Vol.2011-DBS-153 No.23
2011/11/3

Topic-based Relevance Modeling of Metadata to Transactions for Collaborative Filtering

Robert Sumi,[†1] Yutaka Kabutoya,[†1]
Tomoharu Iwata,[†2] Toshio Uchiyama[†1]
and Ko Fujimura[†1]

We propose a probabilistic topic model that infers the relevance of content-descriptive metadata to assist in the task of collaborative filtering. Metadata can be used to add functionality to purely user-based models, such as, in a purchase recommendation service, the ability to recommend items that have not yet been purchase by any user in the system. However, content-based methods are vulnerable overfitting on metadata that is not descriptive of the users interests, and thus detrimental to recommendation accuracy. We describe a model based GM-LDA proposed by Blei, et al. that incorporates a relevance-sensitive topic model for metadata to assist collaborative filtering methods. We seek to mitigate the potential negative impact of fitting to unhelpful metadata by assuming that metadata words are either relevant or irrelevant in terms of descriptive power, and allow our model to infer the relevance of each metadatum before using the information. We compare our model's normal and cold-start recommendation accuracy on data taken from MovieLens against that of conventional methods found in the literature.

1. Introduction

The popularity of services such as the online retailer Amazon[⋆1] and the online film and television service Netflix[⋆2] has shown that recommender systems have immense utility in today's world. Many of these popular services are built upon collaborative filtering algorithms[12], which find similarites between the transaction histories of users of the system and use them in making recommendations. Others employ content-based filtering methods[9], which make recommendations based on similarities between the content of items and users' described preferences, or the preferences implied by their transaction history[1]. The enormous efficiency of recommender systems of both types has seen their rise to widespread use in online applications. However, neither is without problems.

Purely collaborative filters are unable to handle certain important situations, such as recommending an item that has not yet been rated or purchased by any user in the system. These situations are referred to as *cold starts*[13]. Purely content-based filtering experiences overfitting problems, in that, because all recommendations are generated based on the users information in relation to the content information, principled recommendations cannot be made for items that do not conform to the user's existing tastes[1]. Also, because the content-based information for each of the users purchases is representative of the whole item purchased, a user's preference profile in a content-based recommender can become overfitted, as all the content of the item is assumed to be relevant to the user's interests, not just the parts that actually are.

In this paper we introduce a generative probabilistic model to address these weaknesses. It infers whether or not each metadatum in the input is relevant to the content it describes, and then models relevant and irrelevant metadata separately alongside a conventional collaborative filtering algorithm based on Latent Dirichlet Allocation. More specifically, the model introduced here extends the GM-LDA model introduced by Blei, et al.[2], with the aim of addresing the relevance between two different types of data. GM-LDA can model the relationships between two kinds of data, as in it, the topics of the two data types are assumed to be generated from the same parameter individually. On the other hand, Corr-LDA, also proposed by Blei, et al[2], generates the topics of one type of data from the distribution of the topics of the other type. This creates a very strong dependency between the two types of data, and thus makes GM-LDA better suited to address the cold start problem, one of its principal advantages.

As stated, our model differs from GM-LDA in that GM-LDA does not attempt to model the relevance of the content-based information. The content-based portion of GM-LDA thus experiences overfitting, which our relevance-sensitive model seeks to correct. A similar relevance-modeling method was proposed by Iwata, et al.[6], but it cannot make cold-start recommendations, as it is based on

†1 NTT Cyber Solutions Laboratories, NTT Corporation
†2 NTT Communcation Science Laboratories, NTT Corporation
⋆1 http://www.amazon.com
⋆2 http://www.netflix.com

Corr-LDA, due to the dependencies between the items and the metadata inherent in Corr-LDA. Our model, conversely, is well-suited to addressing the problem of making cold-start recommendations.

We will show that our model is capable of integrating noisy metadata into a collaborative recommender system without loss of recommendation accuracy with respect to well-known recommender systems found in the literature.

