Performance of Recommendation Systems in Dynamic Streaming Environments[*]

Olfa Nasraoui[†] Jeff Cerwinske[‡] Carlos Rojas[§] Fabio Gonzalez[¶]

Abstract

In this paper, we study the behavior of collaborative filtering based recommendations under evolving user profile scenarios. We propose a systematic validation methodology that allows for simulating various controlled user profile evolution scenarios and validating the studied recommendation strategies. Through the presented work, we observe the effect of the curse of dimensionality *and* sparsity *that can wreck havoc on collaborative filtering in a streaming scenario, and conclude that a* hybrid approach with both content and collaborative filtering *may be the way to go in a high sparsity streaming scenario.*

1 Introduction

Increasing information abundance and overload, as well as awareness of the different tastes of different users has started motivating websites to adapt to different user interests using various techniques that fall within the broad family of Personalization methods. Moreover, the increasing number and diversity of users (most of whom remain anonymous), number of pages/products/items on websites give strong motivation for *automated* personalization techniques.

In order to be automated, Web personalization must be based on some kind of adaptive learning procedure. Guidance or Recommender Systems try to automatically recommend hyperlinks that are deemed to be relevant to the users' interests, in order to facilitate access to the needed information on a website. They are usually implemented on the Web server, and rely on data that reflects the users' interest implicitly (browsing history as recorded in Web server logs) or explicitly (user profile as entered through a registration form or questionnaire). The implicit approach is the focus of the work presented in this paper, because it necessitates little or no input from the user. Also, systems that are implemented on the server side tend to benefit from a global view of all users activities. Recommender systems have already found some success in real e-commerce applications such as Amazon, where they are used to recommend to online shoppers, products and services that they might otherwise never discover on their own. One of the most popular recommendation strategies is *collaborative filtering* [1, 2] which is an instance-based method that recommends items that are relevant to similar users. Typically, this is done by forming a neighborhood of historic user session instances around a new user session, and then recommending pages visited in sessions in this neighborhood using a K-Nearest-Neighbors (K-NN) procedure. As a website receives more and more visits, accumulating and using a large instance base of historic sessions becomes cumbersome, and scalability is compromised. One approach to reduce the size of this instance base has been to replace it by a summarized model, such as by a set of mass user profiles that represent a large number of similar users. This approach typically relies on *Web Usage Mining*, a family of techniques that use data mining in order to extract usage patterns from Web log data [3, 4, 5, 6, 7, 8]. So far, we are aware of no previous work that studied the strengths or limitations of collaborative filtering under the effect of *stream evolution* or change in the Web usage domain. In fact, with the exception of the preliminary results in [9], there has not been any investigation of the performance or even viability of standard recommendation strategies such as collaborative filtering in the demanding environment of evolving data streams.

In this paper, we study the behavior of new and classical approaches to collaborative filtering based recommendations under evolving user profile scenarios, and propose a systematic validation methodology that allows the simulation of various controlled user profile evolution scenarios and evaluation of the studied recommendation strategies.

Our work differs from existing literature in the following aspects: ***(i)*** existing work assumes unrestricted limits on memory (and even computations) for the recommendation stage (for example, k-NN collaborative filtering based recommenders), while the proposed work works under severe memory and computational restrictions that amount to a stream mining framework; ***(ii)*** existing work does not distinguish between different temporal trends, while the proposed work gives the temporal dimension a primordial

---

[*] supported by the National Science Foundation CAREER Award IIS-0133948 to O. Nasraoui

[†] Department of Computer Engineering and Computer Science, Speed School of Engineering, University of Louisville, Louisville, KY 40292.

[‡] Department of Computer Engineering and Computer Science, Speed School of Engineering, University of Louisville, Louisville, KY 40292.

[§] Department of Computer Engineering and Computer Science, Speed School of Engineering, University of Louisville, Louisville, KY 40292.

[¶] Department of Computer and Systems Engineering, University of Colombia, Bogota, Colombia 40292.

stance. Note that because of the special nature of collaborative filtering which makes it imperative to resort to a stream mining framework in the presence of massive usage data, we do not discuss other recommendation families such as content-based filtering, ***(iii)*** So far, we are aware of no other work that studied the strengths or limitations of collaborative filtering under the effect of stream evolution or change in the Web usage domain.

