Context-aware LDA: Balancing Relevance and Diversity in TV Content Recommenders

Jing Yuan[1], Fikret Sivrikaya[1], Frank Hopfgartner[2], Andreas Lommatzsch[1], Mu Mu[3]
[1] DAI-Lab, Berlin, Germany – {jing.yuan, fikret.sivrikaya, andreas.lommatzsch}@dai-labor.de
[2] University of Glasgow, Glasgow, UK – frank.hopfgartner@glasgow.ac.uk
[3] Lancaster University, Lancaster, UK – m.mu@lancaster.ac.uk

ABSTRACT

In the vast and expanding ocean of digital content, users are hardly satisfied with recommended programs solely based on static user patterns and common statistics. Therefore, there is growing interest in recommendation approaches that aim to provide a certain level of diversity, besides precision and ranking. Context-awareness, which is an effective way to express dynamics and adaptivity, is widely used in recommender systems to set a proper balance between ranking and diversity. In light of these observations, we introduce a recommender with a context-aware probabilistic graphical model and apply it to a campus-wide TV content delivery system named "Vision". Within this recommender, selection criteria of candidate fields and contextual factors are designed and users' dependencies on their personal preference or the aforementioned contextual influences can be distinguished. Most importantly, as to the role of balancing relevance and diversity, final experiment results prove that context-aware LDA can evidently outperform other algorithms on both metrics. Thus this scalable model can be flexibly used for different recommendation purposes.

CCS Concepts

•**Information systems** → **Recommender systems;** *Personalization;*

General Terms

Algorithms, Theory, Experimentation.

Keywords

TV recommender, context-aware, Latent Dirichlet Allocation, ranking, diversity.

1. INTRODUCTION

IP-based TV providers strive to empower their users with effective recommender systems in order to eliminate the users' perplexity in discovering and selecting content among the immense set of options and to increase their interactivity with the system [12]. For instance, association rule mining was used in YouTube [8], matrix factorization has been tried in TV1 and TV2 datasets [6], collaborative filtering was chosen for an Australian IPTV provider [18], among many others.

Workshop RecSysTV'15, in conjunction with RecSys'15,
Sept. 19th, 2015, Vienna, Austria

This paper utilizes the web-based TV delivery system "Vision" [1], serving the campus users at Lancaster University. Apart from users' viewing behavior, we exploit information about the TV programs' genres, channels, performers, and program names as provided by The Electronic Program Guide (EPG). Yet how to select the optimal sources for providing *customized recommendations* is an open research question. We have thus performed pair-comparison of normalized entropy between users' choices on *genres*, *channels*, and *programs* to solve this problem and found that *program* is the most representative field to express users' preferences.

It is commonly recognized in the literature that contextual factors, including timing, location, company by other people, etc., help observe users' dynamically changing preferences [1, 16, 20, 7, 10]. Therefore, we introduce contextual factors into our recommender to increase the dynamics and diversity of recommendation results. In our scenario, facing the accessible contextual factors of "live/VoD environment", "time of day" and "day of week", the determination of which factors to use is another non-trivial task. By analyzing the programs' significant difference w.r.t. these factors, we consider "live/VoD environment" and "time of day" to study the role of context in this scenario.

Taking these two contextual factors into account, we choose the extended topic model of LDA to design a recommender. Topic model was first proposed in the text retrieval area, yet its powerful properties of dimension reduction and hidden topic generation made it popular in the recommender field as well. For example, usage of Latent Semantic Index (LSI) has been discussed in an IPTV provider system [4], Latent Dirichlet Allocation (LDA) has been attempted in online course recommender [3], questioning community recommender [17], and spatial item recommender [20]. In this paper, based on user-program matrix and two contextual factors "live/VoD environment" and "time of day", we realize recommenders using pure user-oriented LDA and context-aware LDA, and demonstrate that context-aware LDA provides apparent performance gains on both ranking and diversity metrics.

The main contributions of this work can be summarized as follows.

- A recommender algorithm customized for a specific TV content provider based on the analysis of the dataset, through which the proper data fields and suitable contextual factors can be determined.
- Extension of the topic model LDA to a contextual scal-

[1] http://vision.lancs.ac.uk/

able model, where other contextual factors could be also introduced in the future.

- Comparison of algorithms on both *ranking oriented* metrics and *diversity oriented* metrics, showcasing the dominant performance of context-aware LDA model on both measurements.

The rest of the paper is structured as follows. Section 2 introduces the concrete steps of customized analysis on our "Vision" dataset, followed by the implementation procedure of user-oriented LDA, context-aware LDA and queries in Section 3. Experiments are presented in Section 4 to demonstrate the effect of the algorithms on both ranking and diversity measurements. Section 5 concludes the article with discussion of open points and plans for future research.

2. CUSTOMIZED ANALYSIS

Just as feature selection is a vital procedure in the machine learning pipeline, how to appropriately benefit from the rich EPG data is an intractable issue for IP-based TV recommenders as well. In this section, based on the dataset condition of the "Vision" system, we propose a pre-analysis on several candidate fields and different contextual factors, through which program is chosen as the training field and "time of day" and "live/VoD environment" are decided as contextual factors in the following algorithms.

2.1 Dataset Description

Our recommender scenario is a campus IPTV provider which mainly serves university students with standard TV content. The dataset used in this paper is comprised of one year transactions (from October 1th, 2013 to October 1th, 2014) recorded by the provider. We apply two criteria on the raw dataset to filter some meaningless transactions out:

- Playback duration less than 15 seconds and playback percentage less than 15%. A short viewing duration typically denote users' short hesitation on a program rather than their true interests, only relatively stable viewing behavior is maintained to represent users' preferences.

- Users with total transaction number less than 10. Too small transaction number means low interactivity of a user. With such sparse data, it would be also difficult to meaningfully separate the training set, cross validation set and test set for those specific users. Thus, we simply ignore such users in this study.