2. Related Work

There are many generative topic models extant, due to their good performance in myriad contexts. Among the most well-known and studied is latent Dirichlet allocation (LDA), proposed by Blei, et al. in 3). Existing research has seen LDA ornamented with content-related information for such purposes as modeling image annotation[2] and document annotation[6]. Probabilistic latent semantic analysis (PLSA), proposed by Hofmann[5], is also among the most popular generative topic models, and has been used with content information to perform tasks such as music recommendation[14] and document filtering[10].

Methods of tag processing have been experimented with in a variety of recommenders. External taxonomies[8] and knowledge bases[4] have been used, but these approaches rely on the pre-existence of an appropriate taxonomy or knowledge base, and, failing that, must rely on human effort to create one.

Automatic image annotation methods are related to the method we propose, and are well-explored. Jeon, et al.[7] use a clustering algorithm, and then generate annotations from analysing the resultant clusters. Blei, et al.[2] have looked into automatic image annotation, and their GM-LDA model is the base from which the method proposed in this paper was built.

Generative metadata relevance modeling has been explored before by Iwata, et al.[6], but, as previously mentioned, the dependencies between the user transaction topics and the relevance modeling of the metadata makes their approach unsuitable for cold-start recommendations.

3. Proposed Method

3.1 Preliminaries

We assume a set of M user purchases and N metadata words. The mth transaction is expressed as a pair (i_m, u_m), where $i_m \in \{1, \ldots, I\}$ is the item being purchased and $u_m \in \{1, \ldots, U\}$ is the user making the purchase. The nth metadatum is expressed as a pair (j_n, w_n), where $j_n \in \{1, \ldots, I\}$ is the item being described, and $w_n \in \{1, \ldots, W\}$ is the description being associated with the item.

3.2 Model

Table 1 Notation

Symbol	Description
M	the number of transactions
N	the number of metadata entries
K	the number of latent topics
I	the number of items
W	the number of unique metadata words
U	the number of accounts
i_m	the item in the mth transaction, $i_m \in \{1, \ldots, I\}$
u_m	the user in the mth transaction, $u_m \in \{1, \ldots, U\}$
z_m	the topic of the mth transaction, $z_m \in \{1, \ldots, K\}$
j_n	the item in the nth metadatum, $j_n \in \{1, \ldots, I\}$
w_n	the phrase in the nth metadatum, $w_n \in \{1, \ldots, W\}$
y_n	the topic of the nth metadatum, $y_n \in \{1, \ldots, K\}$
r_n	the relevance of the nth metadatum, $r_n \in \{0, 1\}$

Our model assumes that the mth transaction and nth metadatum each have a latent topic z_m or y_n. The topics for both are drawn from the same multinomial distribution over topics $\boldsymbol{\pi}$. In each transaction and metadatum, the items i_m and j_n are drawn from a topic-specific multinomial distribution $\boldsymbol{\theta}_{z_m}$ (or $\boldsymbol{\theta}_{y_n}$) over all items. For each transaction, the user u_m is drawn from a topic-specific multinomial distribution over users $\boldsymbol{\phi}_{z_m}$. Each metadatum is considered to be either relevant or irrelevant, denoted by r_n, with relevance being generated from a Bernoulli distribution $\boldsymbol{\lambda}$. Each word in the metadata is generated either from a topic-unspecific multinomial background distribution $\boldsymbol{\psi}_0$ if the word is not relevant ($r_n = 0$), or a topic-specific multinomial distribution $\boldsymbol{\psi}_{y_n}$ if it is ($r_n = 1$). **Table 1** offers a summary of the notation we use in this paper.