The rest of the paper is organized as follows. In Section 2, we present strategies for implementing a reasonable collaborative filtering recommender system in an evolving data stream scenario, then present a methodology to perform controlled validation in Section 3. In section 4, we present our experimental results, and finally make our conclusions in section 5.

2 Collaborative Filtering in Streaming Scenarios

In a streaming scenario, a recommender system must handle a huge flux of user data under restricted memory and time constraints. Hence, K-Nearest-Neighbor based collaborative filtering must work with limited memory to store the previous instances. TECNO-Streams [10] is a robust stream clustering algorithm that works *in one pass* and under *restricted* space limits, by continuously computing a *limited-size* synopsis of cluster representatives/usage patterns, that can serve as an *evolving* instance base to provide recommendations. We present two adapted recommendation strategies based on K-NN (Algorithm 1) and TECNO-Streams (Algorithm 2); as well as their validation (Algorithm 3).

Algorithm 1 K-NN-Streams Recommender

```
Input: - Current Instance Base Buffer $M_w$ with up to $W$ most recent sessions $S_i$,$i = 1, \cdots, W$; - Number of neighbors ($K$); - maximum number of recommendations ($N$); - Number of sessions received so far: $N_s$; - input subsession: $s_j$: formed by selecting a random subset of $SS$ URLs from a complete new (ground-truth) user session $S_j$, not yet added to $M_w$;
Output: Recommendations: $r_j$
Algorithm:
Set $W_{eff} = Min\{W, N_s\}$;      // as many sessions as available
FOR $i = 1, \cdots, W_{eff}$    Compute similarity $SIM(s_j, S_i)$;      // cosine
Set $K_{eff} = Min\{K, W_{eff}\}$;
Get Neighborhood $\mathcal{N} = \{$Closest $K_{eff}$ historic sessions $S_{(i)}\}$;
Set Recommendations $r_j =$ Top $N$ frequent URLs in $\mathcal{N}$;
```

Algorithm 2 TECNO-Streams Recommender

```
Input: - Current stream synopsis consisting of profiles $P_i$, and their robust variance $\sigma_i$, $i = 1, \cdots, N_{Pmax}$; - maximum number of recommendations ($N$); - input subsession: $s_j$: formed by selecting a random subset of $SS$ URLs from a complete new (ground-truth) real user session $S_j$ that has not yet been presented to learning in TECNO-Streams.
Output: Recommendations: $r_j$
Algorithm:
FOR $i = 1, \cdots, N_{Pmax}$ {
   Compute similarity, $SIM(s_j, P_i)$, and distance,
   $Dist(s_j, P_i) = (1 - SIM(s_j, P_i))$, between $s_j$ and $P_i$;
   Compute robust activation weight $w_{ij} = e^{-\left(\frac{Dist^2(s_j, P_i)}{2\sigma_i^2}\right)}$;
   Accumulate activations of URLs $u$ in profile $P_i$: $w_u = w_u + w_{ij}$;
}
Set Recommendations $r_j =$ Top $N$ URLs with highest activations $w_u$;
```

Algorithm 3 Recommendation and Validation in Streams

```
Input: - Current stream synopsis: For K-NN-Streams, this is the current Instance Base Buffer $M_w$ with up to $W$ most recent sessions $S_i$,$i = 1, \cdots, W$. For TECNO-Streams, this is the summarized profiles $P_i$, $i = 1, \cdots, N_{Pmax}$; - A complete new (ground-truth) real user session $S_j$that has not yet been processed by the Stream synopsis learner or saved to the instance base; - Number of sessions received so far: $N_s$;
Output: - Evaluation metrics for this session, and updated usage synopsis:
   - for k-NN-Streams: New Instance Base Buffer $M_w$;
   - for TECNO-Streams: New stream synopsis consisting of profiles $P_i$, $i = 1, \cdots, N_{Pmax}$;
Algorithm:
FOR Each incomplete session $s_j$, formed by selecting a random subset of $SS$ URLs from a complete new (ground-truth) real user session $S_j$ {
   Apply Streams Recommender in Algorithm 1 or 2;
   Compute evaluation metrics: precision, recall, and $F_1$.
}
For TECNO-Streams: Present complete new (ground-truth) complete user session $S_j$ to TECNO-Streams algorithm (for 1 step) [10];
For K-NN-Streams:
   IF $N_s < W$ THEN Add complete session $S_j$ to Instance Base $M_w$;
   ELSE Replace oldest session from $M_w$ with complete session $S_j$;
```

