Collaborative Filtering in Dynamic Streaming Environments

Olfa Nasraoui, Jeff Cerwinske, Carlos Rojas
Dept. Computer Sciences & Engineering
University of Louisville
olfa.nasraoui@louisville.edu

Fabio Gonzalez
Dept. Computer Engineering & Systems
Universidad Nacional de Colombia
fagonzalezo@unal.edu.co

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Information Filtering; H.5.4 [Hypertext/Hypermedia]: User issues

General Terms: Algorithms, Performance, Reliability.

Keywords: Collaborative filtering, mining evolving streams, recommendation systems, evaluation.

1. INTRODUCTION

The increasing expansion of websites and their web usage necessitates increasingly scalable techniques for Web usage mining that can be better cast within the framework of *mining evolving data streams* [1, 5]. Despite recent developments in mining evolving Web clickstreams [3, 6], there has not been any investigation of the performance of collaborative filtering [2] in the demanding environment of evolving data streams. In this paper, we study *limited memory* collaborative filtering based recommendations in evolving scenarios using a systematic validation methodology.

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 $\mathcal{N}$;
```

2. COLLABORATIVE FILTERING IN STREAMING SCENARIOS

In a streaming scenarios, 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 [3] 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 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**

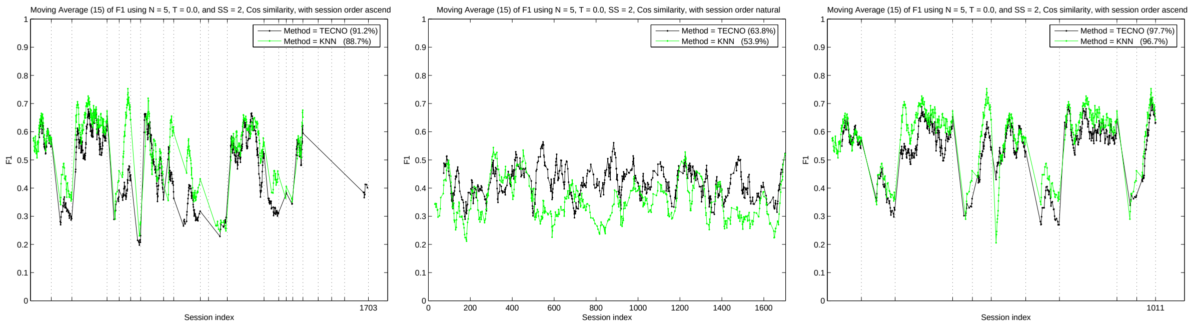
```
Input: - Current stream synopsis: For K-NN-Streams, this is the current Instance Base Buffer $M_w$ with up to $W$ most recent sessions $S_i$,$i = 1, \cdots, W$. For TECNO-Streams, this is the summarized profiles $P_i$, $i = 1, \cdots, N_{Pmax}$; - A complete new (ground-truth) real user session $S_j$that 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) [3];
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 RESULTS IN EVOLVING SCENARIOS

20 User profiles were mined from a benchmark clickstream data set consisting of 12 day accesses to the CECS Department website of the University of Missouri-Columbia. After pre-processing as explained in [4], 1,704 sessions[1] were extracted accessing a total of 343 URLs. For TECNO-Streams,

CIKM'06, November 5–11, 2006, Arlington, Virginia, USA.
ACM 1-59593-433-2/06/0011.

9[1] a session consists of consecutive and close requests from the same IP address, differing by no more than 45 minutes



(a) (b) (c)

Figure 1: F_1 **quality of recommendations on new sessions versus session number** (t) **of the input data stream when sessions are presented in (a)** *ascending order: drastic changes* **(Scenario D), (b)** *natural order: mild changes* **(scenario M), (c)** *drastic changes with repeats* **(scenario R). Due to space limitations, we show the results only for** $SS = 2$ **clicks per subsession and** $N = 5$ **recommendations. The legend box shows the proportion of sessions receiving** *at least one recommendation*, **i.e. with non-zero similarity to at least one of the instances in the memory buffer or synopsis.**

the control parameters for compression [3] were $K = 5$, and periodical compression every $T = 10$ sessions. The activation threshold was $w_{min} = 0.375$. The parameter τ which affects the rate of forgetting in the stream synopsis is set to 50. The memory restriction which is quantified by the maximum synopsis size of the network (maximum number of nodes), $N_{p_{max}}$, is set to 30. To make a fair comparison, K-NN-Streams Recommender is also limited to work with the same space limitation as TECNO-Streams, i.e., $W = K = 30$. The results are visualized by plotting the validation metrics versus time/session index with a moving average of window size 15.