After filtering, there are a total of 587 users and 119,435 transactions remaining. 16 genres, 33 channels and 7,065 different program names had been recorded among these transactions. Fields as "time of day" and "day of week" were recorded in categorical counting number form as 0-23 and 0-6 respectively for each transaction. In addition to the users' interactions, we annotated the programs that were viewed with detailed EPG data (synopsis, performers and so on).

2.2 EPG Field Determination

As introduced in above subsection, genres, channels and program names are all accessible fields of a transaction. Yet which of them can be chosen as reasonable training input needs to be carefully considered. Before the recommender algorithm design, we make use of normalized entropy to distinguish the usefulness of different fields. The definition of normalized entropy nH can be formalized as

$$nH(X) = -\sum_{i}^{n} \frac{p(x_i)\log_b(p(x_i))}{\log_b(n)} \tag{1}$$

where $H(X) = -\sum_i^n p(x_i)\log_b(p(x_i))$ is the Shannon entropy of n possible states $\{x_1, x_2, ..., x_n\}$, and $p(x_i)$ is the occurrence probability of x_i. Specifically, $\log_b(n)$ is the maximum entropy when dealing with n states given the base b, which is also called the information length of n. To make entropy of spaces with different n comparable, normalized entropy $nH(X)$, i.e., $H(X)$ divided by information length $\log_b(n)$, was proposed. With $nH(X)$, the uncertainty of any space or system can be scaled into the range between 0 and 1 [15]. Analogous to the role of entropy as measuring the certainty in information theory, we make use of normalized entropy to measure users' propensity of being personalized on a specific feature. The greater the normalized entropy is, the more uncertainty the space has, thus the less propensity of personalization this feature shows. In other words, the field with smaller average normalized entropy indicates more representative of personalization in the recommender.

As depicted in Fig. 1, we compare normalized entropy performance on genres, channels and programs from the perspectives of users' mean and the general system, respectively. Despite the normalized entropies of the three fields regarding the general system, represented as blue bars, are all very close to 1, which means that choices are almost evenly distributed. On the other hand, users' individual normalized entropy of three fields, shown as boxplots in Fig. 1, can be clearly distinguished. Since inequality $nH_{genres} > nH_{channels} > nH_{programs}$ evidently holds in terms of individual users, we choose *programs* as the data field to represent users' preferences.

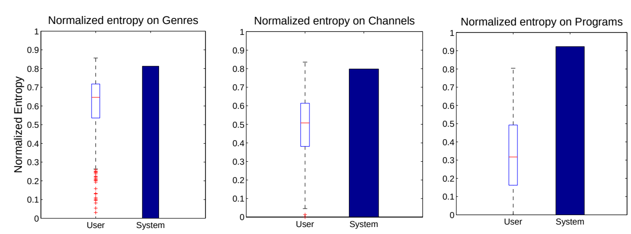

Figure 1: Entropy Comparison on Different Fields

2.3 Contextual Factor Analysis

After the selection of program information as the target field, we analyze the available contextual factors of "live/VoD environment", "time of day", and "day of week" regarding programs to decide on which ones to integrate in the recommender.

First, we observe from the statistics on "live/VoD environment" over the hottest 20 programs that programs' chosen frequency on live and VoD varies from each other significantly, as Fig. 2 depicts. For instance, even though "The Big Bang Theory" and "Frasier" are both comedies, "Frasier" proportionally acquires more attention in the environment of VoD than "The Big Bang Theory" does. Hence, we consider "live/VoD environment" as a candidate contextual factor.

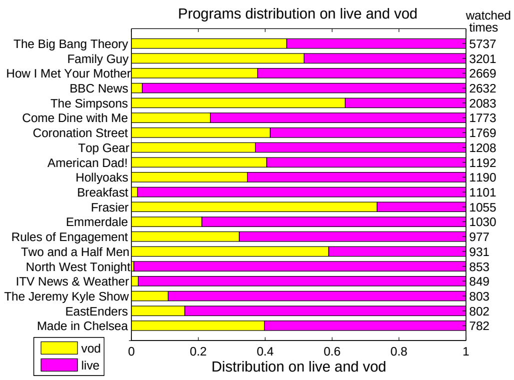

Figure 2: Programs Statistic on Live and VoD

There are two kinds of categorized timing granularity we can obtain from the dataset: one is "day of week" and the other is "time of day". To compare the programs' distribution varying under these two timing indicators, we calculate the normalized entropy again for each program in live or VoD environment in this subsection. From Fig. 3, we see that the inequality $nH_{dow} > nH_{tod}$ (nH_{dow} and nH_{tod} denotes normalized entropy on "day of week" and "time of day" separately) holds for almost every program, and that nH_{dow} is always close to 1. Since "time of day" apparently plays a more dominant role in pattern certainty, in spite of its less obvious performance under "VoD" condition, this factor still deserves being chosen as a second contextual factor in recommender, while "day of week" is tentatively excluded.

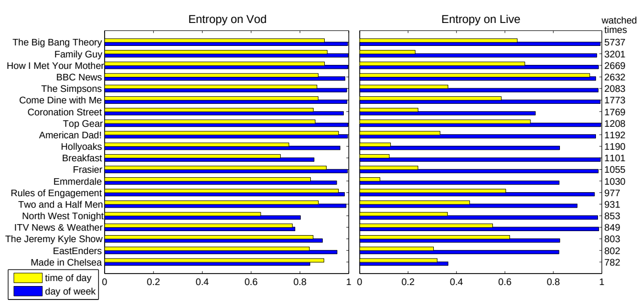

Figure 3: Entropy of Programs w.r.t. Different Categorical Timing

Having two contextual factors "live/VoD environment" and "time of day" to consider in recommender can be a quite challenging task. Even though state-of-art context-awareness handlers like SVD-feature and tensor decomposition have been successfully attempted in recommenders, users' dependencies on their own preference and contextual factors can't be distinguished from these models. Hence scalable probabilistic topic model is introduced in the following section, which can help us solve both context involvement and users' dependencies analysis problems.

