Mining Evolving Web Clickstreams with Explicit Retrieval Similarity Measures

Olfa Nasraoui
Department of Electrical and Computer Engineering
The University of Memphis
206 Engineering Science Bldg., Memphis, TN 38152
onasraou@memphis.edu

Cesar Cardona
Department of Electrical and Computer Engineering
The University of Memphis
206 Engineering Science Bldg., Memphis, TN 38152
ccardona@memphis.edu

Carlos Rojas
Department of Electrical and Computer Engineering
The University of Memphis
206 Engineering Science Bldg., Memphis, TN 38152
crojas@memphis.edu

ABSTRACT

Data on the Web is noisy, huge, and dynamic. This poses enormous challenges to most data mining techniques that try to extract patterns from this data. While scalable data mining methods are expected to cope with the size challenge, coping with evolving trends in noisy data in a continuous fashion, and without any unnecessary stoppages and reconfigurations is still an open challenge. This dynamic and single pass setting can be cast within the framework of mining evolving data streams. The harsh restrictions imposed by the *"you only get to see it once"* constraint on stream data calls for different computational models that may bring some interesting surprises when it comes to the behavior of some well known similarity measures during clustering. In this paper, we explore the task of mining evolving clusters in a single pass with a new scalable immune based clustering approach (TECNO-STREAMS), and study the effect of the choice of different similarity measures on the mining process and on the interpretation of the mined patterns. We propose a simple similarity measure that has the advantage of explicitely coupling the precision and coverage criteria to the early learning stages, and furthermore requiring that the affinity of the data to the learned profiles or summaries be defined by the minimum of their coverage or precision, hence requiring that the learned profiles are simultaneously precise and complete, with no compromises. In our simulations, we study the task of mining evolving user profiles from Web clickstream data (web usage mining) in a single pass, and under different trend sequencing scenarios.

Keywords

artificial immune systems, unsupervised learning, clustering, stream data mining, web usage mining, text mining, mining evolving data

1. INTRODUCTION

Natural organisms exhibit powerful learning and processing abilities that allow them to survive and proliferate generation after generation in ever changing and challenging environments. The natural immune system is a powerful defense system that exhibits many signs of cognitive learning and intelligence [6]. In particular the acquired or adaptive immune system is comprised mainly of lymphocytes which are special types of white blood cells (*B-cells*) that detect and destroy pathogens, such as viruses and bacteria. Identification of a particular pathogen is enabled by soluble proteins on the cell surface, called *antigens*. Special *protein receptors* on the B-cell surface, called *antibodies* are specialized to react to a particular antigen by binding to this antigen. Lymphocytes are only activated when the bond exceeds a minimum strength that may be different for different lymphocytes. A stronger binding with an antigen induces a lymphocyte to clone more copies of itself, hence providing reinforcement. Mature lymphocytes form the long term memory of the immune system, and help recognize and fight similar antigens that may be encountered in the future. Therefore, the immune system can perform pattern recognition and associative memory in a continuous and decentralized manner.

Recently, data mining has put even higher demands on clustering algorithms. They now must handle very large data sets, leading to some scalable clustering techniques. However, most scalable clustering techniques such as BIRCH [27] and the scalable K-Means (SKM) [4] assume that clusters are clean of noise, hyper-spherical, similar in size, and span the whole data space. *Robust* clustering techniques have recently been proposed to handle noisy data. Another limitation of most clustering algorithms is that they assume that the number of clusters is known. However, in practice, the number of clusters may not be known. This problem is called *unsupervised clustering*. A recent explosion of applications generating and analyzing *data streams* has added new unprecedented challenges for clustering algorithms if they are to be able to track changing clusters in noisy data streams using only the new data points because storing past data is not even an option [2, 1, 5, 9].

Web usage mining [24, 26, 20, 7, 21, 3, 19, 22, 18, 17, 25] has recently attracted attention as a viable framework for extracting useful access pattern information, such as user profiles, from massive amounts of Web log data for the purpose of Web site personalization and organization. Most efforts have relied mainly on clustering or association rule discovery as the enabling data mining technologies. Typically, data mining has to be completely re-applied periodically and offline on newly generated Web server logs in order to keep the discovered knowledge up to date.

In [14], we proposed a new immune system inspired approach for clustering noisy multi-dimensional stream data, called TECNO-STREAMS (**T**racking **E**volving **C**lusters in **NO**isy **S**treams), that has the advantages of *scalability, robustness, and automatic scale estimation*. TECNO-STREAMS is a scalable clustering methodology that gleams inspiration from the natural immune system to be able to continuously learn and adapt to new incoming patterns by detecting an unknown number of clusters in evolving noisy data in a single pass.

In this paper, we study the possibility of mining evolving user profiles from Web clickstream data (web usage mining) in a single pass, and under different usage trend sequencing scenarios. We also study the effect of the choice of different similarity measures on the mining process and on the interpretation of the mined patterns. We propose a simple similarity measure that has the advantage of explicitely coupling the precision and coverage criteria to the early learning stages, and furthermore requiring that the affinity of the data to the learned profiles or summaries be defined by the minimum of their coverage or precision, hence requiring

WWW2004, May 17–22, 2004, New York, NY USA.
ACM xxx.xxx.

that the learned profiles are simultaneously precise and complete, with no compromises.

The rest of the paper is organized as follows. In Section 2, we describe the TECNO-STREAMS algorithm. and compare it to some existing scalable clustering algorithms. In Section 3, we describe how we can use TECNO-STREAMS to track evolving clusters in Web usage data, and illustrate using it for mining real Web clickstream data, while studying the effect of the choice of different similarity measures on mining and interpreting the evolving profiles. Finally, in Section 4, we present our conclusions.

2. TECNO-STREAMS (TRACKING EVOLVING CLUSTERS IN NOISY STREAMS)

The immune system (lymphocyte elements) can behave as an alternative biological model of intelligent machines, in contrast to the conventional model of the neural system (neurons). In particular, the Artificial Immune Network (AIN) model is based on Jerne's Immune Network theory [12]. The system consists of a network of B cell lymphocytes that summarize the learned model. The immune network consists of a set, $\mathcal{X}_B$, of artificial B-cells, as well as stimulating and suppressing links between them. Learning takes as input a set of antigen training data, $\mathbf{X}_a$, and tries to learn an optimal immune network consisting of linked B-Cells based on cloning operations as in nature. Each B-Cell represents a learned pattern that could be matched to or validated by an antigen/data item or another B-Cell in the network. A link between two B-Cells gets stronger if they are more similar. Data from the antigen training set is matched against a B-Cell based on a properly chosen similarity measure. This affects the B-Cell's stimulation level, which in turn affects both its outlook for survival, as well as the number of clones that it produces. Because clones are similar to their spawning parent, they together form a network of co-stimulated cells that can sustain themselves even long after the disappearance of antigen data that has initiated the cloning. However, this network of B-cells will slowly wither and die if it is no longer stimulated by the antigen data for which it has specialized, hence gradually forgetting old encounters. This forgetting is the reason why the immune system needs periodical reminders in the form of re-vaccination. The combined recall and forgetting behavior in the face of external antigenic agents forms the fundamental principle behind the concept of emerging or dynamic memory in the immune system. This is specifically the reason why the immune system metaphor offers a very competitive model within the evolving data stream framework. In the following description, we present a more formal treatment of the intuitive concepts explained above.