3 Validation Methodology for Evolving Stream Scenarios

In this section, we describe a validation methodology that is most useful within the framework of mining *evolving* web data streams. If this were a simpler non-dynamic framework, then all that would need to be measured would be the precision and coverage/recall of the learned synopsis as a faithful representation of the input data stream. However, in this case, we have to test an additional feature of learning which is the adaptability in the face of evolution. Therefore we describe three distinct ways to manipulate the pattern of evolution of the input data stream for testing the adaptability of a recommender system in a streaming environment.

3.1 Induced Drastic Sequential User profile Evolution (Scenario D)

Evolution can be simulated easily if an input data set is pre-partitioned into several subsets, one in each known category. The categories can correspond to class labels available with the data, or they can be categories that are computed and validated using an external technique. In the case of web clickstreams or user sessions, the categories have been pre-discovered and validated using a third method that mines user profiles from web user session data. Once the data has been divided into several categories, which are heretoforth called *trends*, different *trend sequencing scenarios* can be formed simply by presenting the data as a stream in the order of the trends. The main idea in the validation procedure is to track the quality of recommendations provided as the input data stream is presented under a given trend sequencing scenario, and as both the instance base for K-NN collaborative filtering and the synopsis in case of TECNO-Streams evolve and keep up with the input user session stream.

3.2 Natural or Mild Chronological Order (Scenario M)

This validation scenario keeps the order of the data instances unchanged. Hence user sessions are seen exactly in the same order as they are received by the web server. This scenario tests for the ability of the system to keep up with the user input streams under mild evolution, since it is conjectured that drastic evolution tends to be rare.

3.3 Instant Validation Metrics

Given a real set of Web user sessions S_j, we form all input subsessions: s_j, formed by selecting subsets of SS URLs from a complete (ground-truth) real user session S_j. These subsessions are used as *new* input sessions to the recommendation strategies based on K-NN-Streams and TECNO-Streams *before* the current session is added to the historic buffer of instances for K-NN, and *before* it is presented as input to TECNO-Streams for learning. After obtaining up to N recommendations as a set r_j, we evaluate the quality of the recommendations by computing the well known Information retrieval measures [11]: *precision*, *recall*, and F_1 measures as follows.

$$Prec(r_j) = \frac{|(r_j - s_j) \cap (S_j - s_j)|}{|(r_j - s_j)|} \tag{3.1}$$

$$Cov(r_j) = \frac{|(r_j - s_j) \cap (S_j - s_j)|}{|(S_j - s_j)|} \tag{3.2}$$

$$F_1 = \frac{2Prec(r_j) \cdot Cov(r_j)}{(Prec(r_j) + Cov(r_j))} \tag{3.3}$$

4 Validation Results

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

Profiles were mined from two benchmark clickstream data sets: the first data set, that we will refer to as (*MediumData*), consists of access to a Computer science and engineering Department website, and the second data set, that we refer to as (*BigData*), consists of accesses to a main university website. After pre-processing *MediumData* as explained in [8], 1,704 sessions (a session consists of consecutive and close requests from the same IP address, differing by no more than 45 minutes) were extracted, accessing a total of 343 URLs. *BigData* on the other hand, generated 29,876 user sessions and 17,665 URLs.

4.1.1 TECNO-Streams Parameter Settings The control parameter for compression [10] was $K = 5$ for *MediumData* and $K = 20$ for *BigData*, and periodical compression every $T = 10$ sessions. The activation threshold was $w_{min} = 0.375$. It is interesting to note that the *memorization span* of the network is affected by the parameter τ which affects the rate of forgetting in the stream synopsis. 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. Another important parameter is the maximum synopsis size of the network (maximum number of nodes) $N_{p_{max}}$ which can be considered as the number of resources available to make up the stream synopsis. A low value will require a stream synopsis of more modest size that can fit in a smaller memory size (hence more useful for stream mining applications), while a higher value will tend to require more memory, and is therefore more costly. Our prior experiments showed that reasonable values are ($N_{p_{max}} = 30$ synopsis cells for *MediumData* and $N_{p_{max}} = 150$ and 500 synopsis cells for *BigData*) and *memorization span* ($\tau = 50$ for *MediumData*, $\tau = 1000$ for *BigData*). Because of space limitations we will only show the results of $N_{p_{max}} = 150$ for *BigData*.