3. RECOMMENDER DESIGN

In order to illustrate the influential role of contextual factors in our scenario, we make use of Latent Dirichlet Allocation (LDA), an extendable topic model, to build the algorithm. In this section, from user-oriented LDA to extended context-aware LDA, we show the variation of probabilistic graphical models, and list the Gibbs Sampling formulas for both of them. In addition, corresponding query procedure is also presented.

3.1 User-oriented LDA

Latent Dirichlet Allocation (LDA) is a well-defined generative topic model proposed in the information retrieval area. It can solve the problems as over fitting resulting from estimated parameters' linear growth and the unpredictability of unseen documents [5]. There have already been multiple transformations of LDA applied in recommender systems [17, 20], yet the basic idea behind these applications still heavily relies on the term frequency (TF) and document matrix, i.e., text description of items, whereas in this paper, combined with the analysis of the last section, we make use of users and programs matrix to describe the generative procedure.

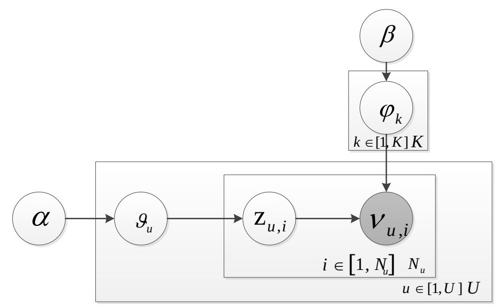

Figure 4: User-oriented LDA Graphical Model

Given the similar structure of user-program matrix and document-term matrix, we apply LDA here to infer users' preferences distribution on hidden topics. As depicted in Fig. 4, each user is treated as an outer plate and each transaction is treated as an inner plate. $v_{u,i}$, the only observed variable in the graphical model, represents the i_{th} item viewed by user u. $z_{u,i}$ indicates the topic assigned to $v_{u,i}$, which is also the sampling target in the sampling period. Symbols and their explanations used in this paper are listed in Table 1, where this user-oriented and the following contextual LDA share most symbols and definitions. Owing to the analogous data structure, we generally follow the classical symbols' definition in [11], and tune them slightly in accordance with our specific scenario.

3.2 Context-aware LDA

From the user-oriented LDA model, we can obtain users' individual preference, yet contextual variation cannot be reflected. Considering the extensibility of probabilistic graphical model [20], we extend the original LDA mentioned in last subsection to a context-aware LDA. Fig. 5 shows us the structure of this generating procedure. The probabilistic paths $\alpha \to \vartheta_u \to z_{u,i} \to v_{u,i}$ and $\beta \to \varphi_k \to v_{u,i}$ are the same as in user-based LDA, while the difference is reflected as the appearance of parts "User Inclination on Factors", "Time of Day Context" and "Live/VoD Environment". Among them, "user inclination" is the indicator that distinguishes users' propensity on three factors: "his/her own preference", "time of day context" and "live/VoD environment". Since the dependence on personal preference and contextual factors can be individually different [9, 21], besides $z_{u,i}$, another hidden variable $s_{u,i}$ needs to be sampled in this model

Table 1: Symbols and Notations

Symbols	Notations
$\alpha, \alpha^{c1}, \alpha^{c2}$	symmetric hyper parameter, Dirichlet prior of $\vec{\vartheta}_u$, $\vec{\vartheta}_t^{tod}$, $\vec{\vartheta}_e^{type}$ respectively.
β	symmetric hyper parameter, Dirichlet prior of $\vec{\varphi}_k$.
γ	symmetric hyper parameter, Dirichlet prior of $\vec{\lambda}_u$.
φ_k	programs' distribution w.r.t. topic k.
ϑ_u	topics' distribution w.r.t. user u.
$\vartheta_t^{tod}, \vartheta_e^{type}$	topics' distribution w.r.t. timing t and "live/VoD" condition e respectively.
λ_u	propensity on three influential factors of user u.
$\underline{\phi}$, $\underline{\theta}$, $\underline{\lambda}$	estimation of φ, ϑ, λ, respectively.
u, k, p, t, e	index of users, topics, programs, time of day and environment(live or VoD) respectively.
U, K, V	number of all users, topics and programs.
N_u	number of transactions w.r.t. user u.
$v_{u,i}$	the i_{th} viewed item of user u.
$z_{u,i}$	topic z, because of which user u chose his or her i_{th} item.
$s_{u,i}$	path s, along which the topic $z_{u,i}$ is generated.
$t_{u,i}$	time of day when recorded $v_{u,i}$.
$e_{u,i}$	live/VoD environment of $v_{u,i}$.

either. With the Dirichlet prior γ, $s_{u,i}$ can be sampled from multinomial distribution λ_u. It is clear from the graph that $s_{u,i} = 0$ determines $z_{u,i}$ generated from "User Preference" path, $s_{u,i} = 1$ points to the path of "Time of Day Context", while $s_{u,i} = 2$ matches the path of "Live/VoD Environment".