Here, we summarize the TECNO-STREAMS approach omitting some of the details and proofs that can be found in [14]. In a dynamic environment, the objects from a data stream $\mathbf{X}_a$ are presented to the immune network one at a time, with the stimulation and scale measures re-updated with each presentation. It is more convenient to think of the antigen index, j, as monotonically increasing with time. That is, the antigens are presented in the following chronological order: $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$. The Dynamic Weighted B-Cell (*D-W-B-cell*) represents an influence zone over the domain of discourse consisting of the training data set. However, since data is dynamic in nature, and has a temporal aspect, data that is more current will have higher influence compared to data that is less current. Quantitatively, the influence zone is defined in terms of a weight function that decreases not only with distance from the antigen/data location to the D-W-B-cell prototype, but also with the time since the antigen has been presented to the immune network. It is convenient to think of time as an additional dimension that is added to the D-W-B-Cell compared to the classical B-Cell, traditionally statically defined in antigen space only [15].

Definition 1: (Robust Weight/Activation Function) For the i^{th} D-W-B-cell, DWB_i, $i = 1, \cdots, N_B$, we define the activation caused by the j^{th} antigen data point, after J antigens have been presented, as

$$w_{ij} = w_i\left(d_{ij}^2\right) = e^{-\left(\frac{d_{ij}^2}{2\sigma_{i,j}^2}+\frac{(J-j)}{\tau}\right)} \quad (1)$$

where τ controls the time decay rate of the contribution from old antigens, and hence how much emphasis is placed on the currency of the immune network compared to the sequence of antigens encountered so far. d_{ij}^2 is the distance from antigen $\mathbf{x}_j$ (which is the j^{th} antigen encountered by the immune network) to D-W-B-cell, DWB_i. $\sigma_{i,j}^2$ is a scale parameter that controls the decay rate of the weights along the spatial dimensions, and hence defines the size of an influence zone around a cluster prototype. Data samples falling far from this zone are considered outliers. The weight functions decrease exponentially with the order of presentation of an antigen, j, and therefore, will favor more current data in the learning process.

Definition 2: (Influence Zone) The i^{th} D-W-B-cell represents a soft influence zone, $\mathbf{IZ}_i$, that can be interpreted as a robust zone of influence, consisting of all the data points that succeed in acticating this cell.

$$\mathbf{IZ}_i = \left\{\mathbf{x}_j \in \mathbf{X}_a | w_{ij} \geq w_{min}\right\}, \quad (2)$$

Each D-W-B-cell is allowed to have is own zone of influence with radial size proportional to σ_i^2, that is dynamically estimated. Hence, outliers are easily detected as data points falling outside the influence zone of all D-W-B-cells or through their weak activations ($w_{ij} < w_{min}, \quad \forall i$).

Definition 3: (Pure Stimulation) The stimulation level, after J antigens have been presented to DWB$_i$, is defined as the density of the *antigen* population around DWB$_i$:

$$s_{a\,i,J} = \frac{\sum_{j=1}^{J} w_{ij}}{\sigma_{i,J}^2}, \quad (3)$$

Lemma 1: (Optimal Scale Update) [14] The equations for optimal scale updates are given by

$$\sigma_{i,J}^2 = \frac{\sum_{j=1}^{J} w_{ij} d_{ij}^2}{2\sum_{j=1}^{J} w_{ij}}. \quad (4)$$

For the purpose of computational efficiency, however, we convert the above equations to incremental counterparts as follows.

Lemma 2: (Incremental Update of Pure Stimulation and Optimal Scale) After J antigens have been presented to DWB_i, pure stimulation and optimal scale can be updated using the following approximate incremental equations, respectively,

$$s_{a\,i,J} = \frac{e^{-\frac{1}{\tau}} W_{i,J-1} + w_{iJ}}{\sigma_{i,J}^2}, \quad (5)$$

$$\sigma_{i,J}^2 = \frac{e^{-\frac{1}{\tau}} \sigma_{i,J-1}^2 W_{i,J-1} + w_{iJ} d_{iJ}^2}{2\left(e^{-\frac{1}{\tau}} W_{i,J-1} + w_{iJ}\right)}. \quad (6)$$

where $W_{i,J-1} = \sum_{j=1}^{J-1} w_{ij}$ is the sum of the contributions from previous antigens, $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{J-1}$, to D-W-B-Cell i.

2.1 Dynamic Stimulation and Suppression

We propose incorporating a dynamic stimulation factor, $\alpha(t)$, in the computation of the D-W-B-cell stimulation level by adding a compensation term that depends on other D-W-B-cells in the network [11, 23]. In other words, a group of co-stimulated D-W-B-cells can self-sustain themselves in the immune network, even after the antigen that caused their creation disappears from the environment. However, we need to put a limit on the time span of this memory to forget truly outdated patterns. This is done by allowing D-W-B-cells to have their own stimulation coefficient, and to have this stimulation coefficient decrease with their age: $\alpha(t) = \frac{1}{1+\frac{t}{\tau_\alpha}}$. We also incorporate a dynamic suppression factor, $\beta(t) = \frac{1}{1+\frac{t}{\tau_\beta}}$, to control the proliferation and redundancy of the D-W-B-cell population.

2.2 Bridging the Scalability Gap: Organization and Compression of the Immune Network

The number of possible internal interactions (between different cells in the network) can be a serious bottleneck in the face of all existing immune network based learning techniques [11, 23]. Suppose that the immune network is compressed by clustering the D-W-B-cells using a linear complexity approach such as K Means. Then the immune network can be divided into K *subnetworks* that form a parsimonious view of the entire network. For global low resolution interactions, such as the ones between D-W-B-cells that are very different, only the *inter-subnetwork interactions* are germane. For higher resolution interactions such as the ones between similar D-W-B-cells, we can drill down inside the corresponding subnetwork and afford to consider all the *intra-subnetwork interactions*.

Lemma 3: (Effect of Network Compression on Scalability) The proposed AIS based clustering model can achieve scalability at a finite compression rate ($K \to \sqrt{N_B}$).

2.3 Effect of the Network Compression on Interaction Terms

Instead of taking into account all possible $(N_B)^2$ interactions between all N_B cells in the immune network, only the intra-subnetwork interactions with the N_B^i D-W-B-cells inside the parent subnetwork (the closest subnetwork to which this B cell is assigned) are taken into account. In case K-Means is used, this representative as well as the organization of the network into subnetworks is a by-product. For more complex data structures, a reasonable best representative/prototype (such as a medoid) can be chosen. Taking these modifications into account, the stimulation and scale values that take advantage of the compressed network are given by

$$s_i = s_{a\,i,J} + \alpha(t)\frac{\sum_{l=1}^{N_B^i} w_{il}}{\sigma_{i,J}^2} - \beta(t)\frac{\sum_{l=1}^{N_B^i} w_{il}}{\sigma_{i,J}^2}, \tag{7}$$

where $s_{a\,i,J}$ is the pure antigen stimulation after encountering J antigens, given by (5) for D-W-B-cell$_i$; and N_B^i is the number of B-cells in the subnetwork that is closest to the i^{th} DWB-cell. This will modify the D-W-B-cell scale update equations to become

$$\sigma_{i,J}^2 = \frac{1}{2}\frac{D_{i,J}^2 + \alpha(t)\sum_{l=1}^{N_B^i} w_{il}d_{il}^2 - \beta(t)\sum_{l=1}^{N_B^i} w_{il}d_{il}^2}{W_{i,J} + \alpha(t)\sum_{l=1}^{N_B^i} w_{il} - \beta(t)\sum_{l=1}^{N_B^i} w_{il}}, \tag{8}$$

where $D_{i,J}^2 = e^{-\frac{1}{\tau}}\sigma_{i,J-1}^2 W_{i,J-1} + w_{iJ}d_{iJ}^2$ and $W_{i,J} = e^{-\frac{1}{\tau}}W_{i,J-1} + w_{iJ}$