4.1.2 K-NN-Streams Parameter Settings To make a fair comparison, K-NN-Streams Recommender is also limited to work with the same space limitation as TECNO-Streams, i.e., $W = K = N_{p_{max}}$.

4.1.3 Inducing the Drastic and Mild Scenarios for Evolving Stream Validation We illustrate the *continuous* learning ability of the proposed technique using the following simulations:

Scenario D: Ascending Order/Drastic Changes: We partitioned the Web sessions into 20 for *MediumData* (and 93 for *BigData*) distinct sets of sessions, each one assigned to the closest of 20 for *MediumData* (and 93 for *BigData*) profiles previously discovered and validated using Hierarchical Unsupervised Niche Clustering (HUNC) [8]. Then we presented these sessions to TECNO-STREAMS one profile at a time: sessions assigned to profile 0, then sessions assigned to profile 1, $\cdots$, etc. This scenario emphasizes *drastic changes* in user access patterns, where the user activity changes from one category to a different one at certain points in time. Note that HUNC was used because it is an efficient Web Usage Mining technique that produces the optimal number of user profiles automatically. It is also robust to noise and is implemented as part of a full suite that includes an easy validation mechanism. Thus all the discovered profiles were validated against the input sessions before being used to induce the different evolving stream validation scenarios below.

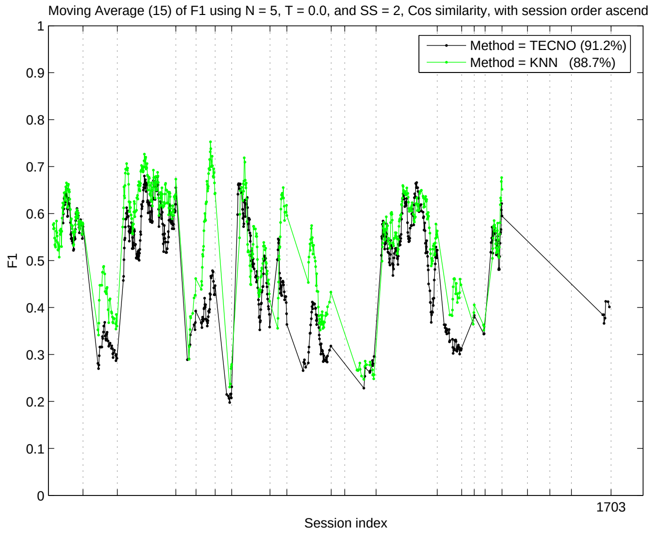
Scenario M: Regular or Natural Order/Mild Changes: The Web sessions are presented in their natural chronological order exactly as received in real time by the web server. This scenario generally results in more continuous and less drastic changes compared to Scenario D, and is therefore termed *mild changes*.

