Probabilistic Sequence Modeling for Recommender Systems

Nicola Barbieri[1,2], Antonio Bevacqua[1], Marco Carnuccio[1], Giuseppe Manco[2] and Ettore Ritacco[2]
[1]*Department of Electronics, Informatics and Systems - University of Calabria, via Bucci 41c, 87036 Rende (CS) - Italy*
[2]*Institute for High Performance Computing and Networks (ICAR) Italian National Research Council via Bucci 41c, 87036 Rende (CS) - Italy*
{*barbieri,manco,ritacco*}*@icar.cnr.it*,{*abevacqua,mcarnuccio*}*@deis.unical.it*

Keywords: Recommender Systems, Collaborative Filtering, Probabilistic Topic Models, Performance.

Abstract: Probabilistic topic models are widely used in different contexts to uncover the hidden structure in large text corpora. One of the main features of these models is that generative process follows a bag-of-words assumption, i.e each token is independent from the previous one. We extend the popular Latent Dirichlet Allocation model by exploiting a conditional Markovian assumptions, where the token generation depends on the current topic and on the previous token. The resulting model is capable of accommodating temporal correlations among tokens, which better model user behavior. This is particularly significant in a collaborative filtering context, where the choice of a user can be exploited for recommendation purposes, and hence a more realistic and accurate modeling enables better recommendations. For the mentioned model we present a fast Gibbs Sampling procedure for the parameters estimation. A thorough experimental evaluation over real-word data shows the performance advantages, in terms of recall and precision, of the proposed sequence-modeling approach.

1 INTRODUCTION

Probabilistic topic models, such as the popular *Latent Dirichlet Allocation (LDA)* (Blei et al., 2003), assume that each collection of documents exhibits an hidden thematic structure. The intuition is that each document may exhibit multiple topics, where each topic is characterized by a probability distribution over words of a fixed size dictionary. This representation of the data into the latent-topic space has several advantages, as topic modeling techniques have been applied to different contexts. Example scenarios range from traditional problems (such as dimensionality reduction and classification) to novel areas (such as the generation of personalized recommendations). In most cases, LDA-based approaches have been shown to outperform state-of-art approaches.

Traditional LDA-based approaches propose a data generation process that is based on a "bag-of-words" assumption, i.e. such that the order of the items in a document can be neglected. This assumption fits textual data, where probabilistic topic models are able to detect recurrent co-occurrence patterns, which are used to define the topic space. However, there are several real-word applications where data can be "naturally" interpreted as sequences, such as biological data, web navigation logs, customer purchase history, etc. Interpreting sequence in accordance to "exchangeability", i.e., by ignoring the intrinsic sequentiality of the data within, may result in poor modeling: according to the bag-of-word assumption, co-occurrences is modeled independently for each word, via a probability distribution over the dictionary in which some words exhibit an higher likelihood to appear than others. On the other hand, sequential data may express causality and dependency, and different topics can be used to characterize different dependency likelihoods. In practice, a sequence expresses a **context** which provides valuable information for a more refined modeling.

The above observation is particularly noteworthy when data expresses preferences made by users, and the ultimate objective is to model a user's behavior in order to provide accurate recommendations. The analysis of the sequential patterns has important applications in modern recommender systems, which are always more focused on an accurate balance between personalization and contextualization techniques. For example, in Internet based streaming services for music or video (such as Last.fm[1] and

[1] `http://last.fm`

Videolectures.net[2]), the context of the user interaction with the system can be easily interpreted by analyzing the content previously requested. The assumption here is that the current item (and/or its genre) influences the next choice of the user.

Recommender systems have greatly benefited from probabilistic modeling techniques based on LDA. Recent works in fact have empirically shown that probabilistic latent topics models represent the state-of-the art in the generation of accurate personalized recommendations (Barbieri and Manco, 2011; Barbieri et al., 2011b; Barbieri et al., 2011a). Probabilistic techniques offer some advantages over traditional deterministic models: notably, they do not minimize a particular error metric but are designed to maximize the likelihood of the model given the data which is a more general approach; moreover, they can be used to model a distribution over rating values which can be used to determine the confidence of the model in providing a recommendation; finally, they allow the possibility to include prior knowledge into the generative process, thus allowing a more effective modeling of the underlying data distribution. Notably, when preferences are implicitly modeled through selection (that is, when no rating information is available), the simple LDA best models the probability that an item is actually selected by a user (Barbieri and Manco, 2011).

A simple approach to model sequential data within a probabilistic framework has been proposed in (Cadez et al., 2000). In this work, authors present a framework based on mixtures of Markov models for clustering and modeling of web site navigation logs, which is applied for clustering and visualizing user behavior on a web site. Albeit simple, the proposed model suffers of the limitation that a single latent topic underlies all the observation in a single sequence. This approach has been overtaken by other methods based on latent semantic indexing and LDA. In (Wallach, 2006; X. Wang and Wei, 2007), for example, the authors propose extension of the LDA model which assume a first-order Markov chain for the word generation process. In the resulting *Bigram Model (BM)* and *Topical n-grams*, the current word depends on the current topic and the previous word observed in the sequence. The LDA Collocation Model (Griffiths et al., 2007) introduces a new set of random variables (for bigram status) x which denotes whether a bigram can be formed with the previous word token. The bigram status adds a more realistic than Wallach model which always generates bigrams.