Due to the involvement of contextual factors "time of day" and "live/VoD environment", $v_{u,i}$ is no longer the only observed variable in the model. Timing indicator $t_{u,i}$ and environment indicator $e_{u,i}$ are also visible while sampling. When $s_{u,i} = 1$, $t_{u,i}$ will decide which ϑ_t^{tod} is to be used to generate topic $z_{u,i}$, while $s_{u,i} = 2$ means the path "live/VoD environment" is chosen and $e_{u,i}$ selects ϑ_e^{type} to sample $z_{u,i}$. In general, there are five hyper parameters: α, β, γ, α^{c1}(hyper parameter for context1) and α^{c2}(hyper parameter for context2) in this model. Mapping each of them, ϑ_u, φ_k, λ_u, ϑ_t^{tod} and ϑ_e^{type} are the distributions to be estimated. Considering the two invisible elements: $s_{u,i}$ and $z_{u,i}$, we need a two-step sampling procedure, which is introduced in next subsection, to do this estimation [20].

3.3 Gibbs Sampling

To avoid a potential local optima resulting from EM algorithm, Gibbs Sampling is chosen to estimate the distribution in this paper. Gibbs Sampling is a high dimensional form of Markov Chain Monte Carlo Sampling. Its advantage is that given the full conditional probability, the acceptance rate can be 1 when sampling state on a single dimension, such that sampling effectiveness is maximized [2]. If we define K topics in the model, it means that K candidate states can be sampled for $z_{u,n}$ at each sampling, and there would be $\sum_{u\in[1,U]} N_u$ dimensions in total to switch at each iteration. The task of Gibbs Sampling is exactly iterative sampling for

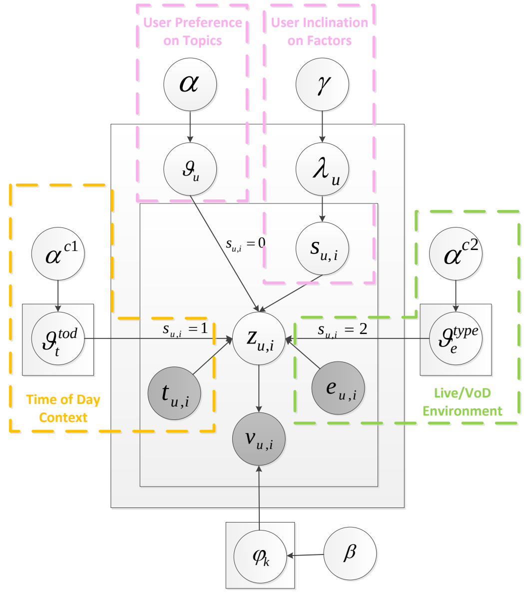

Figure 5: Context-aware LDA Graphical Model

each $z_{u,n}$ until $\underline{\phi}$ and $\underline{\theta}$ reach a relative stable distribution.

In user-oriented LDA, four most important variables n_u (total number of transactions for user u), n_u^k (number of topics assigned with k w.r.t. user u), n_k (total number of transactions sampled as topic k), and n_k^v (total number of occurrences of program v assigned to topic k) contribute together with prior distribution α and β to the final full conditional probability [11]. Inferred by Bayes' rule, this sampling probability is defined in Eq. 2.

$$\begin{aligned}
&p(z_{u,i} = k|\vec{z}_{\neg(u,i)}, \vec{v}, \alpha, \beta) \\
&\propto p(z_{u,i} = k, v_{u,i} = p|\vec{z}_{\neg(u,i)}, \vec{v}, \alpha, \beta) \\
&= \hat{\vartheta}_{u,k} \cdot \hat{\varphi}_{k,p} \\
&= \frac{n_{u,\neg(u,i)}^k + \alpha_k}{\sum_{k=1}^K (n_{u,\neg(u,i)}^k + \alpha_k)} \cdot \frac{n_{k,\neg(u,i)}^p + \beta_p}{\sum_{v=1}^V (n_{k,\neg(u,i)}^v + \beta_v)}
\end{aligned} \tag{2}$$

However, in context-aware LDA, two-step Gibbs Sampling is needed to generate $s_{u,i}$ and $z_{u,i}$, respectively. Referring to the derivation in [19], the full conditional probability of path selection, i.e., the first step sampling, can be represented as in Eq. 3 - Eq. 5, where $n_{u,\neg(u,i)}^{z_{u,i}}$, $n_{t_{u,i},\neg(u,i)}^{z_{u,i}}$ and $n_{e_{u,i},\neg(u,i)}^{z_{u,i}}$ denote the number of topic $z_{u,i}$ regardless of element (u,i) w.r.t. user u, timing $t_{u,i}$ and live/VoD environment $e_{u,i}$ separately. By the same token, $n_{u,\neg(u,i)}^{s=c}$ indicates the number of path c excluding element (u,i) chosen by user u. Thus, together with Dirichlet prior α, α^{c_1}, α^{c_2}, β and γ, these counting numbers comprise the sampling formula for $s_{u,i}$.

$$\begin{aligned}
&p(s_{u,i} = 0|\vec{s}_{\neg(u,i)}, \vec{z}, \vec{v}, \alpha, \alpha^{c_1}, \alpha^{c_2}, \beta, \gamma) \\
&\propto \frac{n_{u,\neg(u,i)}^{z_{u,i}} + \alpha_{z_{u,i}}}{\sum_{k=1}^K (n_{u,\neg(u,i)}^k + \alpha_k)} \cdot \frac{n_{u,\neg(u,i)}^{s=0} + \gamma_0}{\sum_{c=1}^3 (n_{u,\neg(u,i)}^{s=c} + \gamma_c)}
\end{aligned} \tag{3}$$

$$p(s_{u,i}=1|\vec{s}_{\neg(u,i)},\vec{z},\vec{v},\alpha,\alpha^{c_1},\alpha^{c_2},\beta,\gamma)$$
$$\propto \frac{n^{z_{u,i}}_{t_{u,i},\neg(u,i)}+\alpha^{c_1}_{z_{u,i}}}{\sum_{k=1}^{K}(n^{k}_{t_{u,i},\neg(u,i)}+\alpha^{c_2}_{k})}\cdot\frac{n^{s=1}_{u,\neg(u,i)}+\gamma_1}{\sum_{c=1}^{3}(n^{s=c}_{u,\neg(u,i)}+\gamma_c)} \quad (4)$$

$$p(s_{u,i}=2|\vec{s}_{\neg(u,i)},\vec{z},\vec{v},\alpha,\alpha^{c_1},\alpha^{c_2},\beta,\gamma)$$
$$\propto \frac{n^{z_{u,i}}_{e_{u,i},\neg(u,i)}+\alpha^{c_2}_{z_{u,i}}}{\sum_{k=1}^{K}(n^{k}_{e_{u,i},\neg(u,i)}+\alpha^{c_2}_{k})}\cdot\frac{n^{s=2}_{u,\neg(u,i)}+\gamma_2}{\sum_{c=1}^{3}(n^{s=c}_{u,\neg(u,i)}+\gamma_c)} \quad (5)$$