2.4 Cloning in the Dynamic Immune System

The D-W-B-cells are cloned in proportion to their stimulation levels relative to the average network stimulation. However, to avoid preliminary proliferation of D-W-B-Cells, and to encourage a diverse repertoire, new D-W-B-Cells do not clone before they are mature (their age, t_i exceeds a lower limit t_{min}). Similarly, D-W-B-cells with age $t_i > t_{max}$ are frozen, or prevented from cloning, to give a fair chance to newer D-W-B-Cells. This means that $N_{clones_i} = K_{clone}\frac{s_i}{\sum_{k=1}^{N_B} s_k}$ if $t_{min} \leq t_i \leq t_{max}$.

2.5 TECNO-STREAMS: Tracking Evolving Clusters in Noisy Data Streams with a Scalable Immune System Learning Model

TECNO-STREAMS Algorithm:
(optional steps are enclosed in [])

```
Fix the maximal population size, $N_{B\,max}$;
Initialize D-W-B-cell population and $\sigma_i^2 = \sigma_{init}$ using the first $N_{B\,max}$ input antigens;
Compress immune network into $K$ subnets using 2 iterations of K Means;
Repeat for each incoming antigen $\mathbf{x}_j$ {
  Present antigen to each subnet centroid, $\mathbf{C}_k, k = 1, \cdots, K$ in network : Compute distance, activation weight, $w_{kj}$ and update $\sigma_k^2$ incrementally using (6);
  Determine the most activated subnet (the one with maximum $w_{kj}$);
  IF All B-cells in most activated subnet have $w_{ij} < w_{min}$ (antigen does not sufficiently activate subnet) THEN{
    Create by duplication a new D-W-B-cell = $\mathbf{x}_j$ and $\sigma_i^2 = \sigma_{init}$;
  }
  ELSE {
    Repeat for each D-W-B-cell$_i$ in most activated subnet {
      IF $w_{ij} > w_{min}$ (antigen activates D-W-B-cell$_i$) THEN
        Refresh age ($t = 0$) for D-W-B-cell$_i$;
      ELSE
        Increment age (t) for D-W-B-cell$_i$;
      Compute distance from antigen $\mathbf{x}_j$ to D-W-B-cell$_i$;
      Compute D-W-B-cell$_i$'s stimulation level using (7);
      Update D-W-B-cell$_i$'s $\sigma_i^2$ using (8);
    }
  }
  Clone and mutate D-W-B-cells;
  IF population size $> N_{B\,max}$ Then {
    IF (Age of B-cell $< t_{min}$) THEN
      Temporarily scale D-W-B-cell's stimulation level to the network average stimulation;
    Sort D-W-B-cells in ascending order of their stimulation level;
    Kill worst excess (top $(N_B - N_{B\,max})$ according to previous sorting) D-W-B-cells;
    [or move oldest/mature D-W-B-Cells to secondary (long term) storage];
  }
  Compress immune network periodically (after every $T$ antigens), into $K$ subnets using 2 iterations of K Means with the previous centroids as initial centroids;
}
```

2.6 Comparison to Other Clustering Techniques

Because of paucity of space, we review only some related methods, as summarized in Table 1. We note that all immune based techniques, as well as most evolutionary type clustering techniques are expected to benefit from insensitivity to initial conditions (*reliability*) by virtue of being population based. Moreover, most techniques achieve their scalability by using a special indexing structure which requires an additional preliminary scan of the data which may not be acceptable in the context of data streams.

3. MINING EVOLVING USER PROFILES FROM NOISY WEB CLICKSTREAM DATA

Recently, data mining techniques have been applied to extract usage patterns from Web log data [24, 26, 20, 7, 21, 19, 22, 3, 18, 17, 25]. In [19, 18], we have proposed new robust and fuzzy relational clustering techniques that allow Web usage clusters to overlap, and that can detect and handle outliers in the data set. A new subjective similarity measure between two Web sessions, that captures the organization of a Web site, was also presented as well as a new mathematical model for "robust" Web user profiles [19] and quantitative evaluation means for their validation. Unfortunately, the computation of a huge relation matrix added a heavy computational and storage burden to the clustering process.

In [17], we presented a *quasi-linear* complexity technique, called Hierarchical Unsupervised Niche Clustering (H-UNC), for mining both user profile clusters and URL associations in a *single* step. More recently, we have presented a new approach to mining user profiles that

Table 1: Comparison of proposed Scalable Immune Clustering Approach with Other Algorithms

Approach $\rightarrow$	*TECNO-STREAMS*	*SKM* [4]	*DBSCAN* [8]	*DENCLUE* [10]	*BIRCH* [27]
Reliabibilty/Insensitivity to initialization	yes	no	yes	no	no
Robustness to noise	yes	no	yes	yes	no
Requires Pre-Clustering scan/ Spatial Data Structure	no	yes (integrated)	yes (R^*-tree)	yes (B^+-tree)	yes (CF-tree)
Time Complexity: $O()$	N	N	$N \log(N)$	$\log(N)$	N
Requires Buffer for Data	no	yes	yes	yes	yes
Requires No. of Clusters	no	yes	no	yes	no
Handles evolving clusters	yes	no	no	no	no
Robust Automatic Scale Estimation	yes	no	no	no	no
Cluster Model	network	centroids	medoids	centroids	centroids
Handles Arbitrary Dissimilarity Measures	yes	no	yes	no	no
Density/Partition based?	Density	Partition	Density	Density	Partition

is inspired by concepts from the natural immune system [15]. This approach proved to be successful in mining clusters and frequent itemsets from large web session data. This kind of data, which is extremely sparse, presents a real challenge to conventional clustering and frequent itemset mining techniques. Many data sets share this sparsity with clickstream data: these include text data as well as a large number of transactional databases. Unfortunately, all the above methods assume that the entire preprocessed Web session data could reside in main memory. This can be a disadvantage for systems with limited main memory in case of huge web session data, since the I/O operations would have to be extensive to shuffle chunks of data in and out, and thus compromise scalability. Today's web sites are a source of an exploding amount of clickstream data that can put the scalability of any data mining technique into question.

Moreover, the Web access patterns on a web site are very dynamic in nature, due not only to the dynamics of Web site content and structure, but also to changes in the user's interests, and thus their navigation patterns. The access patterns can be observed to change depending on the time of day, day of week, and according to seasonal patterns or other external events in the world. As an alternative to locking the state of the Web access patterns in a frozen state depending on when the Web log data was collected and preprocessed, we propose an approach that considers the Web usage data as a reflection of a dynamic environment which therefore requires dynamic learning of the access patterns. An intelligent Web usage mining system should be able to continuously learn in the presence of such conditions without ungraceful stoppages, reconfigurations, or restarting from scratch. In this section, we illustrate using TECNO-STREAMS to continuously and dynamically learn evolving Web access patterns from non-stationary Web usage environments.