Hidden Markov models (Bishop, 2006, Chapter 13) are a general reference framework for modeling sequence data. HMMs assume that sequential data are generated using a Markov chain of latent variables, with each observation conditioned on the state of the corresponding latent variable. The resulting likelihood can be interpreted as an extension of a mixture model in which the choice of mixture components for each observation is not selected independently but depends on the choice of components for the previous observation. (Gruber et al., 2007) delve in this direction, and propose an *Hidden Topic Markov Model (HTMM)* for text documents. HTTM define a Markov chain over latent topics of the document. The corresponding generative process assume that all words in the same sentence share the same topic, while successive sentences can either rely on the previous topic, or introduce a new one. The topics in a document form a Markov chain with a transition probability that depends on a binary topic transition variable ψ. When $\psi = 1$, a new topic is drawn for the n-th sentence, otherwise the same previous topic is used.

Following the research direction outlined above, in this paper we study the effects of "contextual" information in probabilistic modeling of preference data. We focus on the case where the context can be inferred from the analysis of the sequence data, and we propose a topic model which explicitly makes use of dependency information for providing recommendations. As a matter of fact, the issue has been dealt with in similar papers (like, e.g. (Wallach, 2006)). Here, we resume and extend the approaches in the literature. by concentrating on the effects of such modeling on recommendation accuracy, as it explicitly reflects accurate modeling of user behavior.

In short, the contributions of the paper can be summarized as follows.

1. We propose an unified probabilistic framework to model dependency in preference data, and instantiate the framework in accordance to a specific assumption on the sequentiality of the underlying generative process;
2. For the proposed instance, we provide the relative ranking function that can be used to generate personalized and context-aware recommendation lists;
3. We finally show that the proposed sequential modeling of preference data better models the underlying data, as it allows more accurate recommendations in terms of precision and recall.

The paper is structured as follows. In Sec. 2 we introduce sequential modeling, and specify in Sec. 2.1 the corresponding item ranking functions for supporting recommendations. The experimental evaluation of the proposed approaches in then presented in

[2] http://videolectures.net

Table 1: Summary of the notation used

notation	*description*
M	# Users
N	# Items
K	# Topics
$\mathbf{W}$	Collection of users' traces, $\mathbf{W} = \{\vec{w}_1, \ldots, \vec{w}_M\}$
n_u	# Items in the user u's trace
$\vec{w}_u$	Item trace of user u, $\vec{w}_u = \{w_{u,1}.w_{u,2}.\cdots.w_{u,N_u-1}.w_{u,n_u}\}$
$w_{u,n}$	n-th item in the trace of user u
$\mathbf{Z}$	Collection of topic traces for each user, $\mathbf{Z} = \{\vec{z}_1, \ldots, \vec{z}_M\}$
$\vec{z}_u$	Topic trace for user u, $\vec{z}_u = \{z_{u,1}.z_{u,2}.\cdots.z_{u,N_u-1}.z_{u,n_u}\}$
$z_{u,n}$	n-th topic in the trace of user u
$n^k_{d,i}$	number of times item i has been associated with topic k for user d
$n^k_{(\cdot),i.j}$	number of times item sequence $i.j$ has been associated with topic k in $\mathbf{W}$
$n^k_{d,(\cdot)}$	number of times an item has been associated with topic k for user d
$\vec{\Theta}$	matrix of parameters $\vec{\theta}_u$
$\vec{\theta}_u$	mixing proportion of topics for the user u
$\vartheta_{u,k}$	mixing coefficient of the topic k for the user u
$\vec{\Phi}$	matrix of parameters $\vec{\varphi}_k = \{\varphi_{k,j.i}\}$
$\varphi_{k,j.i}$	mixing coefficient of the topic k for the item sequence j,i

Sec. 3, in which we measure the performance of the approaches in a recommendation scenario. Section 4 concludes the paper with a summary of the findings and mention to further extensions.

2 MODELING SEQUENCE DATA

Let $\mathcal{U} = \{u_1, \ldots, u_M\}$ be a set of M users and $I = \{i_1, \ldots, i_N\}$ a set of N items. In the general settings, we consider a set $\mathbf{W} = \{\vec{w}_1, \ldots, \vec{w}_M\}$ of user traces, where $\vec{w}_u = \{w_{u,1}, w_{u,2}, \cdots, w_{u,n_i-1}.w_{u,n_i}\}$ is the trace of all items selected by user u in sequence. We also assume that each user action is characterized by a latent factor triggering that action. That is, a latent set $\mathbf{Z} = \{\vec{z}_1, \ldots, \vec{z}_M\}$ is associated to the data, where, again $\vec{z}_u = \{z_{u,1}, z_{u,2}, \cdots, z_{u,n_i-1}.z_{u,n_i}\}$ is a latent topic sequence, and $z_{d,n} \in \{1, \ldots, K\}$ is the latent topic associated with the item $w_{d,n} \in I$. By assuming that $\vec{\Phi}$ and $\vec{\Theta}$ are the distribution functions for $\mathbf{W}$ and $\mathbf{Z}$ (with respective priors $\vec{\beta}$ and $\vec{\alpha}$, we can express the complete likelihood as:

$$P(\mathbf{W}, \mathbf{Z}, \vec{\Theta}, \vec{\Phi} | \vec{\alpha}, \vec{\beta}) = P(\mathbf{W}|\mathbf{Z}, \vec{\Phi}) P(\vec{\Phi}|\vec{\beta}) \cdot P(\mathbf{Z}|\vec{\Theta}) P(\vec{\Theta}|\vec{\alpha}) \quad (1)$$

where

$$P(\mathbf{W}|\mathbf{Z}, \vec{\Phi}) = \prod_{d=1}^{M} P(\vec{w}_d|\vec{z}_d, \vec{\Phi})$$

