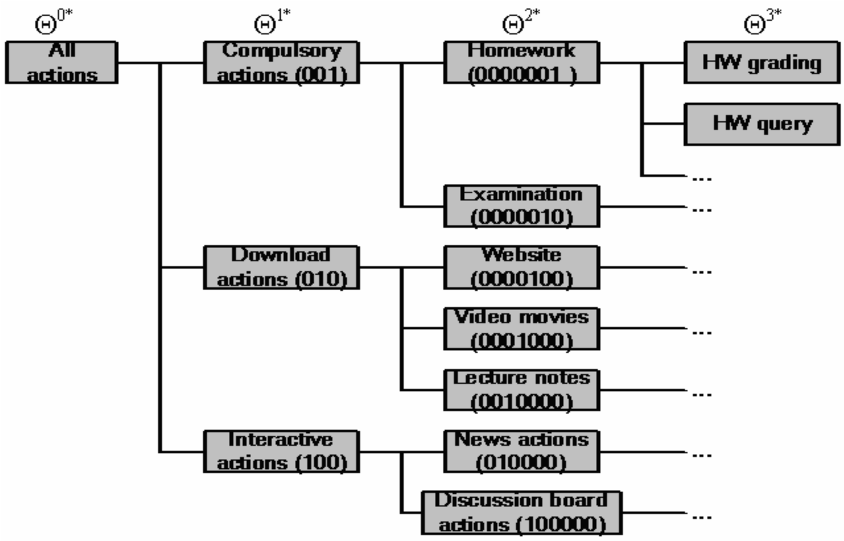

Fig. 4. Hierarchical page code for distance education.

With *DurationThreshold* = 10 and *TransactionBoundary* = 3600, Algorithm 1 calculated all the focused access actions in $\Theta_f(\Psi_p)$ of Eq. (3); there were [32, 24, 15, 12, 11, 20, 5, 10, 5, 13, 11, 64, 9, 20, 7, 4, 3, 3, 9, 15, 10, 8, 23, 5, 7, 7, 11, 10, 23, 6, 12, 16, 18, 9, 17, 29, 11, 21, 27] transactions for all 39 users. For each transaction, all the radix codes of access actions were accumulated using the code summarization strategy to get its behavior code. By means of these behavior codes, no matter whether they were level-1 codes or level-2 codes, all the users were classified into 6 clusters (communities):

Ω_1 = {2, 4, 6, 10, 23, 28, 29, 31, 36},
Ω_2 = {5, 7, 8, 9, 15, 19, 21, 25, 27, 30, 32, 34},
Ω_3 = {13, 16, 17, 18, 22, 24, 26, 37},
Ω_4 = {1},
Ω_5 = {3, 11, 14, 20, 33, 35, 38, 39},
Ω_6 = {12}.

Note that two special users (user 1 and user 12) formed two singleton clusters since these two users have extraordinary use (more than 30, even more than 60, transactions).

The users in a user-behavior cluster should have similar behaviors. For each user cluster, it is possible to get its users' transition probability matrices $[P([\beta^{i2} \mid \beta^{i1}])]$ from Eq. (14) and classify these transition matrices into several behavior models, which are sub-clusters (index s) of the original communities (index q). The STG behavior sets $\Theta^{qs} = \Theta(\Omega_{qs})$ and their transition probability matrices $\boldsymbol{P}(\Theta^{qs})$ of all communities are given in Table 7. One interpretation of these state-transition graphs is illustrated in Fig. 5. The above experiment a data included a total of 523 behaviors for 39 users. For the given 10 STG models with 49 behaviors in total, which was described all 250 behaviors, κ_{ALL} = 250/49 = 5.1020. The overall explaining range was γ_{ALL} = 250/523 = 0.4780, and for each (community, subgroup), we obtained the performance indices shown in the last two columns of Table 6. It can be observed that **STG**(Θ^{22}, $\boldsymbol{P}(\Theta^{22})$) and **STG**($\Theta^{52}$, $\boldsymbol{P}(\Theta^{52})$) were the worst ones and that **STG**(Θ^{32}, $\boldsymbol{P}(\Theta^{32})$) and **STG**($\Theta^{31}$, $\boldsymbol{P}(\Theta^{31})$) were the best ones.