3.1 Similarity Measures Used in the Learning Phase of Single-Pass Mining of Clusters in Web Data

For many data mining applications such as clustering *text* documents and other *high dimensional* data sets, the Euclidean distance measure is not appropriate. This is due mainly to the high dimensionality of the problem, and the fact that two documents may not be considered similar if keywords are missing in both documents. More appropriate for this application, is the cosine similarity measure between data item $\mathbf{x}_i$ and a learned B-Cell profile $\mathbf{p}_j$, which in the simplest case, can both be defined as binary vectors of length n, the total number of items/URLs or keywords, [13],

$$S_{cos\,ij} = \frac{\sum_{k=1}^{n} x_{ik} \times p_{jk}}{\sqrt{\sum_{k=1}^{n} x_{ik} \sum_{k=1}^{n} p_{jk}}}. \tag{9}$$

We note that it is easy to show that the cosine similarity is related to the well known information retrieval measures of precision and coverage as follows:

$$S_{cos\,ij} = \sqrt{Prec^{L}{}_{ij}\, Covg^{L}{}_{ij}}, \tag{10}$$

where the precision in the learning phase, $Prec^{L}{}_{ij}$ describes the accuracy of the learned B-cell profiles $\mathbf{p}_j$ in representing the data $\mathbf{x}_i$, or the ratio of the number of matching items (URLs or terms) between the learned profile and the data (session or document) to the number of items in the learned profile:

$$Prec^{L}{}_{ij} = \frac{\sum_{k=1}^{n} x_{ik} \times p_{jk}}{\sum_{k=1}^{n} p_{jk}}, \tag{11}$$

while the coverage in the learning phase, $Covg^{L}{}_{ij}$ describes the completeness of the learned B-cell profiles $\mathbf{p}_j$ in representing the data $\mathbf{x}_i$, or the ratio of the number of matching items (URLs or terms) between the learned profile and the data (session or document) to the number of items in the data:

$$Covg^{L}{}_{ij} = \frac{\sum_{k=1}^{n} x_{ik} \times p_{jk}}{\sum_{k=1}^{n} x_{jk}}. \tag{12}$$

In light of (10), we can see that the cosine similarity tries to optimize both precision and coverage simultaneously and equally by combining them through the geometrical average. However, we noticed that when learning in a single pass framework, this tends to favor longer profiles that tend to match more data, while compromising precision. Without loss of generality, if we confine ourselves to the simplest type of recommendation strategy or information retrieval scheme, we can see that compromising precision can have a pernicious effect on the learned profiles, especially when these are viewed as the cluster or profile summaries that will be used later in a recommendation system based on recommending the nearest profile, or in an information retrieval system based on matching a user query to the nearest cluster representative centroid. In order to circumvent this problem, one can simply disregard the coverage component from the cosine similarity, hence using only precision as a similarity measure. However, we noticed that this would tend to suffer from the other extreme, resulting in very short profiles that completely ignore coverage. For this reason, we propose to use different combination strategies of precision and coverage, not necessarily limited to the geometrical average. It can be shown that the most conservative aggregation that places harsh demands on both precision and coverage *simultaneously* must be given by the following pessimistic aggregation,

$$S_{min\,ij} = \min\left\{Prec^{L}{}_{ij}, Covg^{L}{}_{ij}\right\} \tag{13}$$

Therefore, we will compare learning the profiles using cosine similarity S_{cos} to learning using the most pessimistic aggregation of precision and coverage, called *Min-Of-Precision-Coverage* or *MinPC, $S_{min\,ij}$*.

3.2 Similarity Measures Used in the Validation Phase of Single-Pass Mining of Clusters in Web Data

In evaluating the goodness of the learned B-Cell profiles that make up the immune network model, we recall that the B-cell profiles should represent the ground-truth trends as accurately as possible, and as completely as possible, and that the distribution of the learned repertoire of B-cell profiles should mirror the incoming stream of evolving data as represented by the ground truth profiles/topic representatives. Accuracy can be measured based on the precision of the learned B-cell profiles, $\mathbf{p}_{L_j}$ relative to the ground truth profiles $\mathbf{p}_{GT_i}$, while completeness can be measured based on coverage of the learned B-cell profiles, $\mathbf{p}_{L_j}$ relative to the ground truth profiles $\mathbf{p}_{GT_i}$. Here, precision in the validation phase, $Prec^v{}_{ij}$ describes the accuracy of the B-cell profiles $\mathbf{p}_{L_j}$ in representing the ground truth profiles $\mathbf{p}_{GT_i}$, or the ratio of the number of matching items (URLs or terms) between the learned profile and the ground truth profiles to the number of items in the learned profile:

$$Prec^v{}_{ij} = \frac{\sum_{k=1}^{n} p_{L_{ik}} \times p_{GT_{jk}}}{\sum_{k=1}^{n} p_{L_{jk}}}, \tag{14}$$

while the coverage in the validation phase, $Covg^v{}_{ij}$ describes the completeness of the B-cell profiles $\mathbf{p}_j$ in representing the data $\mathbf{x}_i$, or the ratio of the number of matching items (URLs or terms) between the learned profile and the data (session or document) to the number of items in the data:

$$Covg^v{}_{ij} = \frac{\sum_{k=1}^{n} p_{L_{ik}} \times p_{GT_{jk}}}{\sum_{k=1}^{n} p_{GT_{jk}}}. \tag{15}$$

These are the measures that are computed as TECNO-STREAMS continuously learns the profiles from the incoming stream of web sessions or text documents.

3.3 Simulation Results with Single-Pass Mining of User Profiles from Real Web Clickstream Data

Profiles were mined from the 12-day clickstream data (from 1998) with 1704 sessions and 343 URLs from the website of the department of Computer Engineering and Computer Science at the University of Missouri. This is a benchmark data set used in [16, 17]. The profiles that were discovered using TECNO-STREAMS in a single pass are comparable to the ones previously obtained using a variety of less scalable techniques [16, 17]. The maximum population size was 50, the control parameter for compression was $K = 10$, with periodical compression every $T = 10$ sessions. The activation threshold was $w_{min} = 0.375$, and $\tau = 100$. We illustrate the *continuous* learning ability of the proposed technique using the following simulations:

Scenario 1: We partition the Web sessions into 20 distinct sets of sessions, each one assigned to the closest of 20 profiles previously discovered and validated using Hierarchical Unsupervised Niche Clustering (HUNC) [17], and listed in Table 2. Then we presented these sessions to TECNO-STREAMS one profile at a time: sessions assigned to trend 0, then sessions assigned to profile 1, $\cdots$, etc.

Scenario 2: We used the same session partition as scenario 1, but presented the profiles in reverse order: sessions assigned to trend 19, then sessions assigned to trend 18, $\cdots$, etc, ending with trend 0.

Scenario 3: The Web sessions are presented in their natural chronological order exactly as received in real time by the web server.