$$P(\mathbf{Z}|\vec{\Theta}) = \prod_{d=1}^{M} P(\vec{z}_d|\vec{\theta}_d)$$

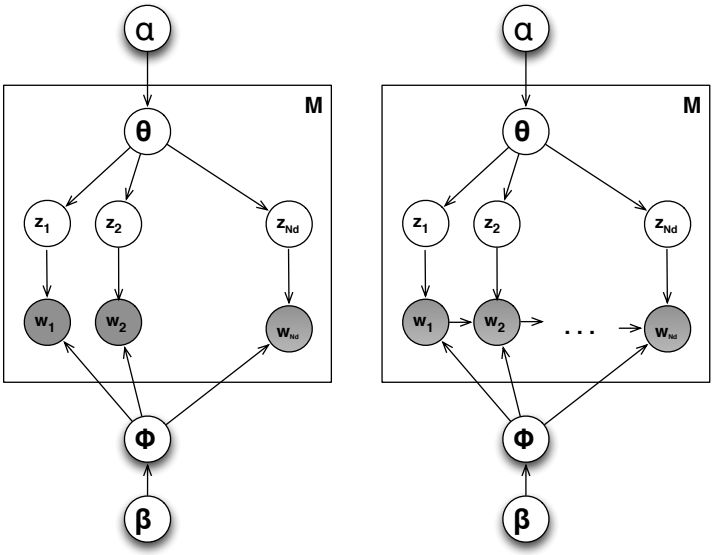
and $P(\vec{\Phi}|\vec{\beta})$ and $P(\vec{\Theta}|\vec{\alpha})$ are specified according to the modeling. For example, in the standard LDA settings where all terms are independent and exchangeable, we have:

$$\begin{aligned}
P(\vec{w}_d|\vec{z}_d, \vec{\Phi}) &= \prod_{i=1}^{n_d} P(w_{d,n}|z_{d,n}, \vec{\Phi}) \\
P(w|k, \vec{\Phi}) &= \prod_{i=1}^{N} \varphi_{k,i}^{\delta_{i,w}} \\
P(\vec{z}_d|\vec{\theta}_d) &= \prod_{i=1}^{n_d} P(z_{d,n}|\vec{\theta}_d) \\
P(z|\vec{\theta}_d) &= \prod_{k=1}^{K} \vartheta_{d,k}^{\delta_{k,z}} \\
P(\vec{\Theta}|\vec{\alpha}) &= \prod_{d=1}^{M} P(\vec{\theta}_d|\vec{\alpha}) \\
P(\vec{\theta}_d|\vec{\alpha}) &= \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \vartheta_{d,k}^{\alpha_k - 1} \\
P(\vec{\Phi}|\vec{\beta}) &= \prod_{k=1}^{K} P(\vec{\varphi}_k|\vec{\beta}_k) \\
P(\vec{\varphi}_k|\vec{\beta}_k) &= \frac{\Gamma(\sum_{i=1}^{N} \beta_{k,i})}{\prod_{i=1}^{N} \Gamma(\beta_{k,i})} \prod_{i=1}^{N} \varphi_{k,i}^{\beta_{k,i} - 1}
\end{aligned}$$

Here, $\delta_{h,k}$ represents the Kronecker delta. Figure 1(a) graphically describes the generative process. As usual, the joint topic-data probability can be obtained by marginalizing over the $\vec{\Phi}$ and $\vec{\Theta}$ components:

$$P(\mathbf{W}, \mathbf{Z}|\vec{\alpha}, \vec{\beta}) = \int_{\vec{\Phi}} \int_{\vec{\Theta}} P(\mathbf{W}|\mathbf{Z}, \vec{\Phi}) P(\vec{\Phi}|\vec{\beta}) P(\mathbf{Z}|\vec{\Theta}) \cdot P(\vec{\Theta}|\vec{\alpha}) d\vec{\Theta} d\vec{\Phi}$$

In the following, we model further assumptions on both w_d and z_d, which explicitly deny the exchangeability assumption. Several other models can be obtained, which rely on more complex assumptions. However, the models delved in here subsume the main characteristics of sequential modeling. We observed that, in the real world, past decisions affect future decisions. In particular we focused on the behavior of a user base which is used to frequently buy items from a provider. A user tend to choose items according her tastes, but her tastes change over the time influenced by the purchased items. The sequence of these items depends on the fact that nearly purchased items are similar or share some features. For instance, let us consider the sequence of items $u.v.t$: initially the user bought the item u, then she chose v because of its similarity to u and finally she acquired t, that shares some features with v. Note that t should be completely different from u, but because of the taste



(a) Latent Dirichlet Allocation (b) Token-Bigram Model

Figure 1: Graphical Models

change of the user they are in the same sequence. According to these assumptions, we choose to model the item sequence as a stationary Markov Chain of order 1:

- we choose to use a Markov Chain because of the sequential nature of the purchased item list, moreover the Markov Chain can model the user's taste changing over the time;
- the chain is stationary because users frequently buy items;
- the order of the chain is 1 because the probability that two subsequent purchases share some features or are dependent each other is higher than that of two purchases distant in time.