When the first step sampling is finished, the value of $s_{u,i}$, i.e., the path of generating procedure is determined. Given this $s_{u,i}$, the second step of topic sampling can be realized by the following posterior probabilities Eq. 6 - Eq. 8. Mapping with the contextual graphic model in Fig. 5, we can see how to discriminate these sampling formulas. Estimated users' preferences on topics $\hat{\vartheta}_u$ (calculated by $n^{k}_{u,\neg(u,i)}$) is multiplied when $s_{u,i}=0$. Topics' trend estimation w.r.t. "time of day" $\hat{\vartheta}^{tod}_t$ (statistics on $n^{k}_{t_{u,i},\neg(u,i)}$) replaces this multiplier if $s_{u,i}=1$, while under the condition $s_{u,i}=2$ only $\hat{\vartheta}^{type}_e$ (estimation on topics' popularities regarding "live/VoD environment" using $n^{k}_{e_{u,i},\neg(u,i)}$) can be involved in the formula. In addition, $\hat{\varphi}_k$, i.e., the estimation of programs' distribution on topic k, which is calculated by $n^{v}_{k,\neg(u,i)}$, is the common part for these formulas.

$$p(z_{u,i}=k|s_{u,i}=0,\vec{z}_{\neg(u,i)},\vec{v},\alpha,\alpha^{c_1},\alpha^{c_2},\beta,\gamma)$$
$$\propto \frac{n^{k}_{u,\neg(u,i)}+\alpha_k}{\sum_{k=1}^{K}(n^{k}_{u,\neg(u,i)}+\alpha_k)}\cdot\frac{n^{v_{u,i}}_{k,\neg(u,i)}+\beta_{v_{u,i}}}{\sum_{v=1}^{V}(n^{v}_{k,\neg(u,i)}+\beta_v)} \quad (6)$$

$$p(z_{u,i}=k|s_{u,i}=1,\vec{z}_{\neg(u,i)},\vec{v},\alpha,\alpha^{c_1},\alpha^{c_2},\beta,\gamma)$$
$$\propto \frac{n^{k}_{t_{u,i},\neg(u,i)}+\alpha^{c_1}_{z_{u,i}}}{\sum_{k=1}^{K}(n^{k}_{t_{u,i},\neg(u,i)}+\alpha^{c_2}_{k})}\cdot\frac{n^{v_{u,i}}_{k,\neg(u,i)}+\beta_{v_{u,i}}}{\sum_{v=1}^{V}(n^{v}_{k,\neg(u,i)}+\beta_v)} \quad (7)$$

$$p(z_{u,i}=k|s_{u,i}=2,\vec{z}_{\neg(u,i)},\vec{v},\alpha,\beta,\gamma)$$
$$\propto \frac{n^{k}_{e_{u,i},\neg(u,i)}+\alpha^{c_2}_{z_{u,i}}}{\sum_{k=1}^{K}(n^{k}_{e_{u,i},\neg(u,i)}+\alpha^{c_2}_{k})}\cdot\frac{n^{v_{u,i}}_{k,\neg(u,i)}+\beta_{v_{u,i}}}{\sum_{v=1}^{V}(n^{v}_{k,\neg(u,i)}+\beta_v)} \quad (8)$$

Having these sampling formulas, we can implement the Gibbs Sampling algorithms for both user-oriented LDA and context-aware LDA. Since sampling procedure of user-oriented LDA can be found as the same in [11, 13], we only list algorithm for context-aware LDA in Alg. 1 here.

3.4 Query Procedure

Through LDA model training, we may acquire users' personal preference on topics or topics' distribution regarding contextual factors. Nevertheless, getting recommended items through certain queries is the ultimate purpose of training this model, thus query strategy is also important for recommender. This section introduces the corresponding query procedure for both user-oriented LDA and context-aware LDA.

Algorithm 1: Gibbs Sampling for context-aware LDA

Input: programs vector $\vec{v}$, hyper parameters α, α^{c_1}, α^{c_2}, β, γ, topic number K, paths number 3
Output: paths assignments $\vec{s}$, topics assignments $\vec{z}$, estimations: $\underline{\phi}$, $\underline{\lambda}$, $\underline{\theta}$, $\underline{\theta}^{tod}$ and $\underline{\theta}^{type}$

```
//initialization;
for all users u ∈ [1,U] do
    for all program-based transactions i ∈ [1,N_u] do
        //sample path s_{u,i} for (u,i)
        s_{u,i} = c ~ Mult(1/3);
        //sample topic z_{u,i} for (u,i)
        z_{u,i} = k ~ Mult(1/K);
    end
end
//Gibbs sampling, burn-in period;
while not converged do
    for all users u ∈ [1,U] do
        for all program-based transactions i ∈ [1,N_u] do
            //sample path s_{u,i}
            s_{u,i} = c ~ p(s_{u,i}|s_{¬(u,i)}, z, v, α, α^{c1}, α^{c2}, β, γ);
            //sample topic z_{u,i}
            z_{u,i} = k ~
            p(z_{u,i}|s_{u,i} = c, z_{¬(u,i)}, v, α, α^{c1}, α^{c2}, β, γ);
        end
    end
    if converged then
        read out estimated distribution: φ, λ, θ, θ^{tod}
        and θ^{type} ;
    end
end
```

