

# Personalised Fading for Stream Data

Bruno Veloso  
University of Vigo, Spain  
INESC TEC, Porto, Portugal  
bmdvo@isep.ipp.pt

Juan Carlos Burguillo  
University of Vigo, Spain  
J.C.Burguillo@det.uvigo.es

Benedita Malheiro  
ISEP/Polytechnic of Porto  
INESC TEC, Porto, Portugal  
mbm@isep.ipp.pt

Jeremy Foss  
Birmingham City University  
Jeremy.Foss@bcu.ac.uk

## ABSTRACT

This paper describes a forgetting technique for the live update of viewer profiles based on individual sliding windows, fading and incremental matrix factorization. The individual sliding window maintains, for each viewer, a queue holding the last  $n$  viewer ratings. As new viewer events occur, they are inserted in the viewer queue, by shifting and fading the queue ratings, and the viewer latent model is faded. We explored time, rating-and-position and popularity-based fading techniques, using the latter as the base fading algorithm. This approach attempts to address the problem of dynamic viewer profile updating (volatile preferences) as well as the problem of bounded processing resources (fixed size queues). The results show that our approach outperforms previous approaches, improving the quality of the predictions.

## CCS Concepts

•Information systems → Information retrieval;

## Keywords

Forgetting Technique; Stream Mining; Fading Strategies.

## 1. INTRODUCTION

The number of media sources, resources and viewer feedback data available – ratings, likes, posts or reviews – makes real time processing impossible for humans and standard applications. The viewer generated data, which corresponds to explicit viewer preferences and intrinsic behaviours, can be used for defining the viewer profiles. In particular, dynamic viewer profiling, *i.e.* the ability to build and update profiles based on the continuous stream of viewer interactions can be addressed as stream mining. Our approach involves four phases: (i) data set selection; (ii) data set modelling; (iii)

model update; and (iv) model validation. First, we chose, prepared and partitioned the MovieLens 100k (ML100k) and the MovieLens 1M (ML1M) for the application of Stream Mining techniques. Next, we created the initial model and selected the hyper-parameters, using Singular Value Decomposition (SVD) and Stochastic Gradient Descent (SGD), respectively. Then, every time a viewer rated a resource, we updated the viewer's latent matrix, using SGD and different fading strategies. Finally, we validated this methodology by calculating the Root Mean Square Error (RMSE) and Recall between the predicted and the actual viewer ratings.

We propose a personalised rating fading technique using an incremental matrix factorization algorithm designed for streaming data. Off-line, we create the initial model, by determining the optimal learning rate and over-fitting parameters, and a fixed-size first in, first out (FIFO) queue for each viewer, containing the last  $n$  viewer ratings. On-line, for every new rating event, first we calculate the viewer faded predictions and, then, update the model, using SGD to recalculate the viewer latent matrix. The calculation of the viewer faded predictions involves creating a temporary copy of the latent item matrix, fading the viewer queue ratings and copying the latent item matrix and, finally, multiplying the latent viewer row by the faded copy of the latent item matrix. We explored with linear and staircase time fading, which are based on the epoch of the viewer rating, and with sorted and unsorted rating-and-position fading, which are based on the rating and on the position of the rating in the viewer rating queue. Our methodology, when compared with the pre-existing approaches referred in Section 2, provides predictions with increased accuracy and lower errors. This paper is organised in five sections, addressing stream profiling, the proposed forgetting techniques, the experiments and results and, finally, the conclusions.

## 2. RELATED WORK

Live viewer profiling is typically performed using sliding windows ([3, 5]) or factor fading ([4]) together with incremental matrix factorization. In particular, sliding windows are FIFO queues of dynamic or fixed size representing a number of viewer events [3] or a time span [5].