All these aspects lead us to the definition of the *Token-Bigram Model*, described as follows. We assume that $\vec{w}_d$ represents a first-order Markov chain, where, each item selection $w_{d,n}$ depends on the recent history $w_{d,n-1}$ of selections performed by the user. This is essentially the same model proposed in (Wallach, 2006; Cadez et al., 2000), and the probability of a user trace can be expressed as

$$P(\vec{w}_d|\vec{z}_d,\vec{\Phi}) = \prod_{n=1}^{N_d} P(w_{d,n}|w_{d,n-1},z_{d,n},\vec{\Phi}) \quad (2)$$

In practice, an item $w_{d,n}$ is generated according to a multinomial distribution $\vec{\phi}_{z_{d,n},w_{d,n-1}}$ which depends on both the current topic $z_{d,n}$ and the previous items $w_{d,n-1}$. (Notice that when $n = 1$, the previous item is empty and the multinomial resolves to $\vec{\phi}_{z_{d,n}}$, representing the initial status of a Markov chain). As a consequence, the conjugate prior has to be redefined as:

$$\begin{aligned} P(\vec{\Phi}|\vec{\beta}) &= \prod_{k=1}^{K}\prod_{m=0}^{N} P(\vec{\varphi}_{k,m}|\vec{\beta}_{k,m}) \\ &= \prod_{k=1}^{K}\prod_{m=0}^{N} \frac{\Gamma(\sum_{n=1}^{N}\beta_{k,m,i})}{\prod_{n=1}^{N}\Gamma(\beta_{k,m,n})}\prod_{n=1}^{N}\varphi_{k,m,n}^{\beta_{k,m,n}-1} \end{aligned}$$

Since the Markovian process does not affect the topic sampling, both $P(\vec{z}_d|\vec{\theta}_d)$ and $P(\vec{\Theta}|\vec{\alpha})$ are defined as in equation 2. The generative model, depicted in Fig. 1(b), can be described as follows:

- For each user $d \in \{1,\ldots,M\}$ sample user community-mixture components $\vec{\theta}_d \sim Dirichlet(\vec{\alpha})$ and sequence length $n_d \sim Poisson(\xi)$
- For each user attitude $k \in 1,\ldots,K$ and item $v \in \{0,\ldots,N\}$
  - Sample item selection components $\vec{\varphi}_{k,v} \sim Dirichlet(\vec{\beta}_{k,v.})$
- For each user $d \in \{1,\ldots,M\}$ and $n \in \{1,\ldots,n_d\}$
  - sample a user attitude $z_{d,n} \sim Discrete(\vartheta_u)$
  - sample an item $i_{d,n} \sim Discrete(\vec{\varphi}_{z_{d,n},i_{d,n-1}})$

Notice that we explicitly assume the existence of a family $\{\vec{\beta}_{k,m}\}_{k=1,\ldots,K;m=0,\ldots,N}$ of Dirichlet coefficients. As shown in (Wallach, 2006), different mod-

eling strategies (e.g., shared priors $\beta_{(k,m),n} = \beta_n$) can affect the accuracy of the model.

By algebraic manipulations, we obtain the following joint item-topic distribution:

$$P(\mathbf{W},\mathbf{Z}|\alpha,\beta) = \left(\prod_{d=1}^{M} \frac{\Delta\left(\vec{n}_{d,(\cdot)} + \vec{\alpha}\right)}{\Delta(\vec{\alpha})}\right) \cdot \left(\prod_{k=1}^{K}\prod_{m=0}^{N} \frac{\Delta\left(\vec{n}^{k}_{(\cdot),m.} + \vec{\beta}_{k,m.}\right)}{\Delta(\vec{\beta}_{k,m.})}\right) \tag{3}$$

The latter allows us to define a collapsed Gibbs sampling procedure:

E step: iteratively sampling of topics, according to the probability

$$P(z_{d,n} = k|\vec{Z}_{-(d,n)},\vec{W}) \propto \left(n^{k}_{d,(\cdot)} + \alpha_k - 1\right) \cdot \frac{n^{k}_{(\cdot),u.v} + \beta_{k,u.v} - 1}{\sum_{r=1}^{N} n^{k}_{(\cdot),u.r} + \beta_{k,u.r} - 1} \tag{4}$$

relative to the topic to associate with the n-th item of the d-th document, exhibiting $w_{d,n-1} = u$ and $w_{d,n} = v$.

M Step: estimating both $\vec{\Phi}$ and $\vec{\Theta}$, according to the following equations:

$$\vartheta_{d,k} = \frac{n^{k}_{d,(\cdot)} + \alpha_k}{\sum_{k'=1}^{K} n^{k'}_{d,(\cdot)} + \alpha_{k'}}$$
$$\varphi_{k,r.s} = \frac{n^{k}_{(\cdot),r.s} + \beta_{(k,r.s}}{\sum_{s' \in U} n^{k}_{(\cdot),r,s'} + \beta_{(k,r).s'}} \tag{5}$$

Log-Likelihood. The data likelihood, given the model parameters, $\vec{\Theta},\vec{\Phi}$, is defined as follows:

$$P(\mathbf{W}|\vec{\Theta},\vec{\Phi}) = \prod_{d=1}^{M} P(\vec{w}_d|\vec{\theta}_d,\vec{\Phi}) \tag{6}$$

Where:

$$\begin{aligned} P(\vec{w}_d|\vec{\theta}_d,\vec{\Phi}) &= P(w_{d,n_d},\ldots,w_{d,1}|\vec{\theta}_d,\vec{\Phi}) \\ &= \sum_{k=1}^{K} P(w_{d,n_d},\ldots,w_{d,1}|z_{d,n_d} = k,\vec{\theta}_d,\vec{\Phi}) \\ &\cdot P(z_{d,n_d} = k|\vec{\theta}_d) \\ &= \sum_{k=1}^{K} P(w_{d,n_d}|z_{d,n_d} = k, w_{d,n_d},\vec{\Phi}) \\ &\cdot P(w_{d,n_d-1},\ldots,w_{d,1}|\vec{\theta}_d,\vec{\Phi}) \\ &\cdot P(z_{d,n_d} = k|\vec{\theta}_d) \\ &= P(w_{d,n_d-1},\ldots,w_{d,1}|\vec{\Theta},\vec{\Phi}) \\ &\cdot \sum_{k=1}^{K} \varphi_{k,w_{d,n_d},w_{d,n_d-1}} \cdot \vartheta_{d,k} \end{aligned} \tag{7}$$

This formulation triggers a recursive procedure for the likelihood computation, whose trivial case is:

$$\begin{aligned} P(w_{d,1}|\vec{\Theta},\vec{\Phi}) &= \sum_{k=1}^{K} P(w_{d,1}|,z_{d,1} = k,\vec{\Phi}) P(z_{d,1} = k|\vec{\theta_d}) \\ &= \sum_{k=1}^{K} \phi_{k,w_{d,1},} \cdot \vartheta_{d,k} \end{aligned} \tag{8}$$

Where $\vec{\phi}_k$ represents the initial state probabilities, as introduced above.