For each of the above scenarios, we repeated the experiment using cosine similarity $S_{cos\,ij}$ in learning as given by (9), and then again using the *MinPC* similarity $S_{min\,ij}$ as given by (13).

We track the number of B-cells that succeed in learning each one of the 20 ground truth profiles after each session is presented, by counting the number of B-cells registering a sufficient match (i.e., above a certain threshold) with each ground truth profile based on one of the following criteria: **(i)** precision $Prec^v{}_{ij}$, measuring the accuracy of the learned profiles compared to the ground truth profiles as given by (14), **(ii)** coverage $Covg^v{}_{ij}$, measuring the completeness of the learned profiles compared to the ground truth profiles as given by (15). These two measures provide an evolving number of hits per profile relative to each of the above criteria, as shown in Figures 2 - 7, for the two different learning similarity options, and the three above scenarios respectively.

Table 2: Summary of some usage trends previously discovered using Hierarchical Unsupervised Niche Clustering (only URLs with top 3 to 4 relevance weights shown in each profile)

i	$\lvert P_{T\,i} \rvert$	$P_{T\,i}$
0	106	{0.99 - /people_index.html}, {0.98 - /people.html}, {0.97 - /faculty.html}
1	104	{0.99 - /}, {1.00 - /cecs_computer.class}
2	177	{0.90 - /courses_index.html}, {0.88 - /courses100.html}, {0.87 - /courses.html} , {0.81 - /}
3	61	{0.80 - /}, {0.48 - /degrees.html}, {0.23 - /degrees_grad.html}
4	58	{0.97 - /degrees_undergrad.html}, {0.97 - /bsce.html}, {0.95 - /degrees_index.html}
5	50	{0.56 - /faculty/springer.html}, {0.38 - /faculty/palani.html}
6	116	{0.91 - /~saab/cecs333/private}, {0.78 - /~saab/cecs333}
12	74	{0.57 - /~shi/cecs345}, {0.45 - /~shi/cecs345/java_examples}, {0.46 - /~shi/cecs345/Lectures/07.html}
13	38	{0.82 - /~shi/cecs345}, {0.47 - /~shi}, {0.34 - /~shi/cecs345/references.html}
14	33	{0.55 - /~shi/cecs345}, {0.55 - /~shi/cecs345/java_examples}, {0.33 - /~shi/cecs345/Projects/1.html}
15	51	{0.92 - /courses_index.html} , {0.90 - /courses100.html}, {0.86 - /courses.html}, {0.78 - /courses200.html}
16	77	{0.78 - /~yshang/CECS341.html}, {0.56 - /~yshang/W98CECS341}, {0.29 - /~yshang}
19	120	{0.27 - /access} , {0.23 - /access/details.html}

The y-axis is split into 20 intervals, with each interval devoted to the trend/profile number indicated by the lower value (from 0 to 19). A hit for the i^{th} profile for session No. t is shown in these figures at location (t, i), and indicates the presence of at least one B-cell profile that achieved the desired threshold in the validation measures of precision or coverage.

The proposed immune clustering algorithm can learn the user profiles in a single pass. A single pass over all 1704 Web user sessions (with non-optimized Java code) took less than 7 seconds on a 2 GHz Pentium 4 PC running on Linux. With an average of *4 milliseconds per user session*, the proposed profile mining system is suitable for use in a real time personalization system to constantly and continuously provide a fresh and current list of an unknown number of evolving user profiles. Old profiles can be handled in a variety of ways. They may either be discarded, moved to secondary storage, or cached for possible re-emergence. Even if discarded, older profiles that re-emerge later, would be re-learned from scratch just like new profiles. Hence the logistics of maintaining old profiles are less crucial compared to existing techniques.

Figures 2 and 3 show the evolving hits per usage trend for the cosine similarity and the *MinPC* similarity, respectively when scenario 1 is deployed for sequencing the usage trends. They both exhibit an expected *staircase pattern* proving the gradual learning of emergent usage trends as these are experienced by the immune network in the order from trend 0 to 19. The plot shows some peculiarities, for example at trend 15 since it records hits at the same time as trends 0, 2, 3, and 5. Table 2 and the examination of the user sessions in each of these trends show that these trends do indeed share many similarities with trend 15, especially in terms of overlap. Typical cross reactions between similar patterns are actually desired and illustrate a certain tolerance for inexact matching.

Figures 2(a) and 3(a) show that the number of learned profiles satisfying more than 0.5 precision evolves in synchrony with the usage trends being presented. Furthermore, Figure 3(a) shows that the *MinPC* similarity allows learning and maintaining *high-precision* profiles longer than cosine similarity in Figure 2(a). For instance, compare the top 3 profiles in each figure corresponding to trends 17, 18, and 19 that are presented last in that sequence. Similarly, Figure 3(b) shows that the *MinPC* similarity allows learning more *high-coverage* profiles and can keep them longer than the plain cosine similarity in Figure 2(b). This can be seen in the top 5 profiles corresponding to trends 15, 16, 17, 18, and 19 that are the last to be encountered in that sequence.

Figures 4 and 5 show the evolving hits per usage trend for the cosine similarity and the *MinPC* similarity, respectively when scenario 2 is deployed for sequencing the usage trends. They show an interesting *inverted staircase* pattern due to the reverse presentation order. Again, comparing Figures 4(a) with Figure 5(a) shows that the *MinPC* similarity allows learning more *high-precision* profiles and can maintain them longer than cosine. Similarly, by contrasting Figure 5(b) and Figure 4(b), we can infer that the *MinPC* similarity allows learning more *high-coverage* profiles and can keep them longer.

Finally Figures 6 and 7 show the evolving hits per usage trend for the cosine similarity and the *MinPC* similarity, respectively when the sessions are presented in their original chronological order corresponding to scenario 3. In this case, the order of presentation of the trends is no longer sequenced in straight or reverse order of the trend number. Instead, the user sessions are presented in completely natural (chronological) order, exactly as in real time. So we cannot expect a staircase pattern. In order to visualize the *expected pattern*, we simply plot the distribution of the original input sessions, but with all the noise sessions excluded, in Figure 1 to further test the robustness to noise. This figure shows that the session data is quite noisy, and that the arrival sequence and pattern of sessions belonging to the same usage trend may vary in a way that makes incremental tracking and discovery of the profiles even more challenging than in a batch style approach, where the sessions can be stored in memory, and a standard iterative approach is used to mine the profiles. It also shows how some of the usage trends (e.g: No. 13, 14, 15) are not synchronized with others, and how some of the trends (No. 5, 9, 13, 14) are weak and noisy. Such weak profiles can be even more elusive to discover in a real time web mining system. While Figures 6 and 7 show the high precision and high-coverage B-cell distribution with time, Figure 1 shows the distribution of the input data with time. The fact that all these figures show a striking similarity in the emergence patterns of the trends, attests to the fact that the immune network is able to form a reasonable dynamic synopsis of the usage data, even after a *single pass* over the data, for both types of similarity measures (cosine or *MinPC*). Again, even here, we notice that *MinPC* succeeds slightly better than cosine similarity in learning high-precision and high-coverage profiles. This can be seen for example by the fact that profiles 10 and 19 end up lost with the cosine similarity in Figures 6, because their corresponding learned profiles fall below the precision and coverage threshold.

