Adaptive collaborative topic modeling for online recommendation

Marie Al-Ghossein, Pierre-Alexandre Murena, Talel Abdessalem, Anthony Barré, Antoine Cornuéjols

Adaptive Collaborative Topic Modeling for Online Recommendation

Marie Al-Ghossein
LTCI, Télécom ParisTech
Paris, France
malghossein@enst.fr

Pierre-Alexandre Murena
LTCI, Télécom ParisTech
Paris, France
murena@telecom-paristech.fr

Talel Abdessalem
LTCI, Télécom ParisTech
Paris, France
UMI CNRS IPAL NUS
talel.abdessalem@enst.fr

Anthony Barré
AccorHotels
Paris, France
anthony.barre@live.fr

Antoine Cornuéjols
UMR MIA518, AgroParisTech INRA
Paris, France
antoine.cornuejols@agroparistech.fr

ABSTRACT

Collaborative filtering (CF) mainly suffers from rating sparsity and from the cold-start problem. Auxiliary information like texts and images has been leveraged to alleviate these problems, resulting in hybrid recommender systems (RS). Due to the abundance of data continuously generated in real-world applications, it has become essential to design online RS that are able to handle user feedback and the availability of new items in real-time. These systems are also required to adapt to drifts when a change in the data distribution is detected. In this paper, we propose an adaptive collaborative topic modeling approach, CoAWILDA, as a hybrid system relying on adaptive online Latent Dirichlet Allocation (AWILDA) to model newly available items arriving as a document stream and incremental matrix factorization for CF. The topic model is maintained up-to-date in an online fashion and is retrained in batch when a drift is detected using documents automatically selected by an adaptive windowing technique. Our experiments on real-world datasets prove the effectiveness of our approach for online recommendation.

KEYWORDS

Online recommendation; Concept drift; Topic modeling; Collaborative filtering

ACM Reference Format:

Marie Al-Ghossein, Pierre-Alexandre Murena, Talel Abdessalem, Anthony Barré, and Antoine Cornuéjols. 2018. Adaptive Collaborative Topic Modeling for Online Recommendation. In *Twelfth ACM Conference on Recommender Systems (RecSys '18), October 2–7, 2018, Vancouver, BC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240323.3240363

1 INTRODUCTION

Due to the abundance of available choices in online platforms and services, recommender systems (RS) have been playing an essential role to help users and empower companies. Approaches to recommendation can mainly be categorized into three classes. First, *content-based* (CB) approaches rely on information extracted from user profiles and item descriptions. Second, *collaborative filtering* (CF) approaches make use of user activities and past interactions, e.g., ratings and clicks, to learn preferences and generate recommendations. Lastly, *hybrid approaches* aim to combine both techniques in order to overcome their weaknesses: While CB methods tend to be overspecialized and lack a sense of novelty, the performance of CF methods drops with an increase of rating sparsity and in the cold-start setting. To get the best of both worlds, *hybrid* approaches allow CF approaches to exploit auxiliary information like text [41, 42] and images [20]. Collaborative Topic Regression (CTR) [41] is a popular hybrid approach combining probabilistic topic modeling for content analysis [6] and latent factor models for CF [36].

Most hybrid RS proposed in the literature are meant to work in batch where an initial model is first built from a static dataset and then rebuilt periodically as new chunks of data arrive. Since models are not continuously maintained up to date, they cannot capture the user feedback generated after a model update before the next one. They are also not able to adapt to changes in user preferences and item descriptions which occur due to temporal dynamics.

In real-world applications, the recommendation problem can be formulated as a data stream problem where RS are designed to learn from continuous data streams and adapt to changes in real-time. Recent work [15] has shown that simple online algorithms can generate better recommendations than more complex ones that are only updated periodically. Online RS are mainly based on incremental learning to continuously update models when receiving new observations. Incremental CF approaches have been proposed in this direction, like incremental neighborhood-based methods [32] and incremental matrix factorization [21, 40]. Learning from data streams should also account for concept drifts [17] which occur when the definition of modeled concepts changes over time. Detecting drifts and adapting to them become thus necessary.

In this paper, we address the problem of online recommendation in a dynamic environment where users interact with items in real-time and where new items are expected to arrive accompanied by a textual description. This setting is common in (but not restricted to) the news and tweet recommendation domains for example, where

RecSys '18, October 2–7, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5901-6/18/10…$15.00
https://doi.org/10.1145/3240323.3240363

new articles and tweets are continuously generated and read by users.

We propose an adaptive collaborative topic modeling approach for online recommendation. Our approach combines AWILDA, an adaptive version of online Latent Dirichlet Allocation (LDA) [23] that is able to analyze and model documents arriving in a stream on one side, and incremental matrix factorization [40] to leverage user interactions for the learning of preferences on the other side. Our approach is adaptive as we actively detect changes of topics that may occur in the document stream and adjust accordingly. We alternate online refinement of the topic model when no drift is found with batch training when it is needed. The decision of retraining and the chunk of data on which we retrain the model are automatically determined by an adaptive sliding window technique [3].

The purpose of introducing our approach is threefold. First, it is fully incremental and can thus be used in an online setting to generate recommendations. Second, since it is a hybrid approach relying on users' past interactions and on textual information, it addresses in particular the item cold-start problem which, to the best of our knowledge, has not been studied yet in the context of online recommendation. Third, its capacity of automatically detecting and adapting to drifts makes it suitable for real-world scenarios where changes in topics of document streams are frequently happening due to unexpected events and need to be considered for a better quality of recommendation.

The remainder of the paper is organized as follows. In Section 2, we discuss previous work on several topics related to our work. Section 3 proposes a reminder of LDA, ADWIN, and Incremental Matrix Factorization on which our method, detailed in Section 4, is based. Experiments and results are presented and discussed in Section 5. Finally, Section 6 concludes the paper.