2.1 Item Ranking

The probabilistic framework is quite flexible, as it provides in general different choices for item ranking (Barbieri and Manco, 2011) an item for recommendation purposes. We next propose the functions relative to each model to be tested in the experimental section. In the following, we assume that a user can be denoted by a unique index u, and a previous history is given by $\vec{w}_u$ of size $n-1$. We are interested in providing a ranking for the n-th choice $w_{u,n}$.

LDA. Following (Barbieri and Manco, 2011) we adopt the following ranking function:

$$\begin{aligned} rank(i,u) &= P(w_{u,n} = i|\vec{w}_u) \\ &= \sum_{k=1}^{K} P(i|z_{u,n} = k) P(z_{u,n} = k|\vec{\theta}_u) \\ &= \sum_{k=1}^{K} \varphi_{k,i} \cdot \vartheta_{u,k} \end{aligned} \tag{9}$$

It has been shows that LDA, equipped with the above ranking function, significantly outperforms

the most significant approaches to modeling user preferences. Hence, it is a natural baseline function upon which to measure the performance of the other approaches proposed in this paper.

Token-Bigram Model. The dependency of the current selection from the previous history can be made explicit, thus yielding the following upgrade to the LDA ranking function:

$$\begin{aligned} rank(i,u) &= P(w_{u,n} = i|\vec{w}_u) \\ &= \sum_{k=1}^{K} P(i|z_{u,n} = k, \vec{w}_u) P(z_{u,n} = k|\vec{\theta}_u) \\ &= \sum_{k=1}^{K} P(i|z_{u,n} = k, w_{u,n-1}) \vartheta_{u,k} \\ &= \sum_{k=1}^{K} \varphi_{k,j,i} \cdot \vartheta_{u,k} \end{aligned} \tag{10}$$

where $j = w_{u,n-1}$ is the last item selected by user u in her current history.

3 EXPERIMENTAL EVALUATION

In this section we present an empirical evaluation of the proposed models which focuses on the recommendation problem. Given the past observed preferences of a users, the goal of a recommender systems (RS) is to provide her with personalized (and contextualized) recommendations about previously non-purchased items that meet her interest. Note that, although usually the standard benchmarks for evaluating recommendations are Movielens and Netflix data, they do not guarantee that the timestamp associated with each pair $\langle user, item \rangle$ corresponds to the timestamp of the effective purchase of the item, since the timestamp refers to the rating and the user may specify ratings in a different order. Moreover, we cannot rely on Videolectures data because, due to the privacy preserving constraints, this dataset do not provide user profiles but pooled statistics. We choose to evaluate the performances of the proposed techniques by measuring their predictive capabilities on two datasets, namely *Iptv1* and *Iptv2*. These data have been collected by analyzing the pay-per-view movies purchased by the users of two European IPTV providers over a period of several months (Cremonesi and Turrin, 2009; Bambini et al., 2011). The original data have been preprocessed by firstly removing users with less of 10 purchases the items with less then the same operation was performed over the items. We perform a chronological split of the data by including in the test set the last 20% purchases of each user. The main features of the datasets are summarized in Tab. 2. For each dataset, the users and items, in the test data, are subsets of the users and items within the training data. The sparseness factors of Iptv1 are 97.5% and 99.3% for the training and test sets (resp.), while the ones for Iptv2 are 99.3% and 99.8% (training and test sets, resp.). These values highlight the difficulty in discovering patterns and regularities within the data, in other words it's hard to define a good model for the recommendation. Fig. 2 and Fig. 3 show the distribution of the users and the bigrams (resp.) for both the datasets. As can be seen, these distributions exhibit the trend of power-laws (Clauset et al., 2007).

	IPTV1		IPTV2	
	Training	Test	Training	Test
Users	16,237	16,153	64,334	63,878
Items	759	731	2802	2777
Evaluations	314,042	78,557	1,224,790	306,271
Avg # evals (user)	19	5	19	5
Avg # evals (item)	414	107	437	110
Min # evals (user)	4	1	4	1
Min # evals (item)	5	1	5	1
Max # evals (user)	252	15	497	17
Max # evals (item)	2284	1527	9606	3167

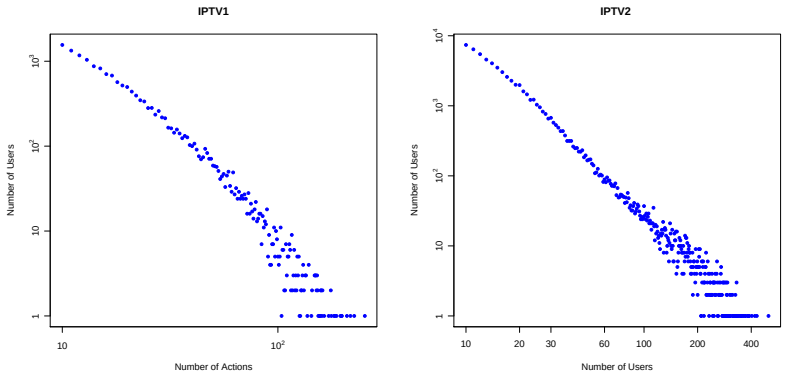
Table 2: Summary of the evaluation data.