We notice furthermore that the gap between the *MinPC* and cosine similarities, in the number and fidelity of learned high-precision and high-coverage profiles compared to the incoming stream of evolving trends, gets wider when the trends are presented one at a time (scenarios 1 and 2) as opposed to when they are presented in a more random, alternating order (scenario 3). Note that scenarios 1 and 2 are much more challenging than scenario 3, and they were simulated intentionally to test the ability of TECNO-STREAMS to learn completely new and unseen patterns (usage trends, topics, ...etc), even after settling on a stable set of learned patterns before. In other words, these scenarios represent an extreme test of the adaptability of the single-pass web mining system.

It is interesting to note that the *memory span* of the network is affected by the parameter τ which affects the rate of forgetting in the immune network. A low value will favor faster forgetting, and therefore a more current set of profiles that reflect the most recent activity on a website, while a higher value will tend to keep older profiles in the network for longer periods.

4. CONCLUSION

We investigated using a new robust and scalable algorithm (TECNO-STREAMS) and the effect of similarity for detecting an unknown number of evolving clusters or trends in a noisy Web data stream. The main factor behind the ability of the proposed method to learn in a single pass lies in the richness of the immune network structure that forms a dynamic synopsis of the data. TECNO-STREAMS adheres to all the requirements of clustering data streams [2]: *compactness of representation*, *fast incremental processing of new data points*, and *clear and fast identification of outliers*. This is mainly due to the compression mechanism and the dynamic B-cell model that make the immune network manageable, and continuous learning possible.

Even though the cosine similarity has been prevalent in the majority of web clustering approaches, it may fail to explicitely seek profiles that achieve high coverage and high precision,empsimultaneously. The *Min-Of-Precision-Coverage* or *MinPC* similarity, proposed and investigated in this paper, overcomes these drawbacks. Our simulations confirmed that the *MinPC* similarity does a better job than cosine in learning from a stream of evolving data in a single pass setting, regardless of the order of presentation. This is because the *MinPC* similarity has the advantage of explicitely coupling the precision and coverage criteria to the early learning stages, and furthermore requiring that the affinity of the data to the learned profiles or summaries be defined by the minimum of their coverage or precision, hence requiring that the learned profiles are simultaneously precise and complete, with no compromises.

With an average of *4 milliseconds per user session*, the proposed profile mining system is suitable for use in a real time personalization system to constantly and continuously provide the recommendation engine with a current set of user profiles. The same can be said about the ability to mine evolving topic profiles/summaries from a stream of text data, even in the presence of outliers. In fact detecting potential outliers with TECNO-STREAMS is a trivial process, limited to identifying input data that fail to activate all the B-cells in the immune network, as described in Section **??**.

The logistics of maintaining, caching, or discarding old profiles are much less crucial with our approach than with most existing techniques. Even if discarded, older profiles that re-emerge later, would be re-learned from scratch just like completely new profiles. Like the natural immune system, the strongest advantage of our approach is expected to be its ease of adaptation in dynamic environments such as the World Wide Web. Our approach is modular and generic enough that it can be extended to handle richer Web object models, such as more sophisticated web user profiles and web user sessions, or more elaborate text document representations. The only module to be extended would be the similarity measure that is used to compute the stimulation levels controlling the survival, interaction, and proliferation of the learned B-cell profiles.

5. ACKNOWLEDGMENTS

This work is supported by National Science Foundation CAREER Award IIS-0133948 to O. Nasraoui.

6. REFERENCES

[1] S. Babu and J. Widom. Continuous queries over data streams. In *SIGMOD Record'01*, pages 109–120, 2001.

[2] D. Barbara. Requirements for clustering data streams. *ACM SIGKDD Explorations Newsletter*, 3(2):23–27, 2002.

[3] J. Borges and M. Levene. Data mining of user navigation patterns. In H. A. Abbass, R. A. Sarker, and C. Newton, editors, *Web Usage Analysis and User Profiling, Lecture Notes in Computer Science*, pages 92–111. Springer-Verlag, 1999.

[4] P. Bradley, U. Fayyad, and C. Reina. Scaling clustering algorithms to large databases. In *Proceedings of the 4th international conf. on Knowledge Discovery and Data Mining (KDD98)*, 1998.

[5] Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang. Multi-dimensional regression analysis of time-series data streams. In *2002 Int. Conf. on Very Large Data Bases (VLDB'02)*, Hong Kong, China, 2002.

[6] I. Cohen. *Tending Adam's Garden*. Academic Press, 2000.

[7] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of knowledge and information systems*, 1(1), 1999.

[8] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland Oregon, 1996.

[9] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. In *IEEE Symposium on Foundations of Computer Science (FOCS'00)*, Redondo Beach, CA, 2000.

[10] A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Knowledge Discovery and Data Mining*, pages 58–65, 1998.

[11] J. Hunt and D. Cooke. An adaptative, distributed learning system, based on immune system. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 2494–2499, Los Alamitos, CA, 1995.

[12] N. K. Jerne. The immune system. *Scientific American*, 229(1):52–60, 1973.

[13] R. R. Korfhage. *Information Storage and Retrieval*. Wiley, 1997.

[14] O. Nasraoui, C. Cardona-Uribe, and C. Rojas-Coronel. Tecno-streams: Tracking evolving clusters in noisy data streams with a scalable immune system learning model. In *IEEE International Conference on Data Mining*, Melbourne, Florida, Nov. 2003.

[15] O. Nasraoui, D. Dasgupta, and F. Gonzalez. An artificial immune system approach to robust data mining. In *Genetic and Evolutionary Computation Conference (GECCO) Late breaking papers*, pages 356–363, New York, NY, 2002.

[16] O. Nasraoui, H. Frigui, R. Krishnapuram, and A. Joshi. Mining web access logs using relational competitive fuzzy clustering. In *Eighth International Fuzzy Systems Association Congress*, Hsinchu, Taiwan, Aug. 1999.

[17] O. Nasraoui and R. Krishnapuram. One step evolutionary mining of context sensitive associations and web navigation patterns. In *SIAM conference on Data Mining*, pages 531–547, Arlington, VA, 2002.

[18] O. Nasraoui, R. Krishnapuram, H. Frigui, and A. Joshi. Extracting web user profiles using relational competitive fuzzy clustering. *International Journal of Artificial Intelligence Tools*, 9(4):509–526, 2000.

[19] O. Nasraoui, R. Krishnapuram, and A. Joshi. Mining web access logs using a relational clustering algorithm based on a robust estimator. In *8th International World Wide Web Conference*, pages 40–41, Toronto, Canada, 1999.

[20] M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically synthesizing web pages. In *AAAI 98*, 1998.

[21] C. Shahabi, A. M. Zarkesh, J. Abidi, and V. Shah. Knowledge discovery from users web-page navigation. In *Proceedings of workshop on research issues in Data engineering*, Birmingham, England, 1997.

[22] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):1–12, Jan 2000.

[23] J. Timmis, M. Neal, and J. Hunt. An artificial immune system for data analysis. *Biosystems*, 55(1/3):143–150, 2000.

[24] T. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From user access patterns to dynamic hypertext linking. In *Proceedings of the 5th International World Wide Web conference*, Paris, France, 1996.

[25] H. Yang, S. Parthasarathy, and S. Reddy. On the use of constrained association rules for web mining. In *WebKDD workshop on Knowledge Discovery in the Web*, pages 77–90, Edmonton, Alberta, Canada, 2002.