Effect of Evolution in User Access Patterns under Drastic Changes (Scenario D): We partitioned the input Web sessions into 20 clusters centered at 20 profiles previously discovered and validated using Hierarchical Unsupervised Niche Clustering (HUNC)[2] [4]. Then we presented these sessions one profile at a time: sessions from profile 0, then sessions from profile 1, $\cdots$, etc. Figure 1 (a) shows the F_1 quality of recommendations over 20 consecutive periods (profiles), separated by dashed vertical lines. With the drastic fluctuations in user access patterns at the start of each new period, F_1 makes a sharp dip for both stream based recommenders since they now must adapt to a completely unseen usage pattern (*ramp up*), with the F_1 reaching peaks slightly higher for K-NN-Streams Recommender, which performs better in periods of maintained stability since it does not perform any optimization in summarizing its instances (lossless compression), while TECNO-Streams optimizes its summary (lossy compression).

Effect of Evolution in User Access Patterns under Mild Changes (Scenario M): We presented the Web user sessions in their natural (*chronological*) order as received by the Web server. Figure 1 (b) shows that with *natural* fluctuations in user access patterns (over 12 days), the F_1 measure fluctuates, but remains 10-30% higher for TECNO-Streams Recommender compared to K-NN-Streams Recommender.

Effect of Evolution in User Access Patterns under Drastic Changes with Repeated Profiles (Scenario R): We selected the first 5 profiles, and presented the sessions of the stream in the order of the profiles as in Scenario D. However, as soon as the 5 profiles were unraveled, we repeated the presentation of the 5 profiles again in the same order. Figure 1 (c) shows the quality of recommendations as the user activity changes drastically in 10 consecutive periods *profile 1,2,3,4,5,1,2,3,4,5*). Compared to the *Drastic scenario* above, we notice a significant difference in the start of the new period for profile 1 when it is repeated the second time (start of 6th period in the figures), with TECNO-Streams Recommendations maintaining a much higher value. Due to its immune based learning (basis for vaccination), TECNO-Streams recommender tends to improve the *second time* it re-encounters a profile (e.g. compare its F_1 value during profile 3 presented first in period 3 and then repeated again in period 8), while K-NN-Streams Recommender simply repeats the same deterministic rote memorization.

4. CONCLUSIONS

We have evaluated the behavior of collaborative filtering recommendations under evolving usage scenarios, showing that K-NN-Streams Recommender performs well when the user activity alternates between different trends, and the activity within each trend is more *stable*, while TECNO-Streams performs better in *naturally changing real* user access patterns, and degrades gracefully when *repeating* drastic changes occur.

Acknowledgment: This work is supported by National Science Foundation CAREER Award IIS-0133948

5. REFERENCES

[1] C. Aggarwal, J. Han, J. Wang, and P. Yu. A framework for clustering evolving data streams. *Proc. VLDB conference*, 2003.

[2] J. Konstan, B. M. J. Maltz, G. Herlocker, and J. Riedl. Grouplens: Collaborative filtering for usenet news. In *Communications of the ACM, March, p. 77-87*, 1997.

[3] 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 Conf. on Data Mining*, 2003.

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

[5] O. Nasraoui and C. Rojas. Robust clustering for tracking noisy evolving data streams. In *Proc. 2006 SIAM Conf. on Data Mining (SDM 2006)*, pages 80–99, Bethesda, MD, 2006.

[6] O. Nasraoui, C. Rojas, and C. Cardona. Using retrieval measures to assess similarity in mining dynamic web clickstreams. In *Proc. ACM KDD: Knowledge Discovery and Data Mining Conference*, pages 439–448, Chicago, IL, Aug. 2005.

9[2]We used HUNC [4] because it is an efficient Web Usage Mining technique that produces the optimal number of user profiles automatically, is robust to noise, and also produces an easy validation mechanism that validates all discovered profiles against the input data