Testing protocol. Given an active user u and a context c_u currently under examination, the goal of a RS is to provide u with a recommendation list $\mathcal{R}$, picked from a list $\mathcal{C}$ of candidates, that are expected to be of interest to u. This clearly involves predicting the interest of u into an item according to c_u. We review here the evaluation metrics and the testing protocols to be used on this purpose.

In general, a recommendation list $\mathcal{R}$ can be generated as follows:

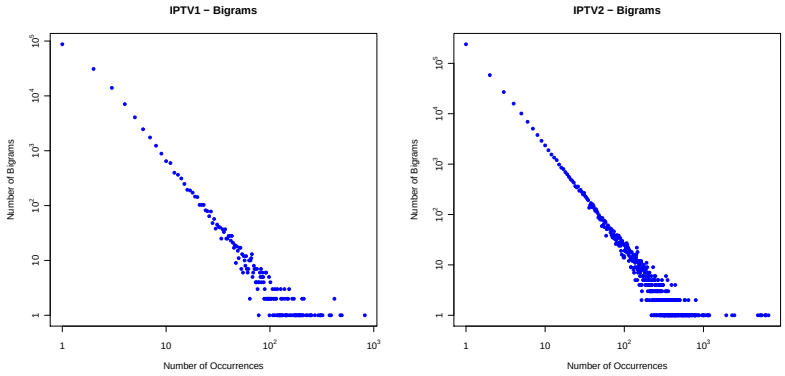
- Let $\mathcal{C}$ be a set of d candidate recommendations to arbitrary items;
- Associate each item $i \in \mathcal{C}$ with a score p^i_{u,c_u} representing u's interest into i in accordance to context c_u.
- Sort $\mathcal{C}$ in descending order of item scores p^i_{u,c_u};
- Add the first k items from $\mathcal{C}$ to $\mathcal{R}$ and return the latter to user u.

A common framework in the evaluation of the predictive capabilities of a RS algorithm is to split the traces $\mathbf{W}$ into two subsets $\mathbf{T}$ and $\mathbf{S}$, such that the former is used to train the RS, while the latter is used



(a) (b)

Figure 2: Distribution of the number of evaluation per user on both datasets



(a) (b)

Figure 3: Distribution of the number of bigrams on both datasets

for validation purposes. Here, for a given user, c_u can be defined according to the technique under examination: the set of previously unseen items for the LDA, or the most recent preference for the Token-Bigram model.

In the latter model, it is required that all sequences in **T** precede those in **S**, in order to provide a fair simulation of real-life scenarios. As a consequence, for a given user u, the trace $\vec{w}_u$ can be split into $\vec{w}_u^{(T)}$ and $\vec{w}_u^{(S)}$, representing the portions of the sequence belonging to **T** and **S**, respectively. By selecting a user u, the set $\mathcal{C}$ of candidate recommendations are evaluated assuming $\vec{w}_u^{(T)}$ part of the context. The recommendation list $\mathcal{R}$ for u is then formed by following the foregoing generation process and the accuracy of $\mathcal{R}$ is ultimately assessed through a comparison with the items appearing in $\vec{w}_u^{(S)}$. Therein, the standard classification-based metrics, i.e., precision and recall, can be adopted to evaluate the recommendation accuracy of $\mathcal{R}$.

The latter can be defined according to an adaptation of the testing protocol defined in (Cremonesi et al., 2010).

- For each user u and for each item $i \equiv w_{u,n}$ relative to a position n of $\vec{w}_u^{(T)}$:
  - Generate the candidate list $\mathcal{C}$ by randomly drawing from $I - \{i\}$.
  - Add i to $\mathcal{C}$.
  - Associate each item $j \in \mathcal{C}$ with the score $rank(i,u)$ and sort $\mathcal{C}$ in descending order of item scores.
  - Consider the position of the item i in the ordered list: if i belongs to the top-k items, there is a *hit*; otherwise, there is a *miss*.

By definition, recall for an item can be either 0 (in the case of a failure) or 1 (in the case case of a hit). Likewise, precision can be either 0 (in the case of a failure) or $\frac{1}{k}$ (in the case of a hit). The overall

precision and recall are defined in (Cremonesi et al., 2010) as the below averages:

$$Recall(k) = \frac{\#hits}{|\mathbf{T}|}$$

$$Precision(k) = \frac{\#hits}{k \cdot |\mathbf{T}|} = \frac{recall(k)}{k}$$

A key role in the process of generating accurate recommendation lists is played by the schemes with which to rank items candidate for recommendation. (Barbieri and Manco, 2011) provides a comparative analysis of three possible such schemes, and studies their impact in the accuracy of the recommendation list. It is worth noting that the score $rank(i,u)$ proposed here follows the main findings in that paper.

Also, (Barbieri and Manco, 2011) shows that item selection plays the most important role in recommendation ranking. As a matter of fact, LDA turns out to be the model that best accommodates item selection in recommendation ranking, thus providing the best recommendation accuracy according to the above described protocol. It is natural hence to compare the Token-Bigram model proposed in this paper with the LDA approach.