Figure 6: Gibbs Sampling for context-aware LDA

3.4.1 Query Generating

We define a segment as a time period within the same "time of day" (granularity of three hours) on a specific day for a user to distinguish query and recommending strategy. The final experiments will be done based on this unit of "segment" as well. Thinking about segment in this form, two conditions need to be considered when generating query; one is the segment with single viewed program and the other one is the segment with multiple viewed programs. In the test set, if there is only one program in the segment, solely users' preferences or contextual factors can be used to predict this program, while in segment with multiple programs, the first program can also be treated as a hint to guess the following items. More concretely, regarding user-oriented LDA and context-aware LDA, the generation of query would be different. Given the four possible combinations of algorithms and session forms, we could summarize the queries generating in Table 2. Here each sort of query $\vec{q}$ can be represented as a vector of distribution on latent topics.

Table 2: Query Vector Generating

Algorithms	Segment with Single Program	Segment with Given Program p
user-oriented LDA	$\vec{q}_u = \hat{\vartheta}_u$	$\vec{q}_{u,p} = \hat{\vartheta}_u \cdot \hat{\varphi}_{:,p}$
context-aware LDA	$\vec{q}_{u,t,e} = \hat{\lambda}_u * \left[\hat{\vartheta}_u;\hat{\vartheta}_t;\hat{\vartheta}_e\right]$	$\vec{q}_{u,t,e,p} = \vec{q}_{u,t,e} \cdot \hat{\varphi}_{:,p}$

3.4.2 *Query Executing*

Targeting on each query vector, we give the solution of similarity calculation, as listed in Table 3, to illustrate concrete approaches to execute query. In the test set, for the segment with only one program, where queries can be $\vec{q}_u$ and $\vec{q}_{u,t,e}$, we utilize the probability of program v's appearance under different conditions, $p(v|u)$ or $p(v|u,t,e)$, as the similarity indicator, while for the segment with multiple programs, in which first program p is given, facing queries as $\vec{q}_{u,p}$ and $\vec{q}_{u,t,e,p}$, we calculate the cosine value between the query vector and $\hat{\varphi}_{:,v}$ to represent the similarity. The final ranked programs list for each segment is turned out through sorting by corresponding similarity.

Table 3: Query Executing

Query	Score Calculation for Program v
$\vec{q}_u$	$p(v\|u) \propto p(u,v) = \hat{\vartheta}_u * \hat{\varphi}_{:,v}$
$\vec{q}_{u,p}$	cosine$(\vec{q}_{u,p}, \hat{\varphi}_{:,v})$
$\vec{q}_{u,t,e}$	$p(v\|u,t,e) \propto$ $\sum_c \lambda_c \cdot p(u,t,e,v\|s=c) = \vec{q}_{u,t,e} * \hat{\varphi}_{:,v}$
$\vec{q}_{u,t,e,p}$	cosine$(\vec{q}_{u,t,e,p}, \hat{\varphi}_{:,v})$

Discussion.

Through model training procedure and query schema design discussed above, users in different context can be recommended with specific programs, while their' satisfaction with these recommended lists will be investigated in the following section.

4. EXPERIMENT

In this section, we compare different approaches as "rand", "hot", "userHot", "user-oriented LDA" and "context-aware LDA" on metrics "nDCG", "MRR", "Recall Rate", "Diversity" and "Novelty", as explained in the next subsections, to illustrate the effect of contextual factors' involvement in the recommender.

4.1 Comparative Approaches

We implement the five approaches below to compare their performance for every evaluation method.

Rand presumes no patterns in user's preferences so that only randomly ranked programs are shown to users.

Hot selects programs with highest occurrence frequency in the training set and also ranks them according to their statistical popularity.

userHot chooses the most frequently viewed programs w.r.t a specific user and also sorts them for this user by their frequencies.

user-oriented LDA, as described in Section 3.1, can help analyze both users' and programs' trends on latent topics.

context-aware LDA, as mentioned in Section 3.2, detects the different trends on topics under "time of day" or "live/VoD environment" context besides users' personal preference.

4.2 Evaluations

To evaluate the ranking quality of approaches, we compare them on "nDCG@K", "MRR@K" and "Recall@K". Generally speaking, when the topic number of context-aware LDA is over 35, it can almost outweigh all other approaches on these three metrics. On the other hand, when we take "Diversity" or "Novelty" also into account, apart from approach "rand", from which extremely disordered list would be turned out, context-aware LDA still shows good performance.

4.2.1 *nDCG@K*

nDCG@K, the normalized value of DCG_K (Discounted Cumulative Gain), is always used to measure the quality of the ranking. DCG_K is defined in Eq.9, where for a ranked list, relevance of i_{th} element is symbolized as rel_i, while the penalizing factor $\log_2(i+1)$ weakens the relevance according to the ranked position. $IDCG_K$ presumes the most ideal condition that items in the top K list are ranked exactly as the same order as items ranked by their relevance descend, and calculate the DCG_K of this most ideal condition. In this paper, we utilize users' implicit feedback, playback percentage, as the relevance of item rel_i in the list. Fig. 7 shows us the value of $nDCG$@5, $nDCG$@10 and $nDCG$@15 of all approaches w.r.t. different latent topic numbers, and the values showed on the figure are the mean $nDCG$ of all segments in the test set. Since the results of "hot" and "userHot" have nothing to do with topic numbers, they are just horizontal straight lines compared with results turned from LDA related approaches. Though "userHot" can outperform "user-oriented LDA" for certain settings, "context-aware LDA" is consistently the best of all for 35 topics or more.