[26] O. Zaiane, M. Xin, and J. Han. Discovering web access patterns and trends by applying olap and data mining technology on web logs. In *Advances in Digital Libraries*, pages 19–29, Santa Barbara, CA, 1998.

[27] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for large databases. In *ACM SIGMOD International Conference on Management of Data*, pages 103–114, New York, NY, 1996. ACM Press.

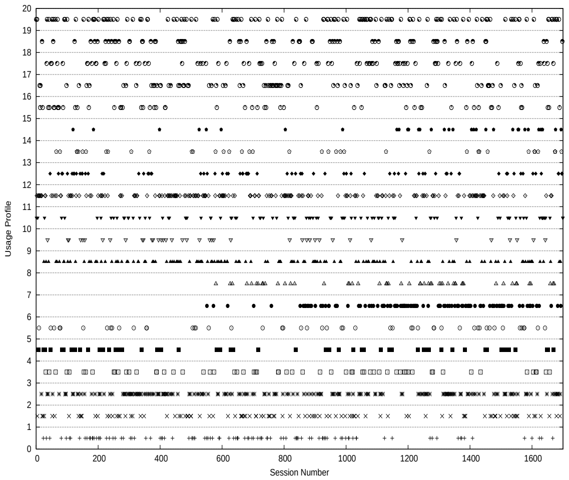
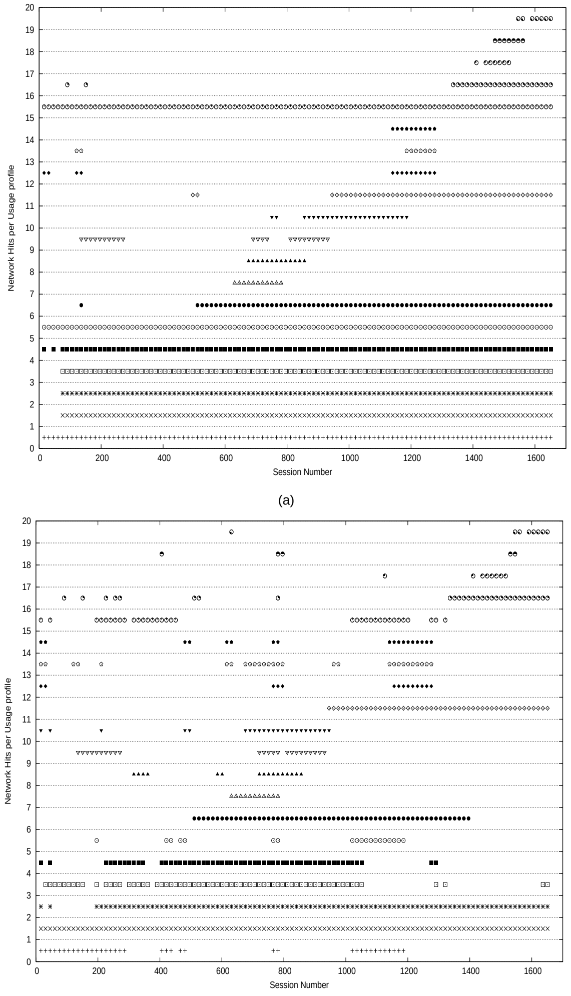


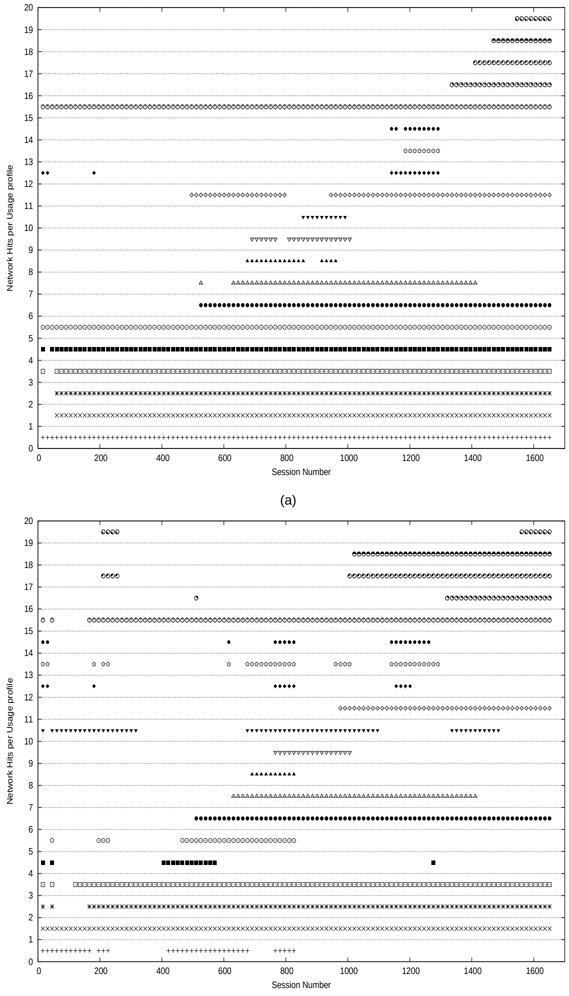
Figure 1: **Distribution of input sessions over usage trend versus session number when only non-noisy ($w_{ij} > 0.6$) sessions are presented in natural chronological order. The horizental axis depicts the session number or a time stamp. The vertical axis is split into several horizental bands, each one depicting one of the 20 usage trends. Trends 5, 9, 13, 14, 15, and 19 appear to be weaker and noisier. Also trends 6 and 7 emerge late in the 12-day access log, while trend 0 weakens in the last days.**



(a)

(b)

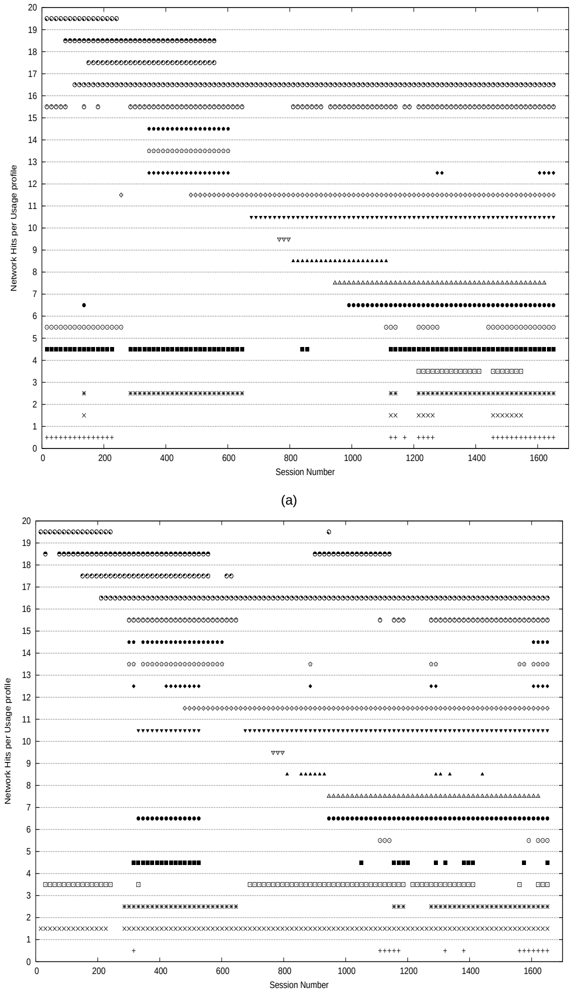
Figure 2: **Hits per usage trend versus session number when sessions are presented in order of trend 0 to trend 19 and cosine similarity is used: (a) Precision ≥ 0.5, (b) Coverage ≥ 0.5**