2 RELATED WORK

In this Section, we review related work on hybrid RS, online RS, relevant variants of LDA, and news recommendation.

Hybrid RS. The cold-start problem [37] occurs in RS when new users or new items are introduced into the system. While recommending new and fresh items is essential, CF methods have difficulties in doing so, as no or few feedback related to these items is observed. Hybrid RS are able to recommend new items by leveraging auxiliary information. They also help alleviate the sparseness of rating or feedback data, thus improving the quality of recommendation.

To this end, previous work has utilized text data such as abstracts [41], synopses [42] or reviews [2]. Several techniques have been used to model documents like LDA [41], stacked denoising autoencoders [42] or convolutional neural networks [26]. Images have also been leveraged in this context and visual appearances of items can be added to the preference model [20].

While most hybrid RS are designed to work in batch, we propose an online hybrid RS that is able to address the item cold-start problem in a dynamic environment where items are added in real-time.

Online RS. Maintaining recommendation models up-to-date as new data is generated is essential to preserve the quality of recommendation over time [9]. Online RS are expected to learn from continuous data streams and adapt to changes in real-time. Elements of a data stream arrive in real-time at high rates and are processed sequentially using limited resources. Online RS are therefore based on incremental models like incremental neighborhood-based models [32] and incremental matrix factorization (using stochastic gradient descent [40] or alternating least squares [21]).

User preferences and item descriptions are expected to change over time in different ways, at different moments, and at different rates. While some existing approaches consider the evolution of entities over time [11, 18, 27], they are not adapted to the online setting. Incremental learning is a way of passively adapting to current changes in the data distribution by continuously learning from new data. Actively accounting for changes in user preferences has been based on the intuition that users' recent observations are more relevant than older ones. Sliding window techniques have been explored in this direction [31, 35, 38]. We note that these techniques make assumptions concerning the relevance of old observations and the rate at which all preferences drift, which are not always accurate.

From the point of view of the RS, a change of user preferences has an impact on a *local* scale since it only concerns a single user, whereas a change of item description or popularity affects the recommendation across all users and has a more *global* impact. Among the few works that considered concept drifts in the context of online recommendation, the focus has been on considering local changes occurring on the user level.

In this work, we manage to detect drifts on the item level in the topic model meant to handle the stream of documents describing items and we adapt the model when it is needed.

Topic modeling and concept drifts. Topic modeling is a machine learning task which consists in associating a document seen as an unordered list of words with a vector of *topics*, i.e., of word distributions. One of the most influential topic modeling method is the generative model of Latent Dirichlet Allocation (LDA) [6]. LDA models a document as a multivariate distribution, the parameter of which is drawn from a Dirichlet distribution. The popularity of LDA is due to its simplicity and modularity, as well as its interpretability.

In se, LDA is not designed for evolving environments but only to infer topics inside a batch of accessible documents. Variants have been proposed to include a temporal aspect to LDA, including situations where the distribution evolves over time. A first variant, called Dynamic Topic Models (DTM) [5], considers that the word-topic distribution varies over time. At each time step, this parameter is re-evaluated, conditioned by its value at the previous step. The same idea, but implemented at the level of a paragraph in a book, is proposed by SeqLDA [13]. A major drawback of these methods and other temporal adaptations of LDA (such as [19]) is the use of time slices, the size of which is arbitrary and does not depend on the observed data. In particular, changes in topic distribution can happen within a time period significantly smaller than the length of the chosen window. Continuous time models offer solutions to this problem [24]. A pioneer continuous time method [43] modifies LDA by assuming that word distribution over topics depends on word co-occurrences, but also on the date of the document. Despite the benefit of this method, it cannot be used in practice for stream analysis since the learning is made offline: it requires the whole dataset to be accessible in one batch to be able to infer the model.

For this reason, existing topic modeling methods are not suitable for data stream mining. Learning from data streams presents two major difficulties: the online nature of learning and *concept drift*. Concept drift designates the possible change in distribution that can happen in temporal and non-stationary environments [17]. Two main strategies are adopted to cope with concept drift. On one hand, passive methods adapt the model at each time step, no matter whether a change actually occurred. On the other hand, active methods focus on detecting drifts and adapt the model only when a drift is detected. Among active methods, ADaptive WINdowing (ADWIN) [3] gained a lot of interest recently for the simplicity of its approach (comparing average values of a time series on sub-windows) and for the theoretical guarantees it proposes.

The approach we propose in this paper is based on AWILDA [34] which combines LDA model and ADWIN algorithm for drift detection in data streams and which we present in Section 3.

News recommendation. The problem we address in this paper is common in the setting of tweets [10], articles [41], and news recommendation [14]. Our approach can be used to perform online recommendation of new items in any domain, whenever a textual description is available. We review in particular the related problem of news recommendation since it has been specifically studied in the literature.

News articles are continuously generated and while some of them could be relevant several weeks after the publication date, others have a shorter life cycle. Therefore, news recommendation often considers recency and popularity [1, 12]. This is done for example by filtering candidate articles for recommendation based on recency among other criteria [4]. Readers' interests are captured using the categories of the articles (if available) or keywords related to the articles' topics [12, 29] and are used for CB recommendation. Hybrid approaches combining CB and CF usually perform best for the problem of news recommendation [8, 28, 29]. Combining long-term and short-term preferences has also proven to be beneficial in this context [14, 29]. In addition, session-aware RS have been used to address this problem, given that users are not always identified when browsing on news platforms. Session-aware RS focus on transitions between items, formulating the problem as a Markov decision process [14] or using recurrent neural networks [22].

Our work addresses the wider and more generic problem of online recommendation that could occur in any domain and with no existence of sessions. We only require a textual description of items which can be easily collected from abstracts or reviews. We leverage CF and online topic models, and we consider the evolution of content through drift detection for modeling items.