Table 7. The state-transition graphs of all communities.

q	s	$\Omega_{qs} = \{\Psi_p\}$	Θ^{qs}	$P(\Theta^{qs})$					κ	γ
1	1	2, 23, 28, 36	001	0.48	0.35	0	0	0	6	0.5854
			011	0	0.45	0	0	0		
			101	1	0	0	0	0		
			111	0.63	0.38	0	0	0		
			211	0.57	0.43	0	0	0		
	2	4, 6, 10, 29, 31	001	0	1	0	0		6	0.5600
			011	0	0.31	0.69	0			
			111	0.35	0.41	0	0.24			
			211	0	0	1	0			
2	1	5, 30, 32, 34	001	1	0	0	0	0	4	0.6316
			011	0	0.4	0.3	0	0.3		
			101	0	0	0	1	0		
			111	0	1	0	0	0		
			211	0	0	0	0	0		
	2	7, 8, 9, 15, 19, 21, 25, 27	011	0	1				6	0.1071
			111	0	0					
3	1	13, 24, 26, 37	001	0.56	0.25				5.5	0.7857
			011	0.56	0.44					
	2	16, 17, 18, 22	001	0	0				6.5	0.9286
			011	0.23	0.77					
4	1	1	011	0	0	1			3.6	0.5806
			101	0	0.5	0.5				
			111	0	0.57	0.43				
5	1	3, 11, 20, 35, 38	011	1	0	0			5.6	0.3784
			111	0	0	1				
			211	0.25	0.25	0.3				
	2	14, 33, 39	011	0	1				4.33	0.2097
			111	0.4	0.6					
6	1	12	101	0	0	1	0		4.5	0.5714
			110	0.5	0	0.5	0			
			111	0.21	0	0.43	0.36			
			211	0	0	0.36	0.64			

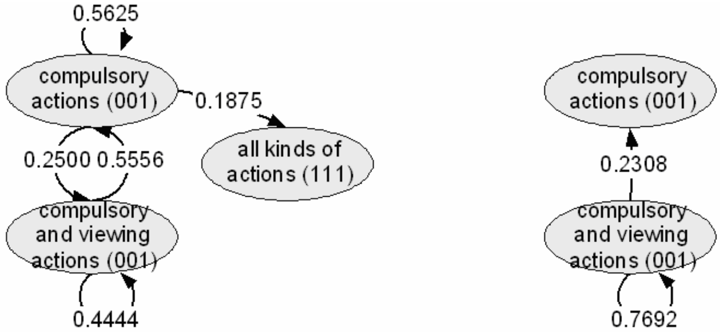

Fig. 5. State-transition graphs for 2 sub-communities (Ω_{31}, Ω_{32}) of Ω_3.

6. CONCLUSIONS

For Internet users, this paper has proposed an approach to transforming resource access actions into behavior models of state-transition graphs. User Ψ_p accesses an Internet resource through action $\boldsymbol{\lambda}$. Such an access action forms the behavior element $\boldsymbol{\beta}(p, \boldsymbol{\lambda}, t) = [\Psi_p, \boldsymbol{\lambda}, t]$ with radix code $code(\boldsymbol{\lambda}) = \sum_{l=1,..,L} \lambda_l r^{L-l}$. When the time differences are investigated, each data sequence can be divided into a series of (m, n) = (transaction-index, action-index) actions, $a^n(p, m)$. For each transaction of a user, the inter-transaction behavior $\boldsymbol{\beta}(p, m)$ can be found with level code $b_l(p, m)$; then, his/her data sequence becomes a behavior sequence, $\Theta(\Psi_p)$. Behavior sequences are aggregated as user-behavior codes and then used in user clustering to find communities, Ω_q's. For all the users in a community, the transition probability matrices between behaviors, $P_q(\boldsymbol{\beta}^{i2} \mid \boldsymbol{\beta}^{i1})$'s, are found and used to classify the subgroups (Ω_{qs}'s) of user transition behaviors. Finally, the state-transition graph of Ω_{qs} is found as $\mathbf{STG}(\Theta(\Omega_{qs}), \boldsymbol{P}(\Theta(\Omega_{qs})))$. In section 5, one example was used to show how state-transition graphs can be found through the successive steps in AprioriAll algorithm, and another example validated this kind of automata for a real-case distance education data set.