(a)

(b)

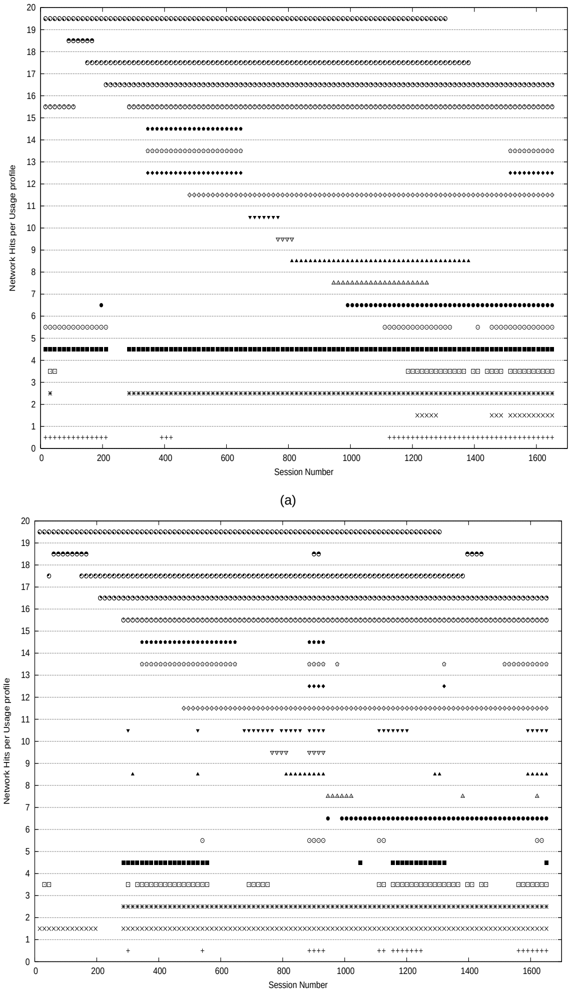
Figure 3: **Hits per usage trend versus session number when sessions are presented in order of trend 0 to trend 19, and *MinPC* similarity is used: (a) Precision ≥ 0.5, (b) Coverage ≥ 0.5**



(a)

(b)

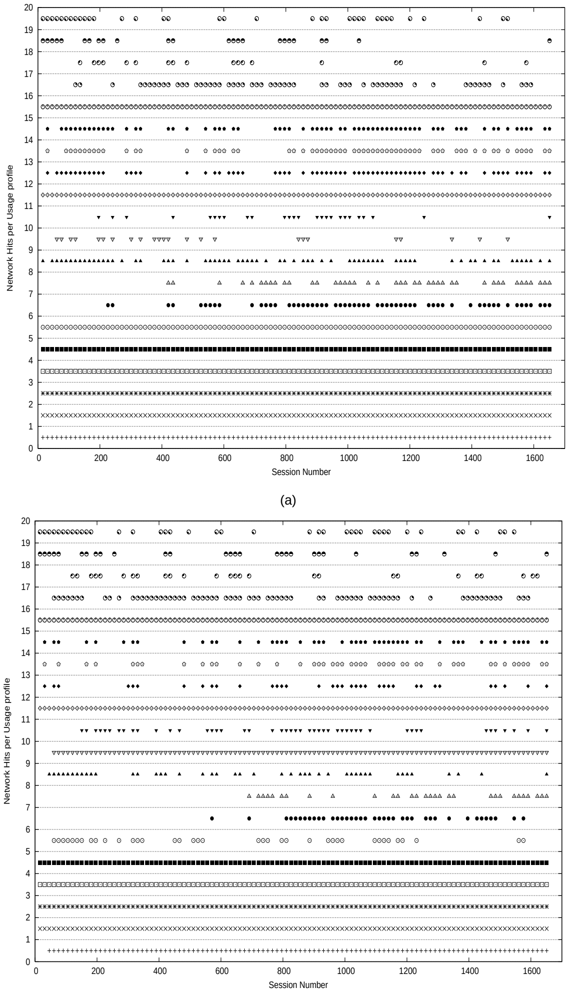
Figure 4: **Hits per usage trend versus session number when sessions are presented in reverse order from trend 19 to trend 0, and cosine similarity is used: (a) Precision ≥ 0.5, (b) Coverage ≥ 0.4**



(a)

(b)

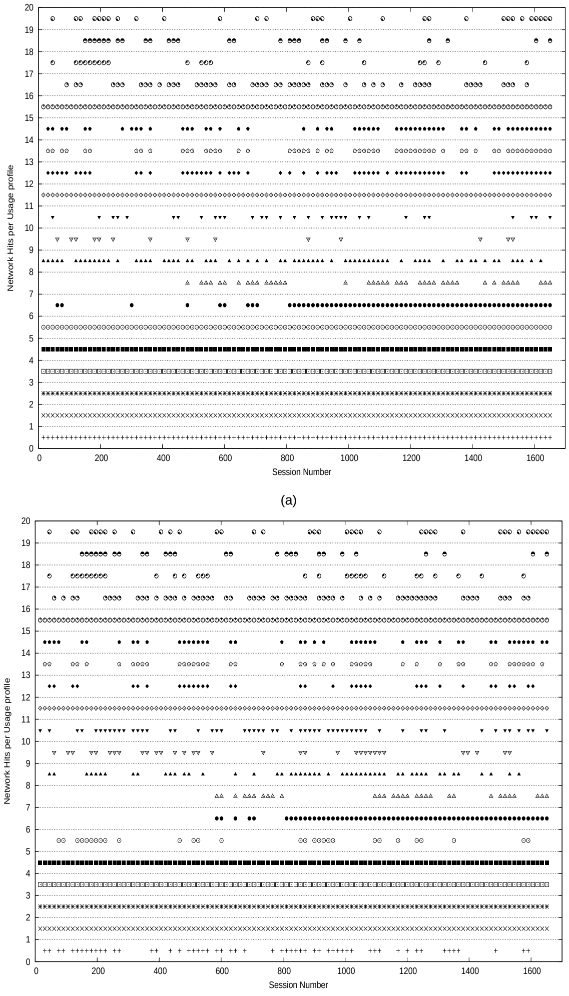
Figure 5: **Hits per usage trend versus session number when sessions are presented in reverse order from trend 19 to trend 0, and *MinPC* similarity is used: (a) Precision ≥ 0.5, (b) Coverage ≥ 0.4**



(a)

(b)

Figure 6: **Hits per usage trend versus session number when all sessions are presented in natural chronological order and cosine similarity is used: (a) Precision ≥ 0.3, (b) Coverage ≥ 0.3**



(a)

(b)

Figure 7: **Hits per usage trend versus session number when all sessions are presented in natural chronological order and *MinPC* similarity is used: (a) Precision ≥ 0.3, (b) Coverage ≥ 0.3**