3 PRELIMINARIES

The approach we propose merges three existing techniques of three unrelated domains. The purpose of this section is to present these techniques.

3.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [6] is a generative model describing text documents and corpora. The key notion involved in the description of a document is the notion of *topic*. A topic corresponds to a word distribution (for instance in a music-oriented corpus, the word "concert" would have a higher probability to be drawn than in a sport-oriented corpus) and a document is described as a mixture of topics. Topics are learned in an unsupervised fashion and, thus, do not necessarily correspond to human-understandable concepts [7]. The generative process, presented in Figure 1, is described as follows:

(1) Choose $\theta \sim Dirichlet(\alpha)$
(2) For each word w_n in document:
    - Choose a topic $z_n \sim Mult(\theta)$
    - Choose a word w_n from the multinomial $p(w_n|z_n, \beta)$

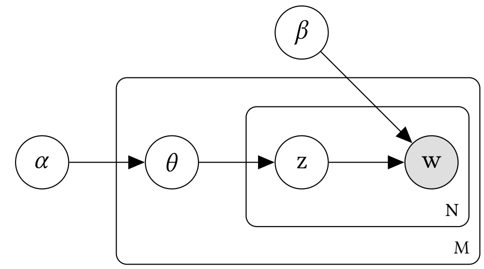

Figure 1: Generative model for Latent Dirichlet Allocation

In terms of document analysis, the parameters of LDA can be understood in the following way:

- Vector α represents the global topic trend. For instance, a parameter $\alpha = (1, \ldots, 1)$ corresponds to a uniform choice over all topics, on average. The higher the component α_i, the more frequent topic i will be in the whole corpus.
- Matrix β stores the probability of words inside topics. If a word w is set to belong to topic z, then it will be chosen with probability $\beta_{z,w}$.
- Vector θ corresponds to the topic distribution inside one document.

A training of LDA model is possible based on maximum likelihood principle [6]. In practice, it is suggested to use either Gibbs sampling or variational inference for this task. An online version of variational inference is usually preferred [23] based on a stochastic gradient descent. In this version, data are not supposed to be received as a batch, but are processed one by one. However, the context is not the same as stream mining. In online LDA, all data are generated by the same distribution (stationary) and concept drift is not considered.

3.2 Adaptive Sliding Window

Adaptive Sliding Window algorithm (ADWIN) [3] is an active strategy used for classification of data with concept drift adaptation. The algorithm uses a sliding window W to detect a change in a series of one-dimensional observations. In a supervised context, these observations usually correspond to risk measures. The scores are stored by ADWIN in a sliding window W at each observation.

A drift is detected if the window W can be separated into two sub-windows $W = W_0 W_1$ such that the difference of means μ_{W_0} and μ_{W_1} in the sub-windows is large enough (above a parameter ϵ).

In addition to the extreme simplicity of the method, ADWIN also benefits from a strong theoretical justification. In particular, it can be shown that, depending on the threshold ϵ only, probabilities of incorrectly detecting a drift (false positive rate) and correctly detecting a drift (false negative rate) are bounded.

3.3 Incremental Matrix Factorization

Matrix factorization (MF) is a popular collaborative filtering technique used to model users' interactions by representing users and items in a space of latent factors learned from the data. If R designates the matrix of interactions (where $R_{ui} = 1$ if the user u interacted with the item i, and 0 otherwise), then MF aims to approximate R as a product of two matrices P and Q by minimizing over P and Q:

$$\sum_{(u,i)\in D} \left(R_{ui} - P_u Q_i^T\right)^2 + \lambda_u \|P_u\|^2 + \lambda_i \|Q_i\|^2 \qquad (1)$$

where D is the set of observed interactions, and λ_u and λ_i are regularization parameters. The score of an item i for a user u, denoted by $\hat{R}_{ui}$, is computed using the scalar product between P_u and Q_i^T. Items are ordered by descending proximity of $\hat{R}_{ui}$ to the value 1, and top-N items are recommended for u.

Classic algorithms for MF are not suitable for a data stream setting. A variant of MF adapted to the incremental nature of data streams [40] suggests the following procedure. Observations $\langle u, i\rangle$ are received one after the other and handled by the algorithm. For each received observation, P and Q are updated using the gradient of the objective for this observation only (which corresponds to an estimator of the gradient on the whole dataset). When either a user or an item are observed for the first time, they are added to the matrices with a random initialization, and the values of P and Q are then updated using the observation.

4 ADAPTIVE COLLABORATIVE TOPIC MODELING

In this Section, we present the proposed algorithm for adaptive collaborative topic modeling. Our algorithm is split into two components: drift detection in topic modeling and collaborative topic modeling.

4.1 Adaptive Window based Incremental LDA

We first present the algorithm used for topic drift detection in LDA which merges the drift detection property of ADWIN with parameter estimation in the training of the LDA model.

The algorithm, called *Adaptive Window based Incremental LDA* (AWILDA) [34], is based on the idea that a change of distribution will induce a drift in the likelihood of the model. Following this idea, a change in the LDA distribution can be detected by ADWIN by processing the series of likelihoods.

AWILDA is based on two models of LDA. The first model, denoted by LDA_m, is used for document modeling. The second model, denoted by LDA_d, is used for the detection of drifts only. When a new document is received, the algorithm works as follows:

(1) Compute likelihood $\mathcal{L} = p(w|LDA_d)$ for model LDA_d.
(2) Process $\mathcal{L}$ with ADWIN.
(3) If ADWIN detects a drift for window decomposition $W = W_0 W_1$:
  - Retrain LDA_m based on the documents in W_1.
  - Retrain LDA_d based on the documents in W_1.
(4) Update LDA_m from the new document based on the online LDA algorithm.