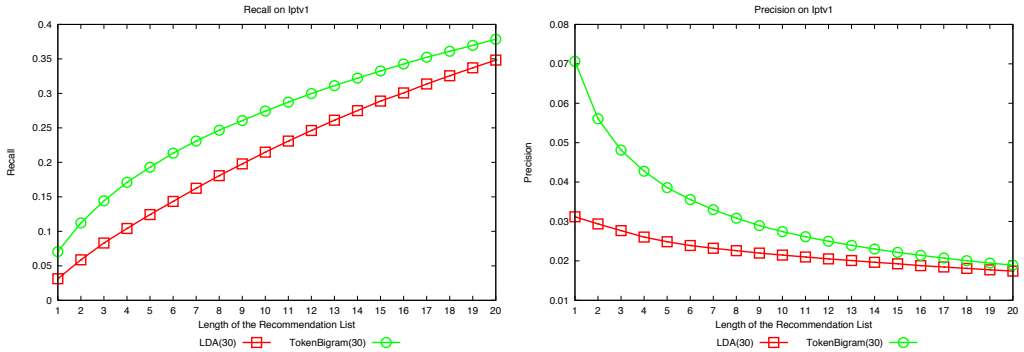
Implementation details. All the considered model instances were run varying the number of topics within the range $[3,20]$. We perform 5000 Gibbs Sampling iterations, discarding the first 1000 (burn in period), and with a sample lag of 30.
Our implementations are based on asymmetric Dirichlet prior over the document-topic distributions (this modeling strategy has reported to achieve important advantages over the symmetric version (Wallach et al., 2009)), while we employ a symmetric prior over the topic distributions. For the LDA and token-bigram models we adopted the procedure for updating the prior $\vec{\alpha}$ as described in (Heinrich, 2008; Minka, 2000). We set the length of the candidate random list (see the testing protocol) equal to about the 35% of the dimension of the item sets for each test set. Precisely, these lists have 250 items for Iptv1 and 1000 items for Iptv2.

Results. In Fig. 4 we summarize the best results in recommendation accuracy achieved by the proposed approach, over the two considered datasets. For each model, the number of topics which leads to the best results is given in brackets. On both datasets, the Token-Bigram models outperform the LDA models, both in recall and precision. At high level, these results suggest that exploiting the previous contextual information, the Token-Bigram Model outperforms LDA in recommendation accuracy. While the ranking function employed for LDA takes into account only the probability of selecting an item given the whole user purchase-history and the whole topic space, the Token-Bigram approach focuses on a region of the topic space determined by considering the previous item, thus providing a better estimate of the selection probabilities for the next user's choice.

In order to assess the stability of the proposed approaches in varying the number of topics, we plot in Fig. 5 and Fig. 6 the recall and the precision (respectively) achieved when the length of the recommendation list is 20. Considering Iptv1, best results are achieved by both the techniques exploiting the largest number of topics we used for the experimentation, 30, with a recall of 0.347 and a precision of 0.017 for LDA and a recall of 0.379 and a precision of 0.019 for the Token Bigram, with a difference in recall of 0.032. For Iptv2, LDA achieves its maximum one again with 30 topics, while the proposed model has the best quality with 5 topics. LDA achieve a recall and a precision of 0.512 and 0.026 (resp.), while the Token Bigram has 0.556 for recall and 0.028 for precision, with a difference in recall of 0.044. It's interesting to note that the performances of the TokenBigram do not change substantially varying the number of topics. The results presented above experimentally prove the effectiveness of sequence-based topic models in modeling and predicting future users' choices. However those models increase significantly the number of parameters to be learned and this implies an increase in the learning time. In Fig. 7 we plot the learning time (5000 Gibbs Sampling iterations) for different numbers of topics. The learning time is consequently considerably larger. This is mainly due to the larger number of hyperparameters ($K \times K$ vs K) and to the complexity of the α-update iterative procedure.

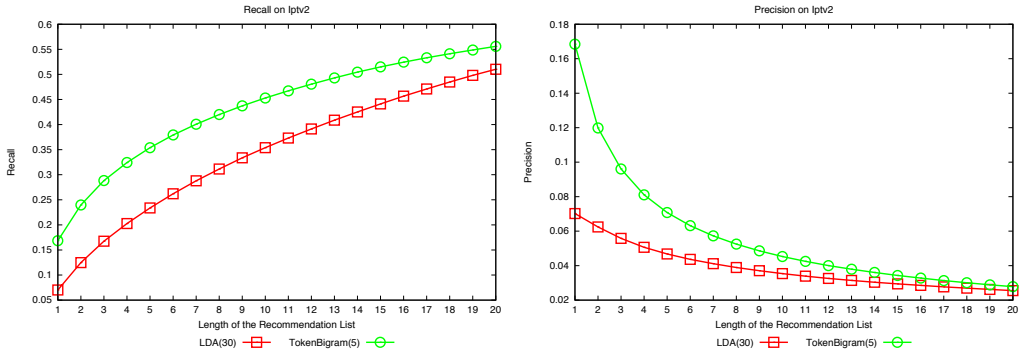
4 CONCLUSION AND FUTURE WORKS

In this paper we proposed an extension of the LDA model. The proposed model relaxes the bag-of-words assumption of LDA, assuming that each token, not only depends on a number of latent factors, but also on the previous token. The set of dependencies has been modeled as a stationary Markov chain, which led us to define a procedure for estimating the model parameters, exploiting the Gibbs Sampling. This model better suites a framework for modeling context in a recommendation setting than LDA, since it takes into account the information about the token sequence. The experimental evaluation, over two real-world datasets expressing sequence infor-



(a)

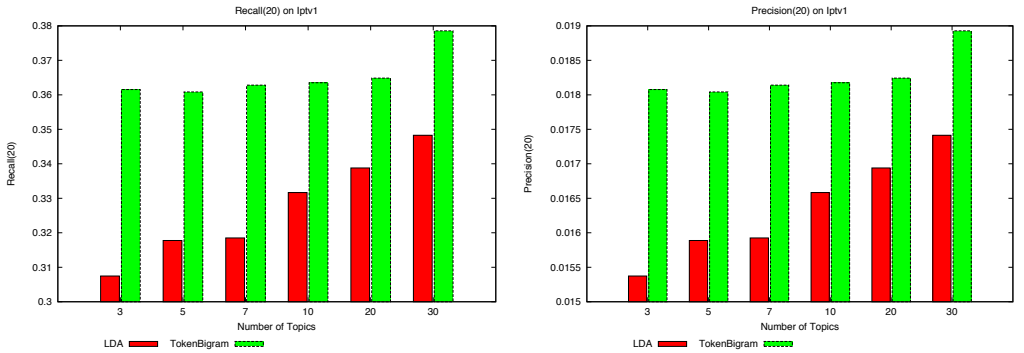
(b)