4.2 Effect of Evolution in User Access Patterns under Drastic Changes (Scenario D)

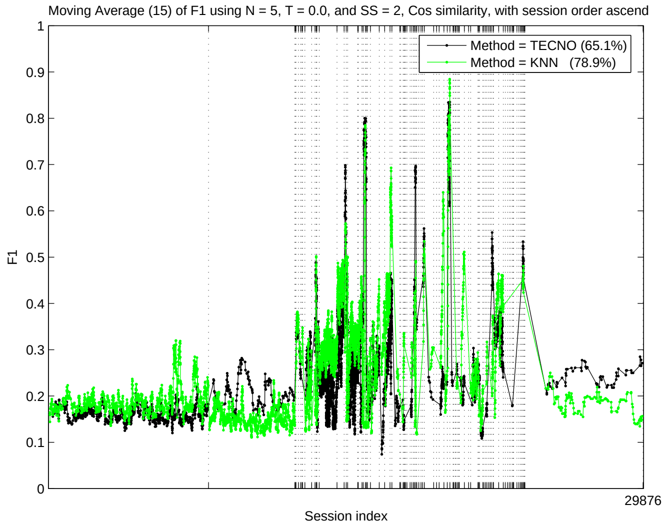
The results are visualized by plotting the validation metrics (such as F_1) versus time or session index, as shown in Figure 1. A moving average with window size of 15 is used to smooth out the noise naturally present in such an experiment with real data, and to be able to observe general trends. We also show the percentage of sessions that succeed to match at least one of the instances (synopsis nodes) in K-NN-Streams (TECNO-Streams) with non-zero similarity in the legend box. This is the proportion of sessions receiving at least one recommendation, i.e., the sessions that end up with a non-zero cosine similarity with at least one of the profiles in the TECNO-Streams synopsis or with at least one of the previous instances in the instance base buffer of K-NN-Streams. To observe space limitations, we will only show the results with $SS = 2$ clicks per subsession and $N = 5$ recommendations. Figure 1 (a) shows the F_1 quality for *MediumData* for TECNO-Streams Recommender and K-NN-Streams Recommender as the user activity changes drastically in 20 consecutive periods from one profile to the next in a total of 20 profiles, with the periods shown separated by dashed vertical lines. As expected, with the drastic fluctuations in user access patterns at the start of each new period, the quality also makes a sharp downward dip for both stream based recommenders since they now must adapt to a completely unseen pattern (also known as a *ramp up*). However, as the activity remains stable within the same period, the recommendations start improving, and generally stabilize, but they may also fluctuate even within the same period, because the sessions do vary even within the same general trend. We notice that the F_1 measure fluctuates, but reaches peaks that are slightly lower (by about 5%) values for TECNO-Streams Recommender as compared to K-NN-Streams Recommender. This behavior may confirm that K-NN-Streams Recommender performs better in periods of maintained stability, which is expected since it does not perform any optimization in summarization of historic data (hence there is no compression mechanism). On the other hand, TECNO-Streams is based on optimizing its summary (hence there is a lossy compression). Figure 1 (b) shows the F_1 quality for *BigData* for $N_{p_{max}} = K = W = 150$ as the user activity changes drastically in 93 consecutive periods from one profile to the next in a total of 93 profiles, showing a similar fluctuation in quality as the environment changes. However, we do notice that in this case, TECNO-Streams Recommender's performance is higher during some of the profiles (e.g. the second and last environments), and that TECNO-Streams Recommender tends to reach higher peaks when the environments change very quickly (observed by short intervals in the figures). We now look at the characteristics of the profiles or environments in this big data to shed some light on possible reasons for the above differences. The first profile is a cluster of more than 8,000 user sessions, the largest majority of which have only one click (the main page), and thus do not contribute to the recommendation experiment. The rest of the sessions in this cluster are all longer noise sessions (i.e. they are different from the majority of other sessions from this and all other clusters) that have been lumped into this cluster simply because they included a click on the main page. Note that we could easily distinguish that these sessions are outliers (in fact HUNC ignores them when it outputs its final profile descriptions). However when we re-partition the input sessions for the purpose of the experiments in this paper, we have intentionally left the outliers in the data, and simply assigned them to the closest cluster. This is because it was important in our simulations to perform experiments that are close to reality and not overly artificial. That is, cleaning all the noise would give a significant advantage to ”any” recommendation strategy, but it would *not* represent a realistic scenario.

Finally, by looking at most of the profiles where the recommendation quality was mediocre, we noticed that they too have similar noisy characteristics as the first profile (i.e. mostly short, hence non-contributing user sessions, with the rest of the sessions, being outliers). Presenting such sessions in succession gives no learning opportunity to either of the recommender systems, because there is no consistent pattern of access in them, and hence appearing always as if they were *brand new* sessions. This is considered one of the weaknesses of collaborative filtering (*ramp up or cold start*), leading us to believe more strongly that a *hybrid* approach merging *both content-based filtering with collaborative filtering* would likely prove essential in *quickly* evolving *noisy* clickstreams, because content-based filtering would be the only strategy that can adapt to new inputs not yet represented in the historic instance base or learned stream synopsis.

An even more interesting observation is that for the same environments (e.g. first two environments) that are plagued by short non-informative sessions and many longer outliers, KNN-Streams Recommender outperforms TECNO-Streams Recommender. This is because TECNO-Streams is the technique that relies more on ”active learning” and most of all relies on *“robust learning”* that was designed from the start to be resistant to noise, i.e. to learn a stream synopsis that is *not derailed by outliers*. On the other hand, KNN performs a lazy learning that does not distinguish outliers from actual core sessions, and hence accumulates these outliers in its instance base without any discrimination, thus using *even outliers* for prediction when needed. This is an ironic finding, because in a sense, the strength of a robust method may become a caveat in a *streaming* recommendation scenario with *fast evolution*.