The idea is to split the task of prediction and the task of drift detection by separating the models. The model used for prediction, LDA_m, is kept up to date while no drift is detected. On the contrary, the model LDA_d is not modified; otherwise, the detected changes might not originate only from a change in the data distribution, but also from the change of the model.

The AWILDA algorithm benefits from all the advantages of ADWIN. In particular, it offers a strict control of false positive rate and false negative rate. If the underlying data generation process does not change, the distribution of the likelihood is stationary, which makes the theorems relative to ADWIN still valid.

In practice, the likelihood cannot be used directly: Since it measures probabilities, the values of likelihood observed on real (and artificial) data are very low and cannot be well distinguished. For this reason, we use the loglikelihood in practice. The value of the loglikelihood is not bounded which is a problem with respect to the theory of ADWIN. However, we noticed that the values evolve inside an interval of low amplitude and can be lower-bounded.

The loglikelihood of an LDA model is not computable in practice. Consequently, we use the lower-bound proposed by [6] and [23] as part of the variational inference training process.

4.2 Adaptive Collaborative Topic Modeling

Collaborative Topic Modeling (also called Collaborative Topic Regression, or CTR) [41] is a popular framework which was initially introduced to recommend scientific articles. CTR assumes that documents describing items are generated by LDA and it represents users with topic interests. CTR introduces an additional latent variable that offsets the topic proportions when modeling the user interactions. This offset is learned from the feedback data using CF and is added to represent the fact that two items having similar topic proportions can be interesting to different types of users. The topic model used in CTR is LDA and the latent factor model is Probabilistic Matrix Factorization [33].

CTR alleviates the problem of feedback sparsity by leveraging auxiliary information. It is also able to recommend existing items and new items in the case of cold start, and proposes an interpretable latent structure for users and items. However, the models used in CTR are learned offline and CTR is not adapted for online recommendation. In the following, we propose CoAWILDA which benefits from the cited advantages of CTR, is adapted to the online setting, and takes into account the non-stationarity nature of data in an evolving environment.

In our setting, observations are supposed to arrive in real-time and are mainly of two types. First, *interactions*, denoted by $\langle u, i \rangle$, designate positive actions (clicks, ratings) performed by users and concerning a certain item. Second, *additions of items*, denoted by $\langle i, doc_i \rangle$, usually occur when a new item becomes available at a certain time step and we consider cases where a textual description of the new item is provided.

Algorithm 1 Overview of CoAWILDA

Data: set of observations O
Input: number of latent factors K, learning rate η, regularization parameters λ_u and λ_i
Output: P, Q

for o in O **do**
  if $o = \langle i, doc_i \rangle$ **then** ▹ new item added
    $\theta_i \leftarrow AWILDA(doc_i)$
    $\epsilon_i \sim \mathcal{N}(0, \lambda_i^{-1} I_K)$
    $Q_i \leftarrow \theta_i + \epsilon_i$
  end if
  if $o = \langle u, i \rangle$ **then** ▹ interaction received
    if $u \notin Rows(P)$ **then** ▹ new user observed
      $P_u \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$
    end if
    $e_{ui} \leftarrow 1 - P_u.Q_i^T$
    $P_u \leftarrow P_u + \eta(e_{ui} Q_i - \lambda_u P_u)$
    $\epsilon_i \leftarrow \epsilon_i + \eta(e_{ui} P_u - \lambda_i \epsilon_i)$
    $Q_i \leftarrow \theta_i + \epsilon_i$
  end if
end for

CoAWILDA is presented in Algorithm 1. When a new item is received, we use AWILDA to model the descriptive document and extract topic proportions θ_i. The item latent vector Q_i representing an item i results of the addition of the topic proportions θ_i and an item latent offset ϵ_i. When a new interaction $\langle u, i \rangle$ is observed, we update the user latent factor P_u and the item latent offset ϵ_i following the procedure of incremental MF. Recommendation is performed as described in Section 3.3 where $\hat{R}_{ui} = P_u.Q_i^T = P_u.(\theta_i + \epsilon_i)^T$.

5 EXPERIMENTS

In this Section, we present the experiments we conducted to prove the effectiveness of our approach. We first show how AWILDA performs when modeling a stream of documents and then discuss how our approach performs when addressing the problem of online recommendation, using real-world datasets.

Datasets. We study the problem of online recommendation where observations are being received in real-time. Data used to evaluate our approach should be chronologically ordered and should include user interactions and the addition of items over time with a corresponding textual description. We note that these two characteristics are not always available in datasets commonly used to evaluate RS. In our work, we use two real-world datasets: the *ml-100k* and the *plista* datasets.

The *ml-100k* dataset corresponds to the MovieLens 100k dataset [1] and gathers 100,000 ratings from 1,000 users on 1,700 movies, spanning over 18 months. Since we are addressing the problem of recommendation with implicit feedback (positive-only data), our goal is to recommend the movies the user is going to rate. We note that the dates reported in *ml-100k* correspond to the rating date of the movies and not to the actual watching date. Knowing that we are not concerned with the problem of evolution of user preferences in this work, we use *ml-100k* to evaluate our approach. Movies become available according to their reported release date, and we use DBpedia [2] to collect abstracts written in English and describing each one of them.

The *plista* dataset is described in [25] and contains a collection of news articles published in German on several news portals. The available dataset captures interactions collected during the month of February 2016. We remove from the dataset interactions corresponding to unknown users, users with less than three interactions, and items with no available textual description. Finally, the dataset gathers 32,706,307 interactions from 1,362,097 users on 8,318 news articles.

Documents from both datasets have been preprocessed by mainly removing stop words, removing words occurring once, and stemming remaining words.