The proposed model assumes the following generative process for the transactions $\boldsymbol{i} = \{i_m\}_{m=1}^{M}$, $\boldsymbol{u} = \{u_m\}_{m=1}^{M}$ and the metadata $\boldsymbol{j} = \{j_n\}_{n=1}^{N}$, $\boldsymbol{w} = \{w_n\}_{n=1}^{N}$:

(1) Draw topic proportions $\boldsymbol{\pi} \sim \text{Dirichlet}(\delta)$

(2) Draw irrelevant word probability $\boldsymbol{\psi}_0 \sim \text{Dirichlet}(\gamma)$

(3) Draw word-relevance probability $\boldsymbol{\lambda} \sim \text{Beta}(\eta)$
(4) For each topic $k = 1, \ldots, K$:
  (a) Draw item probability $\boldsymbol{\theta}_k \sim \text{Dirichlet}(\alpha)$
  (b) Draw user probability $\boldsymbol{\phi}_k \sim \text{Dirichlet}(\beta)$
  (c) Draw word probabilty $\boldsymbol{\psi}_k \sim \text{Dirichlet}(\gamma)$
(5) For each transaction $m = 1, \ldots, M$:
  (a) Draw topic $z_m \sim \text{Multinomial}(\boldsymbol{\pi})$
  (b) Draw item $i_m \sim \text{Multinomial}(\boldsymbol{\theta}_{z_m})$
  (c) Draw user $u_m \sim \text{Multinomial}(\boldsymbol{\phi}_{z_m})$
(6) For each word in the metadata $n = 1, \ldots, N$:
  (a) Draw topic $y_n \sim \text{Multinomial}(\boldsymbol{\pi})$
  (b) Draw item $j_n \sim \text{Multinomial}(\boldsymbol{\theta}_{y_n})$
  (c) Draw relevance $r_n \sim \text{Bernoulli}(\boldsymbol{\lambda})$
  (d) If $r_n = 0$:
    (i) Draw word $w_n \sim \text{Multinomial}(\boldsymbol{\psi}_0)$
  (e) Otherwise:
    (i) Draw word $w_n \sim \text{Multinomial}(\boldsymbol{\psi}_{y_n})$

In the proposed model, each of the multinomial parameters of the topic, word, item and user distributions, $\boldsymbol{\pi}$, $\boldsymbol{\theta}$, $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$, are assumed to be drawn from Dirichlet priors to provide smoothing while retaining mathematical simplicity by exploiting conjugacy, while the Bernoulli parameter $\boldsymbol{\lambda}$ is assumed to be generated by a Beta distribution, chosen for the same reasons.

Figure 1 describes our model graphically in plate notation. The shaded nodes are observed variables, and the unshaded nodes are latent variables and model parameters. The rectangles denote multiplication, where each node and edge contained or exiting the rectangle are replicated the number of times shown in the lower-right.

The joint distribution on the set of topics $\boldsymbol{z} = \{z_m\}_{m=1}^M$, users and items in the transaction data and the set of topics $\boldsymbol{y} = \{y_n\}_{n=1}^N$, relevances $\boldsymbol{r} = \{r_n\}_{n=1}^N$, items and words in the metadata is defined below:

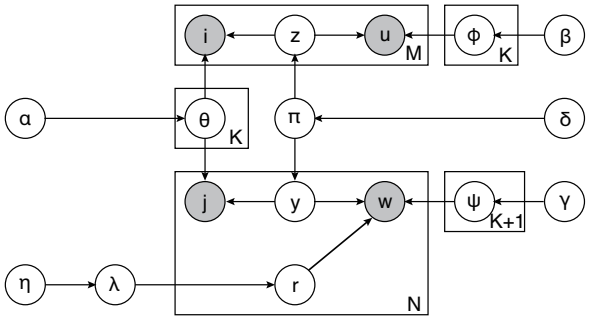

Fig. 1 Graphical representation of the proposed model

$$P(\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{i}, \boldsymbol{w}, \boldsymbol{j} | \alpha, \beta, \gamma, \eta, \delta)$$
$$= P(\boldsymbol{w}|\boldsymbol{r}, \boldsymbol{y}, \gamma) P(\boldsymbol{u}|\boldsymbol{z}, \beta) P(\boldsymbol{r}|\eta) P(\boldsymbol{i}, \boldsymbol{j}|\boldsymbol{z}, \boldsymbol{y}, \alpha) P(\boldsymbol{z}, \boldsymbol{y}|\delta). \quad (1)$$