(a)



(b)

Figure 1. F_1 quality of recommendations with TECNO-Streams and K-NN Streams on new sessions from *MediumData* (top) and *BigData* (bottom: TECNO-Streams with $N_{Pmax} = 150$ and K-NN Streams with $K = W = 150$), shown versus session number (t) of the input data stream when sessions are presented in *ascending order: drastic changes* (Scenario D)

4.3 Effect of Evolution in User Access Patterns under Mild Changes (Scenario M)

Figure 2 (a) for *MediumData* shows that as expected, with the daily fluctuations in user access patterns presented in their natural order, the F_1 also fluctuates and ranges in [0.3, 0.5] for TECNO-Streams Recommender versus [0.2, 0.5] for K-NN-Streams Recommender. Hence, in the case of mild everyday evolution of user activity, TECNO-Streams Recommender has better performance, in the sense that, when the environment changes, the quality does not deteriorate as much as K-NN-Streams Recommender, in fact, the F_1 drops but to a value that is approximately 10 to 20% higher than K-NN-Streams Recommender. Figure 2 (b) shows the F_1 quality for *BigData* for $N_{p_{max}} = K = W = 150$ as the user activity changes in its natural (chronological) order, showing that contrary to *MediumData*, both recommenders seem to struggle to barely reach an F_1 value close to 0.2. We note that the big data set contains a diverse and huge number of accesses comprising a very large number of rather very small and very diverse profiles. Several profiles had as little as 30 sessions compared to others containing more than 8,000 sessions, with mostly very shallow visits and a large number of outliers. The user sessions targeted a large number (17,600) of diverse pages on a main university website, many of which having no consistent topicality. This included, for instance, pages from many unrelated areas or disciplines, pages about academics, pages about the University's music band, the weather, sports or financial aid, and a large number of pages belonging to students posting content that derived from diverse personal interests (see description of some of these profiles in [8]). To cite just a few examples, the latter profiles ranged from the music group *Nirvana* to *Persian poetry*, and from *Antonio Banderas* to *funny jokes* and to the *Euler number "e"*, most of which tending to attract a relatively small number of visits compared to the overall massive data. We conjecture that the big data set exhibits many of the signs of the *curse of dimensionality* and *sparsity* (*BigData* has over 17,600 URLs, roughly *50 times* the dimensionality of *MediumData*) combined together in such a massive and diverse repertoire of profiles. Furthermore the mild scenario's challenge is that sessions from *all 93* different trends are presented close together in time and not separately as in scenario D.

5 Conclusions

In this paper, we studied the behavior of collaborative filtering based recommendations under evolving user profile scenarios using a systematic validation methodology that allowed the simulation of various controlled user profile evolution scenarios for data streams formed from clickstreams of varying sizes, dimensionalities, and *thus sparsity*. We have observed the effect of the *curse of dimensionality* and *sparsity* that tended to wreck havoc on collaborative filtering, and concluded that a *hybrid approach with both content* *and collaborative filtering* may be the way to go in a high sparsity streaming scenario. Finally, and ironically, we have noticed how the presence of outliers is handled differently by the two techniques, and how the same strength (robustness to noise) of a technique may occasionally turn into a caveat in a streaming scenario. Yet, we will not conclude prematurely that *robustness* to noise is a disadvantage. *On the contrary, this may suggest a need for even more complex validation scenarios, as well as the need for a hybrid recommendation strategy that draws on multiple recommenders' strengths while circumventing their respective weaknesses*. To summarize our results, for a website with a modest number of profiles, K-NN-Streams Recommender performs well when the user activity is more stable, and can track stable activity accurately immediately after a drastic change (Scenario D); while TECNO-Streams performs better in natural access patterns which change less drastically, but more often since all profiles are active simultaneously (Scenario M). Both recommenders struggle with a high number of profiles, high number of items/URLs, and sparsity.

References

[1] J.A. Konstan, B. Miller J. Maltz, G. Herlocker, and J. Riedl, "Grouplens: Collaborative filtering for usenet news," in *Communications of the ACM, March, p. 77-87*, 1997.