5.1 Performance of AWILDA for topic modeling

Evaluation protocol. AWILDA is proposed to model a stream of documents using drift detection. In our experimental setting, we consider that we are receiving documents describing items one after the other, ordered by their availability date (e.g., release or publication date). For each received document, we evaluate the topic model and process the document to update the underlying model. We use the first 20% documents of the stream to train the model which is evaluated on the remaining documents.

Evaluation measure. The goal in document modeling is to maximize the likelihood on unseen documents (D_{test}), given a trained topic model. Perplexity measures the ability of a model to generalize to new data and is used to evaluate topic models [6]. Perplexity is defined as follows:

$$perplexity(D_{test}) = \exp\left\{-\frac{\sum_{d=1}^{M} \log_2 p(w_d)}{\sum_{d=1}^{M} N_d}\right\} \quad (2)$$

where M designates the total number of documents in D_{test}. In the actual use of perplexity, the probability $p(w_d)$ is approximated by its upper-bound (given by the variational inference) as explained in Section 4.1. A lower value of perplexity indicates a better generalization capacity. Since we are handling document streams, the perplexity is reported for each received document using the current model.

Methods compared. We compare the performance of AWILDA to the online version of LDA [23], considering its capacity to handle document collections arriving in a stream. We show the results for

[1] http://www.movielens.org
[2] http://www.dbpedia.org

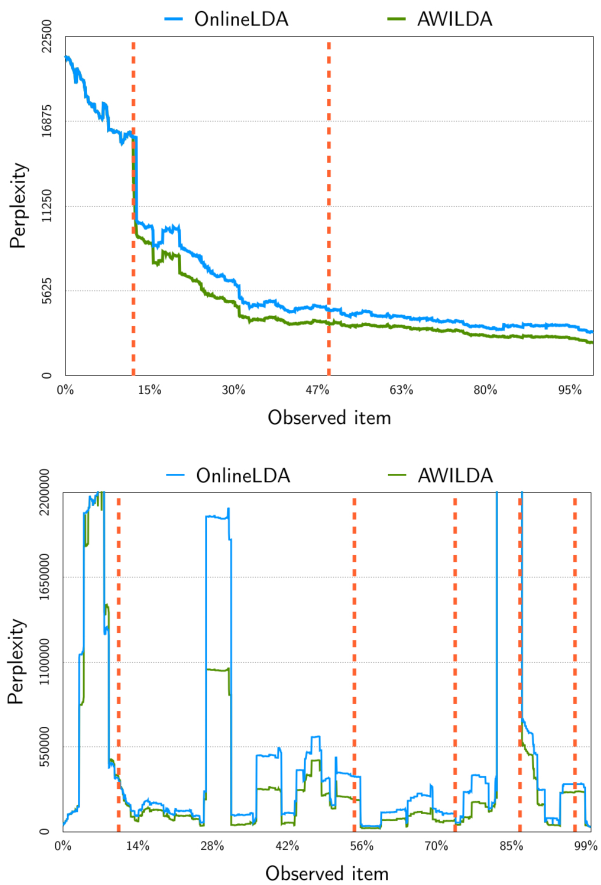

Figure 2: Performance evaluation of online LDA and AWILDA for the task of document stream modeling on *ml-100k* (first subfigure) and *plista*, (second subfigure) using the measure of perplexity.

experiments where we set the number of topics to 10, knowing that similar patterns appear for different values of this parameter.

Results. Figure 2 shows the perplexity measured on the document streams of *ml-100k* and *plista* for AWILDA and online LDA. The perplexity is represented as a moving average with a sliding window of 200 observations. The red dotted vertical line marks the detection of a drift by AWILDA. AWILDA detects two drifts for the held-out documents of *ml-100k* and five drifts for *plista*. This difference in behavior is expected knowing the volume and nature of both datasets (movies vs. news).

Before detecting any drift, online LDA and AWILDA are trained in the same way and on the same data, which explains the close values of perplexity. After detecting the first drift, AWILDA outperforms online LDA for the task of document modeling. As documents continue to arrive, AWILDA is more adapted to the new data. Its drift detection component allows it to adjust to changes after each drift, resulting in a better performance. A further analysis of the datasets with experts from both domains will help to establish the link between the detected drifts and real-life events occurring in the same time period, for better understanding and explainability.

5.2 Performance of CoAWILDA for online recommendation

Evaluation protocol. RS are traditionally evaluated using holdout methods. These methods are not adapted to the online setting [39] mainly because when we randomly sample data for training and testing, we lose the temporal dimension and do not respect the original order of observations.

Since the topic model requires an initial phase of training, we adopt the evaluation process introduced in [30]. We sort the dataset chronologically and then split it into the following three subsets:

- **Batch Train** subset. The first 20% of the dataset is used for the initial training of the models.
- **Batch Test - Stream Train**. The next 30% of the dataset is used for the validation of the initialized models, and for incremental online learning to ensure the transition between the first and the last phase.
- **Stream Test and Train**. The last 50% of the dataset is used for prequential evaluation, which is a test-then-learn procedure performed while iterating over the observations [16]. Each observation $\langle u, i \rangle$ is used to evaluate the model by generating recommendations for user u and then to update the model using $\langle u, i \rangle$.

Evaluation measures. We use recall@N and DCG@N to measure the quality of recommendation. These metrics are described in [15] for the online setting. We report the results for the *Stream Test and Train* subset.

Parameters. We performed a grid search over the parameter space of the methods in order to find the parameters that give the best performance. We report the performance corresponding to the parameters leading to the best results. The parameters are reported along with the methods below.

Methods compared. Since previous work has demonstrated the advantages of using online recommendation compared to batch recommendation [15, 40], we focus on incremental methods. We also only consider one approach for incremental MF, knowing that our method, CoAWILDA, can integrate any other algorithm for incremental MF or any model-based method. We compare the performances of several incremental methods adapted to the online setting, including variants of the one we propose.