In our future work, one research effort will focus on the visualization of behavior models, which are then able to provide social researchers for further interpretation. Another research focus will be collected portfolios, whose behaviors models can be correlated with some property variables of users. We will explore which persons under which conditions have which behaviors. Finally, since the present behavior model is a temporal model with negligible time scale, it can be included in a future model.

ACKNOWLEDGMENTS

The authors wish to express their sincere thanks to the anonymous referees for their valuable comments. Moreover, this work was supported by the National Science Council, Taiwan, R.O.C., under Grant NSC 90-2520-S-033-001.

SYMBOL TABLE

Symbols for Data Sources

λ resource access action

$\boldsymbol{\Lambda}$ the set of resource access actions

Ψ_p user or individual

$\boldsymbol{\Psi}$ the set of all users

$\Omega_q \subseteq \boldsymbol{\Psi}$ user cluster, user group, or community

$\boldsymbol{\beta}(p, \lambda, t) = [\Psi_p, \lambda, t]$ behavior element

$\Theta = \{\boldsymbol{\beta}(p, \lambda, t) = [\Psi_p, \lambda, t]\}$ log database of behavior sequence

$\Theta(\Psi_p)$ user-behavior sequence of a specific user Ψ_p

$\Theta_a(\Psi_p) = \{[\Psi_p, \lambda, D(p, \lambda, m, n), m(p), n(m(p))]\}$ augmented behavior sequence

$\Theta_f(\Psi_p, m) \equiv \{[n, \lambda, t]\}$ transaction-indexed user behavior

$\Theta_{intra} = \{\lambda^n\} \equiv \Theta$ intra-transaction user-behavior

$\Theta_{inter}(\Psi_p) = \{\boldsymbol{\beta}(p, m) = \{\lambda^n\}\}$ inter-transaction user-behavior

Symbols for Data Preprocessing

$D(p, \lambda), D(p, \lambda, m, n)$ access duration

TransactionBoundary the next resource used in another transaction

DurationThreshold duration threshold for selecting actions

$TIND(\Psi_p) = m(\Psi_p) = m(p)$ transaction index

$AIND(\Psi_p, TIND(\Psi_p)) = n(m(p)$ action index

$Twindow = [t_i, t_f]$ time window for the ranges of actions

$RAcenter = \{\lambda_1, \ldots, \lambda_i, \ldots\}$ center of resource access actions

Symbols for Data Coding

$\lambda_1 \ldots \lambda_{level}(\lambda)$ level code of resource access actions

$a(\boldsymbol{\lambda}) = code(\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_l, \ldots, \lambda_L]) = \sum_{l=1,..,L} \lambda_l r^{L-l} = \boldsymbol{\lambda} \cdot \boldsymbol{r}$ radix code of resource access actions

$\boldsymbol{CH}(\Lambda = \{\boldsymbol{\lambda}_{1:l}\}, E_\Lambda)$, where $\Lambda = \{\boldsymbol{\lambda}_{1:l}\}$ and $E_\Lambda = \{\text{edge}(\boldsymbol{\lambda}_{1:l}, \boldsymbol{\lambda}_{1:l+1})\}$ code hierarchy