$$\mathrm{DCG}_K = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{9}$$

$$\mathrm{nDCG}_K = \frac{\mathrm{DCG}_k}{\mathrm{IDCG}_k} \tag{10}$$

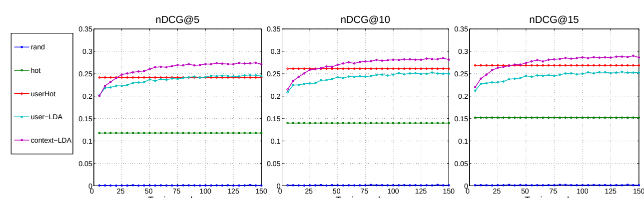

Figure 7: Normalized Discounted Cumulative Gain – The "context-LDA" reaches the highest performance when topic number $\geq$ 20, 35, and 40 for K = 5, 10, and 15 respectively.

4.2.2 *MRR*

Mean Reciprocal Rank is another measurement to judge the ranking lists. As shown in Eq. 11, $rank_i$ represents the first hit item's position in the ranked list; the smaller the $rank_i$, the better the list is. We may infer from its definition that the measurement emphasizes on the first hit item regardless of other bingo items. $|Q|$ denotes the number of queries in the test set, while on our test set it represents the total number of segments. Fig. 8 depicts this mean value of reciprocal rank considering different approaches. It shows that MRR of "user-oriented LDA" is slightly higher than "userHot" after 40 topics, while "context-aware LDA" is always placed at first after 20 topics.

$$\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{11}$$

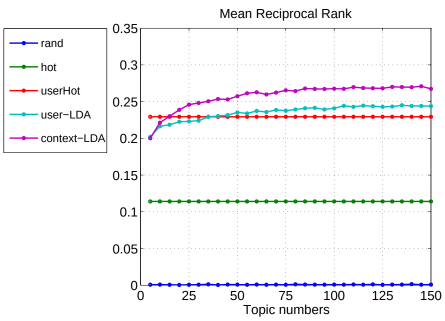

Figure 8: Mean Reciprocal Rank – The "context-LDA" outperforms the other approaches if the number of topics is greater than 15.

4.2.3 Recall

Apart from ranking quality, the proportion of recalled programs among the user's viewed programs is also very important when dealing with relevance. Thus the comparison on recall rate, as given in Eq. 12, is also provided in Fig. 9. However, due to the great performance of "userHot" on recall, "user-oriented LDA" behaves relatively weakly and seems no longer competitive, whereas "context-aware LDA" can still maintain a satisfying result from 15 or 20 topics.

$$\text{Recall}@K = \frac{\#\text{hit}@K}{\#\text{viewedprograms}@K} \tag{12}$$

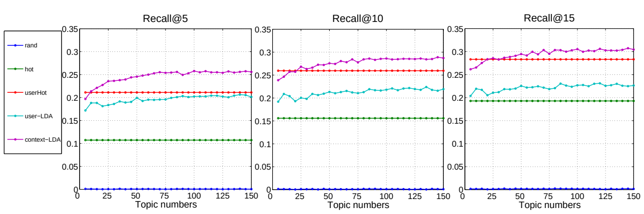

Figure 9: Recall Rate – The "context-aware LDA" outperforms other approaches when topic numbers ≥ 10, 20, and 30 for $K = 5$, 10, and 15 respectively.

4.2.4 Diversity

As mentioned in the introduction part, improving diversity and novelty is another purpose of modern recommenders. The purpose of involving contextual factors in this paper is also to increase the behavior of recommended lists on these two metrics. Therefore, the concept of diversity and novelty defined in [14] is introduced here as measurements.

Eq. 13 expresses the definition of diversity, more specifically, w.r.t. a specific user, L_{now} and L_{last} are recommended lists of arbitrary two neighbor segments. The difference set $L_{now}\backslash L_{last}$ holds the elements included in L_{now} yet not in L_{last}. With $Diversity(L_{now}, L_{last})@K$, only the top K elements will be remained in L_{now}, L_{last} separately, and the number of difference set of these two top-K lists, i.e., $|L_{now}@K\backslash L_{last}@K|$, divided by K is the diversity we want to see in the top K lists. Since performance curves are similar with different K, we only depict $Diversity@10$ in Fig. 10. Even though "rand" unsurprisingly gains the highest diversity owing to its extreme disorder, the absolute second place of "context-aware LDA" also illustrates the functionality of increasing diversity in recommended list of contextual factors involvement. Since "hot" and "userHot" always recommend same content, the diversities of them are all zero.

$$L_{now}\backslash L_{last} = \{x \in L_{now} | x \notin L_{last}\} \tag{13}$$

$$\text{Diversity}(L_{now}, L_{last})@K = \frac{|L_{now}@K\backslash L_{last}@K|}{K} \tag{14}$$

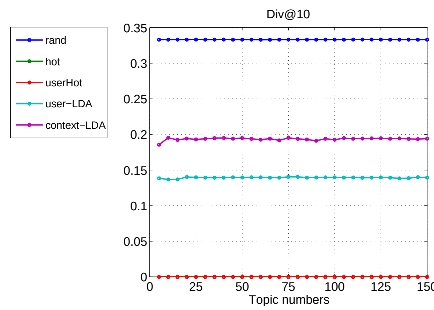

Figure 10: Diversity – Aside from the baseline "rand", the "context-LDA" approach shows dominant role, while number of topics doesn't make a big difference.