- **CoAWILDA** is the method we propose, combining ADaptive Window based Incremental LDA (AWILDA) for topic modeling and incremental MF for CF. For *ml-100k*, we set the number of topics $K = 20$, $\eta = 0.04$, $\lambda_u = 0.01$, and $\lambda_i = 0.1$. For *plista*, we set $K = 10$, $\eta = 0.042$, $\lambda_u = 0.01$, and $\lambda_i = 0.1$.
- **CoLDA** relies on classical online LDA [23] for topic modeling and incremental MF for CF. It replaces AWILDA from CoAWILDA with classical online LDA. For *ml-100k*, we set $K = 20$, $\eta = 0.05$, $\lambda_u = 0.01$, and $\lambda_i = 0.1$. For *plista*, we set $K = 10$, $\eta = 0.045$, $\lambda_u = 0.01$, and $\lambda_i = 0.1$.
- **AWILDA** denotes the method we propose for adaptive topic modeling. We try to use it for recommendation without

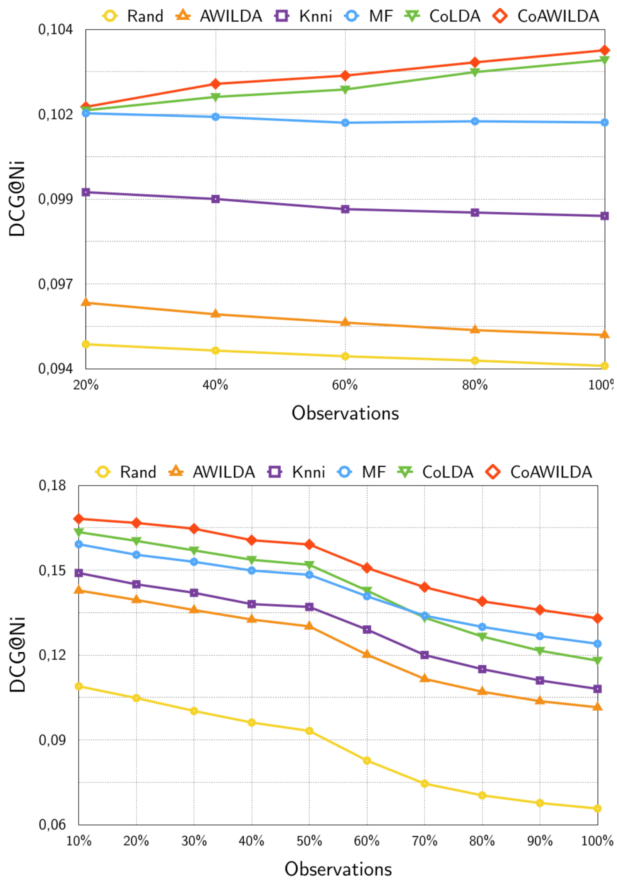

Figure 3: DCG@Ni of our approach CoAWILDA and other variants and incremental methods for *ml-100k* (first subfigure) and *plista* (second subfigure), where Ni is the number of available items. The evolution of DCG@Ni with the number of evaluated observations is reported.

the collaborative component by representing users in the space of topics and updating their profiles as we get more observations. For *ml-100k*, we set $K = 20$, $\eta = 0.04$, and $\lambda_u = 0.01$. For *plista*, we set $K = 10$, $\eta = 0.042$, and $\lambda_u = 0.01$.

- **MF** is the incremental MF [40]. Compared to CoAWILDA and CoLDA, MF does not leverage content information about items. For *ml-100k*, we set $K = 50$, $\eta = 0.01$, $\lambda_u = 0.02$, and $\lambda_i = 0.02$. For *plista*, we set $K = 50$, $\eta = 0.008$, $\lambda_u = 0.01$, and $\lambda_i = 0.01$.
- **Knni** is the incremental item-based approach proposed in [32]. We set the number of neighbors to 300.
- **Rand** randomly selects items for recommendation.

Results. Figure 3 shows the DCG@Ni of the methods we compare for *ml-100k* and *plista*, where Ni is the total number of items included in each dataset. The idea is to evaluate how each approach performs when ranking the items for each user. We report the metric value with respect to the number of observations processed in order to analyze its evolution over the time spanned by the *Stream Test and Train* set.

CoAWILDA outperforms all the other methods evaluated for both datasets. The comparison between CoAWILDA and CoLDA demonstrates the effectiveness of AWILDA for modeling document streams describing new items and for improving the quality of item modeling and thus recommendation. CoLDA is not able to adjust to drifts occurring in topic modeling which deteriorates the recommendation quality over time.

The performance of CoLDA for *plista* can be divided into two phases. In the first one, the topic model is still able to carry out good document modeling and is beneficial for the recommendation: CoLDA performs better in terms of item ranking than MF which does not account for content analysis. In the second phase, and with the incapacity of online LDA to adjust to drifts, MF outperforms CoLDA. This means that not only the topic model is not adapted to newly received data, but it is also badly affecting the recommendation quality and there is no interest in using it anymore. We also note the importance of evaluating the evolution of the models over time to show how they are affected by eventual changes occurring in the data. This phenomenon appears for *plista* where more frequent drifts occur over time, mainly due to the nature of news data. Concerning *ml-100k*, CoLDA performs better than MF but still worse than CoAWILDA.

AWILDA is a content-based method and only relies on topics extracted from items to model user preferences. It performs poorly compared to the other methods and proves the importance of having a CF component. Knni performs better than AWILDA but is not as robust as MF and the hybrid approaches evaluated.

The number of available items grows significantly over time in *plista*. This results in the dropping of performance (in terms of ranking) of all methods over time. This is not the case in *ml-100k*, since only few movies are added in the corresponding time period. More data is received and more learning is done over time, which can explain the improvements in the performance of CoAWILDA and CoLDA.