Matuszyk and Spiliopoulou (2014) [3] use two sliding window: (i) a fixed size event-based individual queues; and (ii) a dynamic size time-based individual queues. In both cases

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'17, April 3-7, 2017, Marrakesh, Morocco

Copyright 2017 ACM 978-1-4503-4486-9/17/04...\$15.00

<http://dx.doi.org/10.1145/3019612.3019868>

the sliding window discards the oldest ratings and the latent matrices remain unchanged. Siddiqui, Tiakas, Symeonidis, Spiliopoulou and Manolopoulos (2014) [5] propose dynamic size time-based individual queues and keep the latent matrices unchanged. We not only experiment with time and rating-and-position based forgetting techniques, using individual fixed-size rating queues, but, additionally, apply fading both to the viewer model and to the viewer rating queue. Our results and those of Matuszyk (2015) [4] show that fading outperforms the sliding window results.

Matuszyk, Vinagre, Spiliopoulou, Jorge and Gama (2015) [4] propose different forgetting techniques for incremental matrix factorization, using global and positive-only feedback, together with the latent matrices update. The forget unpopular items technique, which penalises the items with lower number of ratings, showed the best results. Our approach adopts a fixed-size rating queue and updates the model of each viewer, *i.e.*, every time a new rating is inserted in the viewer queue, the queue and the viewer model are faded.

### 3. FORGETTING TECHNIQUE

This paper proposes four individual forgetting strategies supported by individual fixed-size FIFO queues for viewer profiling: two time-based functions – with linear temporal fading (LTF) and staircase temporal fading (STF) – and two rating-and-position-based functions – the unsorted rating-and-position fading (RPF) and the sorted rating-and-position fading (SRPF) – which take into account the viewer ratings and the position of the item in the queue. Our algorithm, which extends the biased incremental algorithm used by Takács *et al.* (2009) [6], is organised in three main steps: (i) creation of the initial off-line model; (ii) the optimisation of the hyper parameters (learning rate and over-fitting); and (iii) the on-line model update with stream learning.

Algorithm 1 illustrates the on-line stream learning, which can use any of the implemented fading strategies. This approach uses the initial off-line cross-validation data, the streamed data and the individual hyper parameters to update the model. This stream learning algorithm is used both

---

#### Algorithm 1 Stream Learning

---

```

1: for  $rc_{u,i} \leftarrow \text{CrossValidation and DataStream do}$ 
2:    $\text{ViewerQueue}_u \leftarrow (rc_{u,i}, i)$ 
3:   for  $i = 0; i < \text{ViewerQueue.size}; i++$  do
4:      $f_{u,i} = \text{Equation (1), (2), (3) or (4)}$ 
5:      $\hat{r}_{u,i} \approx \mu + \text{bias}_u + \text{bias}_i + f_{u,i} \cdot q_i^* p_u$ 
6:     if  $\text{fading} == \text{SRPF}$  then
7:        $\text{ViewerQueue.sort}()$ 
8:      $e_{u,i} = r_{u,i} - \hat{r}_{u,i}$ 
9:      $f_{u,i} = \text{Equation (1), (2), (3) or (4)}$ 
10:     $p_u \leftarrow p_u + \gamma(e_{u,i} q_i f_{u,i} - \lambda p_u)$ 
```

---

with cross-validation data (off-line) and stream data (on-line). For each new rating, it adds the rating and the item id to the individual viewer queue (line 2), fades all ratings in the viewer queue (line 3-5), sorts the viewer queue if the forgetting strategy selected is SRPF, calculates the error between the predicted and the real rating (line 8), determines the current viewer fading value and, finally, updates

the model using the global hyper parameters, the fading value and the rating error (line 9-10).