(c)

(d)

Figure 4: Recommendation accuracy: LDA vs Sequence-based Approaches



(a)

(b)

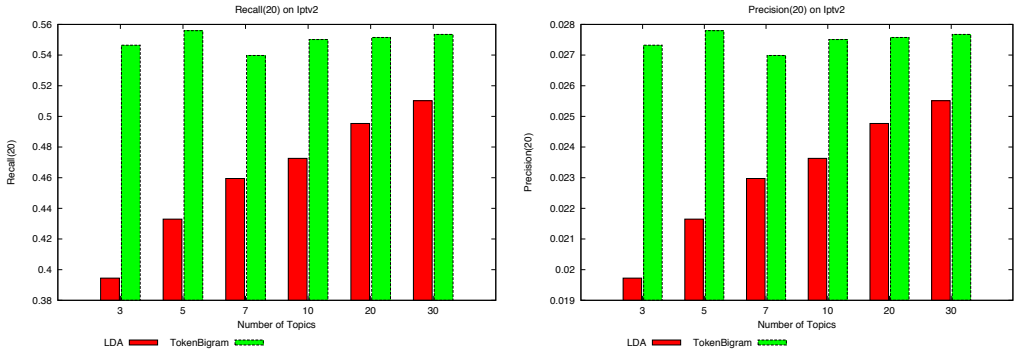
Figure 5: Recall(20) of the considered approaches varying the number of topics on IPTV1

mation, shows that the proposed model outperforms LDA at the expense of an higher execution time when the number of the latent topics is large, since the number of parameters to estimate is bigger than in LDA. In the future we are going to investigate more kinds of Markov chains expressing the sequence of the tokens, moreover we have the intention of improving the proposed model by considering side information such as tags or comments over tokens.

REFERENCES

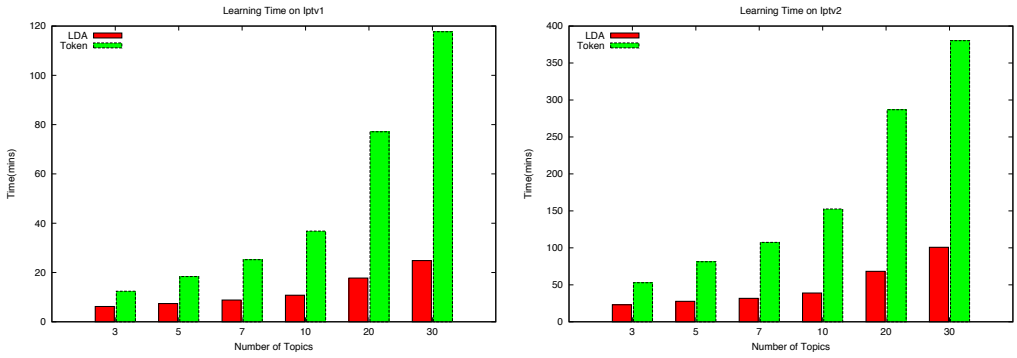
Bambini, R., Cremonesi, P., and Turrin, R. (2011). A recommender system for an iptv service provider: a real large-scale production environment. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 299–331. Springer.

Barbieri, N., Costa, G., Manco, G., and Ortale, R. (2011a). Modeling item selection and relevance for accurate recommendations: a bayesian approach. In *Proc. RecSys*, pages 21–28.



(a) (b)

Figure 6: Precision(20) of the considered approaches varying the number of topics on IPTV2



(a) (b)

Figure 7: Learning time of the models

Barbieri, N. and Manco, G. (2011). An analysis of probabilistic methods for top-n recommendation in collaborative filtering. In *Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part I*, ECML PKDD'11, pages 172–187.

Barbieri, N., Manco, G., Ortale, R., and Ritacco, E. (2011b). Balancing prediction and recommendation accuracy: Hierarchical latent factors for preference data. In *Proc. SDM'12*.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Cadez, I., Heckerman, D., Meek, C., Smyth, P., and White, S. (2000). Visualization of navigation patterns on a web site using model-based clustering. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 280–284.

Clauset, A., Shalizi, C., and Newman, M. E. J. (2007). Power-law distributions in empirical data. *SIAM Reviews*.

Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *ACM RecSys*, pages 39–46.

Cremonesi, P. and Turrin, R. (2009). Analysis of cold-start recommendations in iptv systems. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 233–236. ACM.

Griffiths, T. L., Steyvers, M., and Tenenbaum, J. B. (2007). Topics in semantic representation. *Psychological Review 114*.

Gruber, A., Weiss, Y., and Rosen-Zvi, M. (2007). Hidden topic markov models. *Journal of Machine Learning Research*, 2:163–170.

Heinrich, G. (2008). Parameter Estimation for Text Analysis. Technical report, University of Leipzig.

Minka, T. P. (2000). Estimating a Dirichlet distribution. Technical report, Microsoft Research.

Wallach, H., Mimno, D., and McCallum, A. (2009). Rethinking lda: Why priors matter. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981.

Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 977–984.

X. Wang, A. M. and Wei, X. (2007). Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Procs. ICDM'07*, pages 697–702.