Our choice of a Dirichlet prior on the multinomial distributions, and a Beta prior on the Bernoulli, allows us to easily integrate them out in the factors of the joint distribution shown above. The expansions of the joint distribution's factors are listed below. The distribution of metadata phrases, given their relevance and their topic is given by:

$$P(\boldsymbol{w}|\boldsymbol{r}, \boldsymbol{y}, \gamma) = \frac{\Gamma(W\gamma)^{K+1}}{\Gamma(\gamma)^{(K+1)W}} \prod_{y=0}^{K} \frac{\prod_{w=1}^{W} \Gamma(N'_{yw} + \gamma)}{\Gamma(N'_y + W\gamma)}, \quad (2)$$

where N'_{kw} is the number of metadata tags with word w that are of topic k, $k \in \{0, \ldots, K\}$, with $k = 0$ indicating that the tag is inferred as irrelevant, $k \neq 0$ indicating that the tag is inferred as relevant, and where N'_k is the number of relevant metadata tags of topic k, with $k = 0$ indicating all tags inferred as irrelevant. The distribution of users given their topics us given by:

$$P(\boldsymbol{u}|\boldsymbol{z}, \beta) = \frac{\Gamma(U\beta)^K}{\Gamma(\beta)^{KU}} \prod_{z=1}^{K} \frac{\prod_{u=1}^{U} \Gamma(M_{zu} + \beta)}{\Gamma(M_z + U\beta)}, \quad (3)$$

where M_{ku} is the number of transactions of topic k with user u, and M_k is the number of transactions with topic k.

The distribution of relevances is given by:

$$P(\boldsymbol{r}|\eta) = \frac{\Gamma(2\eta)}{\Gamma(\eta)^2} \frac{\Gamma(N_0' + \eta)\,\Gamma(N - N_0' + \eta)}{\Gamma(N + 2\eta)}, \tag{4}$$

The distribution of items given their topics is given by:

$$P(\boldsymbol{i}, \boldsymbol{j}|\boldsymbol{z}, \boldsymbol{y}, \alpha) = \frac{\Gamma(I\alpha)^K}{\Gamma(\alpha)^{KI}} \prod_{k=1}^{K} \frac{\prod_{i=1}^{I} \Gamma(M_{ki} + N_{ki} + \alpha)}{\Gamma(M_k + N_k + I\alpha)}, \tag{5}$$

where M_{ki} is the number of transactions of topic k with item i, N_{ki} is the number of metadata tags with topic k on item i and N_k is the number of metadata tags of topic k, ignoring relevance. The distribution of topics is given by:

$$P(\boldsymbol{z}, \boldsymbol{y}|\delta) = \frac{\Gamma(K\delta)}{\Gamma(\delta)^K} \frac{\prod_{k=1}^{K} \Gamma(M_k + N_k + \delta)}{\Gamma(M + N + K\delta)}. \tag{6}$$

3.3 Recommendation

We can recommend items to users via the following probability:

$$\begin{aligned} &P(i, u|\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{i}, \boldsymbol{w}, \boldsymbol{j}, \alpha, \beta, \gamma, \eta, \delta) \\ &= \sum_{k=1}^{K} \frac{M_{ki} + N_{ki} + \alpha}{M_k + N_k + I\alpha} \frac{M_{ku} + \beta}{M_k + U\beta} \frac{M_k + N_k + \eta}{M + N + K\eta}. \end{aligned} \tag{7}$$

It's visible from this probability why our model can make cold-start recommendations. In the first term, even if all M_{ki} are zero (no transaction has involved item i), the resultant probability may still be non-zero, as if there is metadata for i, then some N_{ki} will be non-zero, and therefore the summation and thus the probability of user u purchasing item i can be non-zero.

In order to be able to evaluate this probability, we must first infer the topics of the transactions, $\boldsymbol{z}$, the topics of the metadata words, $\boldsymbol{y}$, and their relevance, $\boldsymbol{r}$.

3.4 Inference

Collapsed Gibbs sampling can be used to infer the topics of the transactions, $\boldsymbol{z}$, as well as the topics of the metadata words, $\boldsymbol{y}$, and their relevance, $\boldsymbol{r}$. From a randomly-determined initial state, each latent variable is sampled based on its conditional distribution with respect to the current values of all other latent variables. This process is iterated many times (eg. 500), with all latent variables sampled at each iteration. Collapsed Gibbs sampling requires a set of *updating rules*, which are the conditional sampling probabilities of the latent variables.