**Linear / Staircase Temporal Fading** strategy fades the ratings according to the difference between the timestamps of the rating being faded and the last viewer rating. Equation 1 displays the corresponding formula where  $f_{u,i}$  is the fading value of user  $u$  for item  $i$ ,  $Ts_n$  is the timestamp of the last (most recent) item,  $Ts_i$  is the timestamp of the item in the position  $i$  of the queue and  $\alpha$  is a multiplying factor. All timestamps are in seconds (s) and for the LTF  $\alpha = 1.0$ . The STF strategy defines a staircase of fading levels as the elapsed time increases. This fading function applies a multiplying factor ( $\alpha = 0.05$ ) for each six month period ( $Ts_n = 15\,552\,000$  s) since the last (most recent) viewer rating.

$$f_{u,i} = \alpha \frac{T s_n - T s_i}{T s_n} \quad (1)$$

**Rating and Position Fading** strategy is based on the viewer rating and on the position of the item in the viewer queue, using a linear combination factor  $\beta$ , according to Equation 2, where  $r_{u, inorm}$  is the normalised rating of user  $u$  for item  $i$ ,  $Position_i$  is the position of the rating in the viewer queue and  $QS_u$  is the dimension of the queue.

$$f_{u,i} = \beta r_{u, inorm} + (1 - \beta) \left| \frac{Position_i}{QS_u} - 1 \right| \quad (2)$$

**Sorted Rating and Position Fading** strategy is a variant of the antecedent approach where the elements of viewer rating queue are sorted by rating values, keeping the top ratings at the head of the queue. After calculating the fading value using Equation 2, the viewer queue is sorted.

**Unpopular Fading** strategy, which was initially proposed by Matuszyk (2015) [4], favours the most frequently watched items. We used this technique as the base algorithm.

### 4. EXPERIMENTS AND RESULTS

Our proposal was evaluated with MovieLens 100k (ML100k) and MovieLens 1M. In terms of evaluation metrics, we used the Root Mean Square (RMSE), as defined by Takács *et al.* (2009) [6], to determine the prediction errors, the Recall@N, as proposed by Cremonesi *et al.* (2010) [1], and TRecall@N, a new metric we designed, to calculate the prediction accuracy. Our recall based metric – Target Recall (TRecall) – was used in the case of global feedback, *i.e.*, we use all rated items instead of just the top-rated items [6, 1]. TRecall evaluates the accuracy of the predictions using all viewer ratings. For each new viewer rating event, we determine the TRecall@N. The difference between TRecall@N and Recall@N is that we verify if the newly rated item belongs to the list of the top  $N$  viewer predicted items centred around the target viewer rating rather than below. The adopted evaluation method was inspired by the prequential evaluation proposed by Gama *et al.* (2009) [2]. The predictions are evaluated using RMSE, TRecall@N and Recall@N metrics with  $N = 10$ . We made two sets of experiments to:

**Analyse the Impact of Viewer Queue Size** The results show that, in the case of ML100k with global feedback, the

size of the viewer queue affects  $TRecall@10$ . In particular,  $TRecall@10$  increases until the queue size reaches 20 and, then, remains stationary as the queue size increases. The prediction errors remain unchanged. Using positive feedback, the recall decreases as the size of the queue increases since the number of top-rated items per viewer in this data set is low. In the case of ML1M with global feedback, both the prediction errors and the  $TRecall@10$ , first, improve and, after, remain unchanged as the viewer queue size increases. The reduction of the prediction errors and the improvement of  $TRecall@10$  indicate that the average number of ratings per viewer is higher in this data set than in the ML100k. With positive feedback, the best recall occurs with a queue size of five and decreases for larger queues (the number of top-rated items per viewer is small).

**Evaluate the Fading Strategies** Table 1 displays the normalised results with positive feedback ( $Recall@10$ ) and global feedback ( $RMSE$  and  $TRecall@10$ ). We used the UF strategy as base algorithm since it provided, according to Matuszyk *et al.* (2015) [4], the best results. Since our cross-validation and stream data have a minimum of 18 ratings per viewer, the tests were performed with a queue size (QS) of 10 to ensure forgetting would occur to all viewers.