$\boldsymbol{RCH}(\{a(\boldsymbol{\lambda}_{1:l})\}, RE_\Lambda)$, where $RE_\Lambda = \{\text{edge}(a(\boldsymbol{\lambda}_{1:l}), a(\boldsymbol{\lambda}_{1:l+1}))\}$ radix code hierarchy

$b(p, m) = code(\boldsymbol{\beta}(p, m))$ (inter-transaction) behavior code

$d(a^i, a^j)$ distance between two radix action codes

$d(b^i_{SUM}, b^j_{SUM}),\ d(b^i_{\mathrm{MAX}}, b^j_{\mathrm{MAX}})$ distance between two behavior codes

Symbols for Data (Behavior) Models

$\Xi(\Theta(\Psi_p))$ user model

$\Xi(\{\Theta(\Omega_q)\})$ community behavior

$P(\boldsymbol{\beta}^{i2} \mid \boldsymbol{\beta}^{i1}) = P([\boldsymbol{\beta}^{i1}, \boldsymbol{\beta}^{i2}])/P(\boldsymbol{\beta}^{i1})$ conditional probability density functions

$\mathbf{STG}(\boldsymbol{\Theta}, \boldsymbol{P}), \boldsymbol{P} = [P([\boldsymbol{\beta}^{i2} \mid \boldsymbol{\beta}^{i1}])]$ state-transition graph of behaviors

$\Theta^{q*} = \cup_q \Theta^q = cluster(\Theta, D)$ the clustering of user behaviors

$\Theta^q \rightarrow \cup_{q'} \Theta^{q'}$ nesting of behavior clusters

$\boldsymbol{BH}(\boldsymbol{B}, \boldsymbol{R})$, where behavior-cluster set $\boldsymbol{B} = \cup_{q=1,..,L} \Theta^{q*}$ and $\boldsymbol{R} = \{(\Theta^q, \Theta^{q'}) : \Theta^q \rightarrow \Theta^{q'}\}$ behavior hierarchy

$\mathbf{STG}(\Theta^{l*}, \boldsymbol{P}(\Theta^{l*})) \rightarrow \mathbf{STG}(\Theta^{1+1*}, \boldsymbol{P}(\Theta^{1+1*}))$, evolutionary matrices $\boldsymbol{P}(\Theta^{q*}) \rightarrow \boldsymbol{P}(\Theta^{q'*})$ nested state-transition graph

Symbols for Model Verification

$\kappa(\mathbf{STG}(\Theta^q, \boldsymbol{P}(\Theta^q)))$ explaining capability of the behavior model

$\chi(\mathbf{STG}(\Theta^q, \boldsymbol{P}(\Theta^q)))$ explaining range of the behavior model

REFERENCES

1. R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of International Conference on Very Large Databases*, 1994, pp. 487-499.
2. R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of International Conference on Data Engineering*, 1995, pp. 3-14.
3. A. V. Aho and M. Corasick, "Efficient string matching: an aid to bibliographic search," *Communications of the ACM*, Vol. 18, 1975, pp. 333-340.
4. M. H. Dunham, *Data Mining Introductory and Advanced Topics*, Prentice-Hill, New Jersey, 2003.
5. U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery: an overview," U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, (eds.), *Advances in Knowledge Discovery and Data Mining*, MIT Press, 1996, pp. 1-34.
6. M. N. Garofalakis, R. Rastogi, and K. Shim, "SPIRIT: sequential pattern mining with regular expression constraints," in *Proceedings of 25th International Conference on Very Large Data Bases* (*VLDB '99*), 1999, pp. 223-234.
7. N. Halbwachs, *Synchronous Programming of Reactive Systems*, Kluwer Academic Publishers, 1993.
8. S. Y. Cheng, N. C. Ma, and J. S. Heh, "Nested state-transition graph model of user behaviors," in *Proceedings of 4th International Conference on Intelligent Data Engineering and Automated Learning*, 2003, pp. 21-23.
9. M. Klemettinen, "A knowledge discovery methodology for telecommunication network alarm databases," Ph.D. Thesis, Dept. of Computer Science, University of Helsinki, Finland, 1999.
10. G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice-Hall Inc., New Jersey, 1995.
11. H. Mannila, H Toivonen, and A. I. Verkamo, "Discovering frequent episodes in sequences," in *Proceeding of 1st International Conference on Knowledge Discovery and Data Mining*, 1995, pp. 210-215.
12. MathWorkers Inc., *MATLAB Statistics Toolbox – User's Guide*, MathWorks Inc., 1993.
13. R. E. Mayer, *Educational Psychology*, C. S. Lin translated, Yuan-Liou Publishing Company, Taiwan, 1991.
14. E. M. McCreight, "A space-economical suffix tree construction algorithm," *Journal of ACM*, Vol. 23, 1976, pp. 262-272.
15. K. L. Moore, "Iterative learning control for deterministic systems," *Advances in Industrial Control*, Springer-Verlag, 1993.
16. M. Negnevitsky, *Artificial Intelligence*: *A Guide to Intelligent Systems*, Edinburgh Gate Harlow England, Person Education, 2002.
17. L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE*