The update equations for our model are listed below:

$$\begin{aligned} &P\left(z_m = k|\boldsymbol{z}_{\backslash m}, \boldsymbol{y}, \boldsymbol{r}, \boldsymbol{u}, \boldsymbol{i}, \boldsymbol{w}, \boldsymbol{j}, \alpha, \beta, \gamma, \eta, \delta\right) \\ &\quad \propto \frac{\left(M_{k\backslash m} + N_k + \delta\right)\left(M_{ku_m\backslash m} + \beta\right)\left(M_{ki_m\backslash m} + N_{ki_m} + \alpha\right)}{\left(M_{k\backslash m} + U\beta\right)\left(M_{k\backslash m} + N_k + I\alpha\right)}, \end{aligned} \tag{8}$$

$$\begin{aligned} &P\left(r_n = 0|\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{r}_{\backslash n}, \boldsymbol{u}, \boldsymbol{i}, \boldsymbol{w}, \boldsymbol{j}, \alpha, \beta, \gamma, \eta, \delta\right) \\ &\quad \propto \frac{\left(N'_{0w_n\backslash n} + \gamma\right)\left(N'_{0\backslash n} + \eta\right)}{\left(N'_{0\backslash n} + W\gamma\right)}, \end{aligned} \tag{9}$$

$$\begin{aligned} &P\left(r_n = 1|\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{r}_{\backslash n}, \boldsymbol{u}, \boldsymbol{i}, \boldsymbol{w}, \boldsymbol{j}, \alpha, \beta, \gamma, \eta, \delta\right) \\ &\quad \propto \frac{\left(N'_{y_nw_n\backslash n} + \gamma\right)\left(N_{\backslash n} - N'_{0\backslash n} + \eta\right)}{\left(N'_{y_n\backslash n} + W\gamma\right)}, \end{aligned} \tag{10}$$

$$\begin{aligned} &P\left(y_n = k|r_n = 0, \boldsymbol{z}, \boldsymbol{y}_{\backslash n}, \boldsymbol{r}_{\backslash n}, \boldsymbol{u}, \boldsymbol{i}, \boldsymbol{w}, \boldsymbol{j}, \alpha, \beta, \gamma, \eta, \delta\right) \\ &\quad \propto \frac{\left(M_k + N_{k\backslash n} + \delta\right)\left(N'_{0w_n\backslash n} + \gamma\right)\left(M_{kj_n} + N_{kj_n\backslash n} + \alpha\right)}{\left(N'_{0\backslash n} + W\gamma\right)\left(M_k + N_{k\backslash n} + I\alpha\right)}, \end{aligned} \tag{11}$$

$$\begin{aligned} &P\left(y_n = k|r_n = 1, \boldsymbol{z}, \boldsymbol{y}_{\backslash n}, \boldsymbol{r}_{\backslash n}, \boldsymbol{u}, \boldsymbol{i}, \boldsymbol{w}, \boldsymbol{j}, \alpha, \beta, \gamma, \eta, \delta\right) \\ &\quad \propto \frac{\left(M_k + N_{k\backslash n} + \delta\right)\left(N'_{kw_n\backslash n} + \gamma\right)\left(M_{kj_n} + N_{kj_n\backslash n} + \alpha\right)}{\left(N'_{k\backslash n} + W\gamma\right)\left(M_k + N_{k\backslash n} + I\alpha\right)}. \end{aligned} \tag{12}$$

where a backslash subscript ($\backslash m$ or $\backslash n$) indicates the removal of the given transaction or metadata tag from the count in question.

4. Experiments

4.1 Methodology

We conducted two experiments to measure the performance of our model, investigating both the performance of the model in a typical recommendation situation, and in the cold-start case.