**Table 1: Fading Strategies with QS=10 and  $\beta = 0.5$**

	ML100k			ML1M		
	Global		Positive	Global		Positive
	RMSE	TRecall	Recall	RMSE	TRecall	Recall
LTF	<b>0.180</b>	<b>0.068</b>	<b>0.251</b>	0.175	0.011	0.156
STF	0.180	0.064	0.246	0.175	0.010	0.155
RPF	0.180	0.055	0.236	0.178	0.011	0.311
SRPF	0.191	0.041	0.232	<b>0.178</b>	<b>0.016</b>	<b>0.407</b>
UF	0.208	0.039	0.207	0.237	0.008	0.125

The results for ML100k, which has a time span of eight months, show that our strategies are better than the base algorithm both in terms of predictive accuracy and classification accuracy metrics. The best results were obtained with linear time fading with a reduction of 13.4% of the prediction errors. In terms of classification accuracy,  $TRecall@10$  improved 42.6% in the case of global feedback and, in the case of positive feedback,  $Recall@10$  improved 17.5%. The results with ML1M, which has a time span of 33 months, continue to outperform those of the base algorithm. The best results were obtained with the sorted rating-and-position fading, reducing by 24.9% the prediction errors. In the case of global feedback,  $TRecall@10$  increased 50.0% and, in the case of positive feedback,  $Recall@10$  increased 68.8%.

## 5. CONCLUSIONS

This paper describes four fading strategies for on-line viewer profiling supported by individual fixed-size FIFO queues and incremental matrix factorisation. In terms of the viewer queue size, we concluded from the diverse optimal queue sizes obtained with global and positive feedback for the two data sets that the queue size depends generically on the number of relevant items. In the case of global feedback, since the number of rated items in ML1M is approximately ten-fold those of ML100k, the best results were obtained with queue sizes of 20 and 50 for ML100k and ML1M, respectively. In the case of positive feedback, since the number

of top-rated items is reduced in comparison with the number of rated items, the best results were achieved for both data sets with a queue size of five. The results show that our strategies outperform the UF strategy proposed by Matuszyk *et al.* (2015) [4]. LTF was the best fading strategy for ML100k and SRPF the best fading strategy for ML1M, *i.e.*, the rating-and-position fading strategy provides better results than the time-based fading strategy when the time span of the data stream increases.

Based on the outcomes of this work regarding the application of the time and rating-and-position fading strategies to ML100k and ML1M, we are currently experimenting with rating-and-time based fading strategies. Additionally, we intend to improve the accuracy of the overall algorithm by also updating the item latent matrix.

## 6. ACKNOWLEDGMENTS

This work was partially financed by the European Regional Development Fund (ERDF) through the Operational Programme for Competitiveness and Internationalisation (COMPETE Programme), within project «FCOMP-01-0202-FEDER-023151» and project «POCI-01-0145-FEDER-006961», and by national funds through the Fundação para a Ciência e Tecnologia (FCT) - Portuguese Foundation for Science and Technology - as part of project UID/EEA/50014/2013.

## 7. REFERENCES

- [1] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM, 2010.
- [2] J. Gama, R. Sebastião, and P. P. Rodrigues. Issues in evaluation of stream learning algorithms. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 329–338. ACM, 2009.
- [3] P. Matuszyk and M. Spiliopoulou. *Selective Forgetting for Incremental Matrix Factorization in Recommender Systems*, pages 204–215. Springer International Publishing, Cham, 2014.
- [4] P. Matuszyk, J. Vinagre, M. Spiliopoulou, A. M. Jorge, and J. Gama. Forgetting methods for incremental matrix factorization in recommender systems. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC ’15, pages 947–953, New York, NY, USA, 2015. ACM.
- [5] Z. F. Siddiqui, E. Tiakas, P. Symeonidis, M. Spiliopoulou, and Y. Manolopoulos. xstreams: Recommending items to users with time-evolving preferences. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, WIMS ’14, pages 22:1–22:12, New York, NY, USA, 2014. ACM.
- [6] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Scalable collaborative filtering approaches for large recommender systems. *J. Mach. Learn. Res.*, 10:623–656, June 2009.