Figure 4 and Figure 5 show the recall@5, recall@10, recall@50, and recall@100 of our approach, CoAWILDA, and its variants on *ml-100k* and *plista* respectively.

The experiments for *ml-100k* confirm the ideas we mentioned before. CoAWILDA outperforms the other variants and performs better than CoLDA which relies on online LDA and does not adapt to changes in the data. CoLDA performs better than MF demonstrating the benefits of using content information. AWILDA relies only on content information which is a weak approach to model user preferences when used alone.

The experiments for *plista* highlight an interesting behavior. For recall@5 and recall@10, MF performs better than CoLDA for all considered observations. For recall@50 and recall@100, we observe two different behaviors where, first, CoLDA performs better than MF, and then MF outperforms CoLDA. We recall that the reported results are measured on the second half of the dataset (*Stream Test and Train* subset). Drifts may have occurred during the training phase, which is typically the case for *plista*. When measuring the recall@N, CoLDA is already weakened by the drifts that have happened and that were not taken into account. This leads to a point

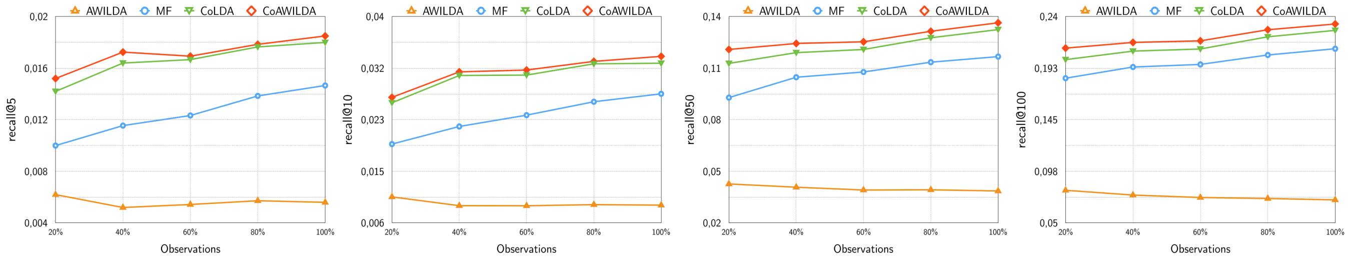

Figure 4: Recall@5, recall@10, recall@50, and recall@100 of our approach CoAWILDA and its variants on *ml-100k*.

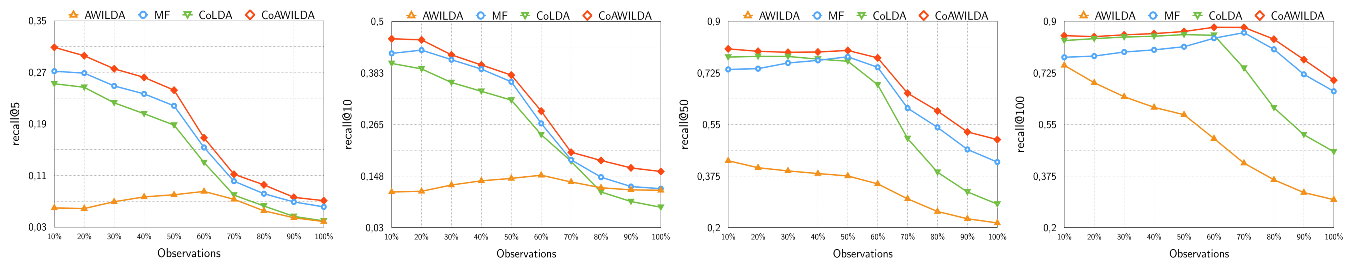

Figure 5: Recall@5, recall@10, recall@50, and recall@100 of our approach CoAWILDA and its variants on *plista*.

where the information learned by the topic model hurts the quality of recommendation, and MF starts performing better than CoLDA. This change of behavior occurs at different moments, depending on the recall we are measuring. For a higher N (i.e., recall@50, recall@100), the performance of CoLDA remains above the performance of MF for a longer time than for a lower N (i.e., recall@5, recall@10). Top list recommendation is thus more affected by the deterioration of the topic model.

All experiments demonstrate the effectiveness of using CoAWILDA, the strength of which relies on adapting to changes occurring in the data. Methods that do not detect and adapt to these changes, i.e., CoLDA, perform worse than CoAWILDA.

6 CONCLUSION

In this paper, we address the problem of online recommendation in dynamic environments and we tackle several subjects at once: hybrid approach for recommendation (merging CF and CB), recommendation with concept drifts, and collaborative topic modeling. The solution we propose, which we call CoAWILDA, is designed for online recommendation where textual descriptions of items are provided as new items are becoming available. It combines the advantages of the drift detection method ADWIN, the flexibility of online LDA, and the online nature of incremental matrix factorization. In the proposed setting, user interactions arrive in a stream and the method adapts to new items becoming available and to user interactions. Leveraging the advantages of both document analysis and users' interactions, CoAWILDA is particularly suitable to alleviate the problem of cold start and offers an elegant and generic solution to deal with drifting item distributions. Since very few training is necessary, the algorithm is truly online and can run in real-time.

An experimental validation on two real-world datasets has shown the actual efficiency of the proposed methodology. CoAWILDA outperforms its variants and other incremental methods we evaluated. In particular, it has been shown that a topic model which is not adjusting to drifts can hurt the quality of recommendation to an extent where its removal results in a much better performance. For that matter, recommendation at the top of the list is firstly affected.

In the context of this paper, we focused on the task of topic mining on textual items, but the idea of combining item modeling with a drift detection method (in particular with ADWIN) can be extended to other domains. Future work will consider new domains and will investigate how adaptive sliding window drift detection can be extended to improve online recommendation.