[2] J.B. Schafer, J. Konstan, and J. Reidel, "Recommender systems in e-commerce," in *Proc. ACM Conf. E-commerce*, 1999.

[3] 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*, Santa Barbara, CA, 1998, pp. 19–29.

[4] R. Cooley, B. Mobasher, and J. Srivastava, "Data preparation for mining world wide web browsing patterns," *Journal of knowledge and information systems*, vol. 1, no. 1, 1999.

[5] M. Spiliopoulou and L. C. Faulstich, "Wum:a web utilization miner," in *Proc. EDBT workshop WebDB98*, Valencia, Spain, 1999.

[6] 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*, Toronto, Canada, 1999, pp. 40–41.

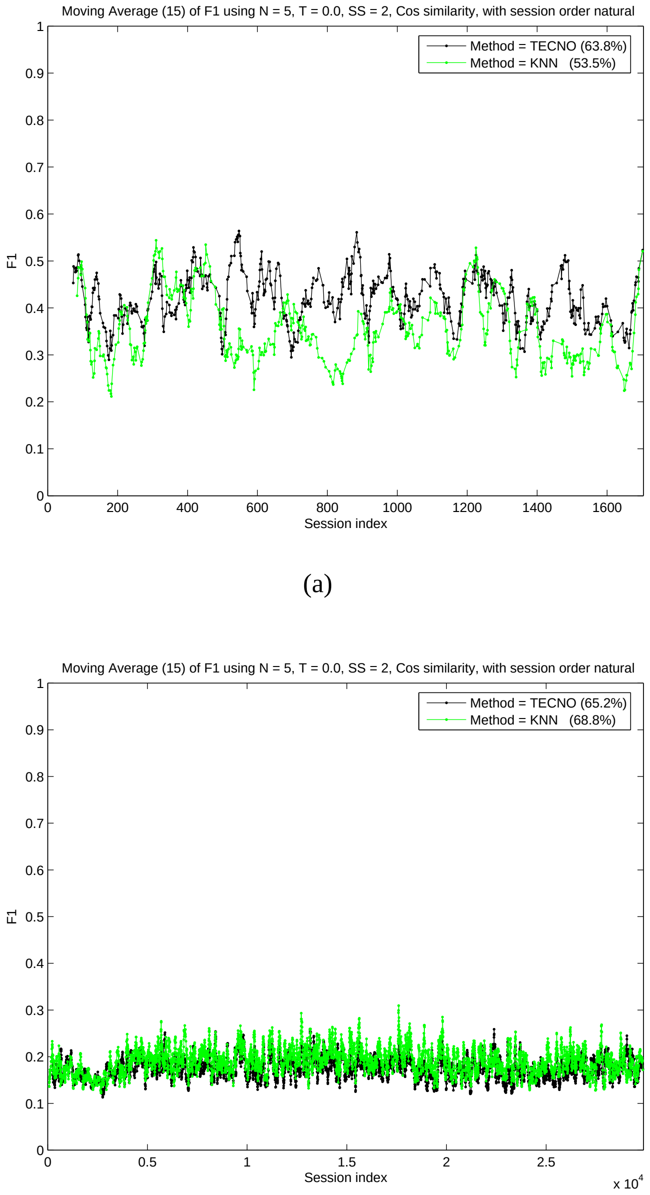
[7] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan, "Web usage mining: Discovery and applications of usage patterns from web data," *SIGKDD Explorations*, vol. 1, no. 2, pp. 1–12, Jan 2000.

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

[9] O. Nasraoui, J. Cerwinske, C. Rojas, and F. Gonzalez, "Collaborative filtering in dynamic usage environments," in *Proc. 15th ACM International Conf. on Information and Knowledge Management (CIKM 2006)*, Arlington, VA, 2006, pp. 794–795.

[10] O. Nasraoui, C. Cardona, and C. Rojas, "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.

[11] C.J. VanRijsbergen, *Information Retrieval*, Buttersworth, London, 1989.



(a)

(b)

Figure 2. F_1 quality of recommendations with TECNO-Streams and K-NN Streams on new sessions from *MediumData* (top) and *BigData* (bottom: TECNO-Streams with $N_{Pmax} = 150$ and K-NN Streams with $K = W = 150$), shown versus session number (t) of the input data stream when sessions are presented in *natural order: mild changes* (scenario M)