The *top-L accuracy* of the proposed model was tested on a dataset constructed from the transaction logs and metadata provided by the MovieLens research project. When measuring top-L accuracy, each user's last purchase is held out, and the rest of the data is used for training the model, which generates a list of its L highest probability items for each user, and checks whether each user's held-

out transaction is among those proposed by the model. A higher ratio of correct predictions to the number of users shows better accuracy. Several well-known recommenders were compared with our proposed model.

Additionally, the *Receiver Operating Characteristic curve* (or ROC curve), which plots the *true positive rate* (sensitivity) vs the *false positive rate* (1 - specificity)[13], was determined. To test cold-start recommendation accuracy, one-tenth of the items in the dataset were held out for transaction training, but not for metadata training. This means that, from the perspective of the models in the experiment, the held-out items have been purchased by no one, but have been tagged. The models then generated a ranking of all the held-out items for each user. The items in each user's list above and below a given rank threshold were examined to determine the true positive rate and false positive rate, which are described below:

$$TPR = \frac{\#\text{ of true positives}}{\#\text{ of true positives} + \#\text{ of true negatives}}, \tag{13}$$

$$FPR = \frac{\#\text{ of false positives}}{\#\text{ of false positives} + \#\text{ of true negatives}}, \tag{14}$$

where a true positive is an item rated *above* the threshold that *was* purchased by the user, a false positive is an item rated *above* the threshold that *was not* purchased by the user, a true negative is an item rated *below* the threshold that *was not* purchased by the user, and a false negative is an item rated *below* the threshold that *was* purchased by the user.

To generate the ROC curve, the rank threshold was lowered at regular intervals, beginning by considering all items as not recommended, and ending by considering all items as recommended.

In our experiments, we choose the hyperparameters $\alpha = \beta = \gamma = \eta = \delta = 1$.

4.2 Dataset

Table 2 Dataset

number of transactions	875925
number of metadata tags	35824
number of users	7886
number of items	3417
number of unique phrases	2293

The MovieLens project is a collaborative filtering recommender system that allows users to rate movies and receive recommendations in turn. The dataset provided by the project consists of real logged user transactions, and has been made publicly available for download.[⋆1] Here, we have interpreted the MovieLens rating data instead as purchase data, and discard the reputation score.

First, we stemmed the words in each metatag individually, then added them to reconstruct a single tag. Then, we reduced the number of transactions and tags as follows: Starting from 10 million transactions and 95580 metadata entries, a set of 1 million consecutive transactions were chosen. Then, movies, users and metadata phrases that were not prolific enough were removed. A phrase needed to be used at least three times to avoid being filtered out. Movies were required to have been tagged three times, users were required to have purchased at least five movies, and each movie was required to have been purchased at least ten times. This resulted in the dataset shown in **Table 2**. When holding out one-tenth of the items for the cold-start experiment, the result was a test set of 341 items.

4.3 Conventional Methods

The proposed model was compared against the following methods:

(1) *Latent Dirichlet Allocation.* Latent Dirichlet allocation is a topic model in which each transaction is assumed to belong to a latent topic (introduced by Blei, et al.[3]). In this paper, we use a form of LDA equivalent to the most widely seen one, in which both users and movies are drawn from topic-specific multinomial distributions. In LDA, all of the model parameters have Dirichlet priors placed on them to reduce overfitting. LDA was used to make recommendations according to the following:

$$P\left(i, u | \boldsymbol{i}, \boldsymbol{u}, \boldsymbol{z}, \alpha, \beta, \delta\right) = \sum_{k=1}^{K} \frac{M_{ki} + \alpha}{M_k + I\alpha} \frac{M'_{ku} + \beta}{M_k + U\beta} \frac{M_k + \delta}{M + K\delta}. \tag{15}$$

We infer the topic assignments $\boldsymbol{z}$ by collapsed Gibbs sampling as in our

⋆1 http://www.grouplens.org/node/73

proposed model, with the update equation given here:

$$P\left(z_m = k|\boldsymbol{z}_{\backslash m}, \boldsymbol{u}, \boldsymbol{i}, \alpha, \beta, \delta\right) \propto \frac{\left(M_{k\backslash m} + \delta\right)\left(M_{ku_m\backslash m} + \beta\right)\left(M_{ki_m\backslash m} + \alpha\right)}{\left(M_{k\backslash m} + U\beta\right)\left(M_{k\backslash m} + I\alpha\right)}. \tag{16}$$

We used $\alpha = \beta = \delta = 1$ for the hyperparameteres in our experiments.