REFERENCES

[1] Amr Ahmed, Choon Hui Teo, SVN Vishwanathan, and Alex Smola. 2012. Fair and balanced: Learning to present news stories. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 333–342.

[2] Yang Bao, Hui Fang, and Jie Zhang. 2014. TopicMF: simultaneously exploiting ratings and reviews for recommendation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, 2–8.

[3] Albert Bifet and Ricard Gavalda. 2007. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 443–448.

[4] Daniel Billsus and Michael J Pazzani. 2007. Adaptive news access. In *The adaptive web*. Springer, 550–570.

[5] David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 113–120.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[7] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*. 288–296.

[8] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 271–280.

[9] M Benjamin Dias, Dominique Locher, Ming Li, Wael El-Deredy, and Paulo JG Lisboa. 2008. The value of personalised recommender systems to e-business: a case study. In *Proceedings of the 2008 ACM conference on Recommender systems*. ACM, 291–294.

[10] Ernesto Diaz-Aviles, Lucas Drumond, Lars Schmidt-Thieme, and Wolfgang Nejdl. 2012. Real-time top-n recommendation in social streams. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 59–66.

[11] Yi Ding and Xue Li. 2005. Time weight collaborative filtering. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 485–492.

[12] Doychin Doychev, Aonghus Lawlor, Rachael Rafter, and Barry Smyth. 2014. An analysis of recommender algorithms for online news. In *CLEF 2014 Conference and Labs of the Evaluation Forum: Information Access Evaluation Meets Multilinguality, Multimodality and Interaction, 15-18 September 2014, Sheffield, United Kingdom*. 177–184.

[13] Lan Du, Wray Lindsay Buntine, and Huidong Jin. 2010. Sequential latent dirichlet allocation: Discover underlying topic structures within a document. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 148–157.

[14] Elena Viorica Epure, Benjamin Kille, Jon Espen Ingvaldsen, Rebecca Deneckere, Camille Salinesi, and Sahin Albayrak. 2017. Recommending Personalized News in Short User Sessions. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 121–129.

[15] Erzsébet Frigó, Róbert Pálovics, Domokos Kelen, Levente Kocsis, and András A. Benczúr. 2017. Online Ranking Prediction in Non-stationary Environments. In *Proceedings of the 1st Workshop on Temporal Reasoning in Recommender Systems co-located with 11th International Conference on Recommender Systems (RecSys 2017)*. ACM, 28–34.

[16] João Gama, Raquel Sebastião, and Pedro Pereira Rodrigues. 2009. Issues in evaluation of stream learning algorithms. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 329–338.

[17] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 46, 4 (2014), 44.

[18] Li Gao, Jia Wu, Chuan Zhou, and Yue Hu. 2017. Collaborative dynamic sparse topic regression with user profile evolution for item recommendation. In *31st AAAI Conference on Artificial Intelligence, AAAI 2017*.

[19] Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences* 101, suppl 1 (2004), 5228–5235.

[20] Ruining He and Julian McAuley. 2016. VBPR: visual Bayesian Personalized Ranking from implicit feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 144–150.

[21] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 549–558.

[22] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. *International Conference on Learning Representations*.

[23] Matthew Hoffman, Francis R Bach, and David M Blei. 2010. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*. 856–864.

[24] Tomoharu Iwata, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. 2010. Online multiscale dynamic topic models. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 663–672.

[25] Benjamin Kille, Frank Hopfgartner, Torben Brodt, and Tobias Heintz. 2013. The plista dataset. In *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*. ACM, 16–23.

[26] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 233–240.

[27] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.

[28] Chen Lin, Runquan Xie, Xinjun Guan, Lei Li, and Tao Li. 2014. Personalized news recommendation via implicit social experts. *Information Sciences* 254 (2014), 1–18.

[29] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 31–40.

[30] Pawel Matuszyk and Myra Spiliopoulou. 2014. Selective forgetting for incremental matrix factorization in recommender systems. In *International Conference on Discovery Science*. Springer, 204–215.

[31] Pawel Matuszyk, João Vinagre, Myra Spiliopoulou, Alípio Mário Jorge, and João Gama. 2015. Forgetting methods for incremental matrix factorization in recommender systems. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. ACM, 947–953.

[32] Catarina Miranda and Alípio Mário Jorge. 2009. Item-based and user-based incremental collaborative filtering for web recommendations. In *Portuguese Conference on Artificial Intelligence*. Springer, 673–684.

[33] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[34] Pierre-Alexandre Murena, Marie Al-Ghossein, Talel Abdessalem, and Antoine Cornuéjols. 2018. Adaptive Window Strategy for Topic Modeling in Document Streams. In *International Joint Conference on Neural Networks (IJCNN)*.

[35] Olfa Nasraoui, Jeff Cerwinske, Carlos Rojas, and Fabio Gonzalez. 2007. Performance of recommendation systems in dynamic streaming environments. In *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 569–574.

[36] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 502–511.

[37] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 253–260.

[38] Zaigham Faraz Siddiqui, Eleftherios Tiakas, Panagiotis Symeonidis, Myra Spiliopoulou, and Yannis Manolopoulos. 2014. xstreams: Recommending items to users with time-evolving preferences. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*. ACM, 22.

[39] João Vinagre, Alípio Mário Jorge, and João Gama. 2014. Evaluation of recommender systems in streaming environments. In *Proceedings of the Workshop on Recommender Systems Evaluation: Dimensions and Design in conjunction with the 8th ACM Conference on Recommender Systems (RecSys 2014)*.

[40] João Vinagre, Alípio Mário Jorge, and João Gama. 2014. Fast incremental matrix factorization for recommendation with positive-only feedback. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 459–470.

[41] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[42] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[43] Xuerui Wang and Andrew McCallum. 2006. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 424–433.