Acoustic, Speech, and Signal Processing Magazine, Vol. 3, 1986, pp. 4-17.
18. R. Srikant and R. Agrawal, "Mining sequential patterns: generalizations and performance improvements," in *Proceedings of 5th International Conference on Extending Database Technology*, 1996, pp. 3-17.
19. J. Srivasta, R. Cooley, M. Deshpande, and P. Tan, "Web usage mining: discovery and applications of usage patterns from web data," *SIGKKD Explorations*, Vol. 1, 2000, pp. 12-23
20. IBM, *Surfaid Analytics*, http://surfaid.dfw.ibm.com/web/home/index.html, 2000.
21. A. K. H. Tung, H. Lu, J. Han, and L. Feng, "Breaking the barrier of transactions: Mining inter-transaction association rules," in *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 297-301.
22. Y. Xiao and M. H. Dunham, "Efficient mining of traversal patterns," *Data and Knowledge Engineering*, Vol. 39, 2001, pp. 191-214.
23. O. Zaiane, "Resource and knowledge discovery from the internet and multimedia repositories," Technical Report, Ph.D. Dissertation, Dept. of Computing Science, Simon Fraser University, 1999.
24. M. J. Zaki, "Efficient enumeration of frequent sequences," in *Proceedings of 7th International Conference on Information and Knowledge Management*, 1998, pp. 68-75.

Jia-Sheng Heh (賀嘉生) received his B.S and M.S. degrees from the Department of Electrical Engineering from National Taiwan University in 1981 and 1983. He was a co-engineer at Chung Shan Institute of Science and Technology from 1983 to 1989. He was in the Ph.D. program in Electrical Engineering at National Taiwan University between 1988 and 1993. He is now an Associate Professor in the Department of Information and Computer Engineering in Chung Yuan Christian University. His current research interests include distance learning, mobile learning, neural network, fuzzy system, data mining, and computational intelligence.

Shein-Yung Cheng (鄭憲永) graduated with B.S. from the Department of Information and Computer Engineering and M.S. degree from the Department of Electronic Engineering from Chung Yuan Christian University, in 1987 and 1992, respectively. He is now a Ph.D. Candidate in the Department of Electronic Engineering and a full-time Lecturer of Department of Information and Computer Engineering of Chung Yuan Christian University. His research interests include data mining, e-learning, and learning technology.

Chang-Kai Hsu (徐長愷) was born in Kaohsiung, Taiwan, on April 17, 1973. He received the B.S. degree in the Department of Computer Science from the Chung Hua University in 1997, and the M.S. degree in the Department of Information and Computer Engineering from Chung Yuan Christian University in 1999, with a thesis on graph entropy evaluating for web-structure clustering. He is currently a Ph.D. candidate, and a Lecturer in the Department of Electrical Engineering, University of Chung Yuan Christian University. His current research interests include semantic web, data mining, and intelligent agent.