4.2.5 Novelty

The novelty definition is similar to diversity to some degree. As shown in Eq. 15, $Novelty(L)@K$ can be seen as the top-K diversity between list L and historical list H, which is the cumulative set of seen items before L appears. Fig. 11 clearly depicts that "context-aware LDA" does not exhibit so much dominance on novelty as on other metrics; only narrow margin can be figured out over "user-oriented LDA" here.

$$\text{Novelty}(L)@K = \frac{|L@K\backslash H|}{K} \tag{15}$$

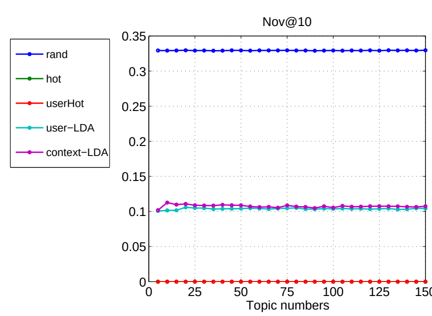

Figure 11: Novelty – The "context-LDA" approach shows only a small advantage over "user-LDA".

Discussion.

Evaluations above comprehensively show the evident advantage of making use of context-awareness in LDA model in terms of either ranking or diversity. In addition, w.r.t. our scenario and dataset, a number of topics greater than 35 can ensure the superiority of context-aware LDA for ranking metrics (nDCG, MRR and Recall), while for diversity and novelty metrics, the number of topics makes no big difference.

5. CONCLUSION AND FUTURE WORK

To adapt to users' dynamically changing preference in our TV content provider "Vision", we designed and realized a context-aware recommender in this paper. During the pre-analysis of rich EPG data, normalized entropy was used as an indicator to tell users' propensity on specific field and contextual factors' influences. Thereby program info was chosen as basic field and "time of day" and "live/VoD environment" were selected as candidate contextual factors. Given these chosen conditions, building user-oriented LDA and context-aware LDA models created the opportunity of analyzing the influence of context-awareness. Final experiments on both ranking based and diversity oriented metrics demonstrate the advantage of involving contextual factors in recommender in our scenario.

Nevertheless, we only focused on LDA topic model in this paper, while other algorithms as collaborative filtering and singular value decomposition can also be added to be compared together in the future. Moreover, solely off-line metrics used in our experiments can be enriched with online recommender tests, such as A/B test, to better capture the users' direct reaction to the recommender. In general, future work includes further algorithm comparisons and online test implementation.

6. ACKNOWLEDGMENTS

The work of the first author has been continuously supported by China Scholarship Council (CSC). In addition, the TV content provider "Vision" is partially sponsored by "EC FP7 FI-Content 2" project.

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. RecSys '08, pages 335–336, NY, USA, 2008. ACM.

[2] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

[3] R. G. Apaza, E. V. Cervantes, L. C. Quispe, and J. O. Luna. Online courses recommendation based on LDA. In *Proc. the 1st Symposium on Information Management and Big Data - SIMBig 2014, Cusco, Peru, October 8-10, 2014.*, pages 42–48, 2014.

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

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[6] P. Cremonesi and R. Turrin. Time-evolution of iptv recommender systems. In *Proc. of the 8th intl. interactive conf. on Interactive TV & Video*, EuroITV '10, pages 105–114, New York, USA, 2010. ACM.

[7] M. Dabrowski, J. Gromada, and H. Moustafa. Context-awareness for iptv services personalization. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth Intl. Conf. on*, pages 37–44, 2012.

[8] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath. The youtube video recommendation system. RecSys '10, pages 293–296, New York, USA, 2010. ACM.

[9] T. Di Noia, V. C. Ostuni, J. Rosati, P. Tomeo, and E. Di Sciascio. An analysis of users' propensity toward diversity in recommendations. RecSys '14, pages 285–288, New York, USA, 2014. ACM.

[10] N. Hariri, B. Mobasher, and R. Burke. Context adaptation in interactive recommender systems. RecSys '14, pages 41–48, New York, USA, 2014. ACM.

[11] G. Heinrich. Parameter estimation for text analysis. Technical report, Fraunhofer IGD, Darmstadt, Germnay, 2009.

[12] F. Hopfgartner. *Understanding Video Retrieval*. VDM Verlag, Saarbrücken, Germany, 2007.

[13] Z. Jin. Lda math gossiping. Technical report, Tencent, China, 2013.

[14] N. Lathia, S. Hailes, L. Capra, and X. Amatriain. Temporal diversity in recommender systems. In *Proceedings of the 33rd intl. ACM SIGIR conf. on Research and Development in Information Retrieval*, SIGIR '10, pages 210–217, New York, USA, 2010. ACM.

[15] L. M. Masisi, F. V. Nelwamondo, and T. Marwala. The use of entropy to measure structural diversity. *CoRR*, abs/0810.3525, 2008.

[16] U. Panniello, A. Tuzhilin, and M. Gorgoglione. Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and User-Adapted Interaction*, 24(1-2):35–65, Feb. 2014.

[17] J. San Pedro and A. Karatzoglou. Question recommendation for collaborative question answering systems with rankslda. RecSys '14, pages 193–200, New York, USA, 2014. ACM.

[18] M. Xu, S. Berkovsky, S. Ardon, S. Triukose, A. Mahanti, and I. Koprinska. Catch-up tv recommendations: show old favourites and find new ones. In *RecSys*, pages 285–294. ACM, 2013.

[19] H. Yin, B. Cui, L. Chen, Z. Hu, and C. Zhang. Modeling location-based user rating profiles for personalized recommendation. *ACM Trans. Knowl. Discov. Data*, 9(3):19:1–19:41, Apr. 2015.

[20] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: A location-content-aware recommender system. KDD '13, pages 221–229, New York, USA, 2013. ACM.

[21] J. Yuan, F. Sivrikaya, S. Marx, and F. Hopfgartner. When to recommend what? a study on the role of contextual factors in ip-based tv services. In *MindTheGap@iConference*, pages 12–18, 2014.