(2) *User-based Collaborative Filtering.* User-based CF, proposed by Resnick et al.[11] makes recommendations based on the transaction histories of users similar to the user in question. The similarity metric chosen for this paper was the *Pearson Correlation Coefficient*, which is described here:

$$Sim\left(u, u'\right) = \frac{\sum_{i=1}^{I}\left(\left(M_{ui} - \overline{M}_u\right)\left(M_{u'i} - \overline{M}_{u'}\right)\right)}{\sqrt{\left(\sum_{i=1}^{I}\left(M_{ui} - \overline{M}_u\right)^2\right)\left(\sum_{i=1}^{I}\left(M_{u'i} - \overline{M}_{u'}\right)^2\right)}}, \tag{17}$$

where M_{ui} is the number of times user u purchased item i, and $\overline{M}_u$ is the number of items purchased by user u divided by the number of items in the system, I.

With the similarities in hand, recommendations are made as follows:

$$P\left(i|u\right) \propto \overline{M}_u + \frac{\sum_{u'\neq u} Sim\left(u, u'\right)\left(M_{u'i} - \overline{M}_{u'}\right)}{\sum_{u'\neq u}\left|Sim\left(u, u'\right)\right|}. \tag{18}$$

(3) *Unigram.* This simple method does not model any user-specific information, and generates all transactions from a single multinomial distribution with hyperparameter $\alpha = 1$ as follows:

$$P(i|\boldsymbol{i}, \alpha) \propto \frac{M_i + \alpha - 1}{M + I\alpha - I}. \tag{19}$$

where M_i is the number of transactions involving item i, $i \in \{1, \ldots, I\}$.

(4) *Content-based Naive Bayes Classifier.* This method is not collaborative, and only uses the purchase history of a user together with content information about the items purhased to make recommendations. Recommendations are generated by determining the probability of an item belonging to the class "purchased" ($c = 1$), rather than "not purchased" ($c = 0$), and are formulated as follows:

$$P(i|u) \propto \frac{\frac{|\boldsymbol{i}_{u1}|+\alpha}{I+2\alpha}\prod_{w\in\boldsymbol{w}_i}\frac{\beta+\sum_{i'\in\boldsymbol{i}_{u1}} N_{i'w}}{W\beta+\sum_{i'\in\boldsymbol{i}_{u1}}\sum_{w'=1}^{W} N_{i'w'}}}{\sum_{c=0}^{1}\frac{|\boldsymbol{i}_{uc}|+\alpha}{I+2\alpha}\prod_{w\in\boldsymbol{w}_i}\frac{\beta+\sum_{i'\in\boldsymbol{i}_{uc}} N_{i'w}}{W\beta+\sum_{i'\in\boldsymbol{i}_{uc}}\sum_{w'=1}^{W} N_{i'w'}}}, \tag{20}$$

where c indicates the class that represents whether or not a user purchased an item, with $c = 1$ indicating a purchase and $c = 0$ indicating no purchase, $\boldsymbol{i}_{uc}$ is the number of items classified as class c by user u, $\boldsymbol{w}_i$ is the number of metadata words associated with item i, and N_{iw} is the number of times metadata word w has been associated with item i.

For our experiments, $\alpha = \beta = 1$ were used for the smoothing parameters.

4.4 Results

Here, we present the results of the discussed experiments using the proposed model. Using the reduced MovieLens dataset, we investigated the cold-start and normal top-L recommendation capabilities of our model in relation to the above mentioned baselines. For LDA and the proposed model, while measuring top-L accuracy, the number of latent topic variables, K, was chosen by testing multiples of 10 from 10 through to 100, then choosing the model with the best accuracy, which resulted in $K = 80$ for LDA and $K = 10$ for our model.

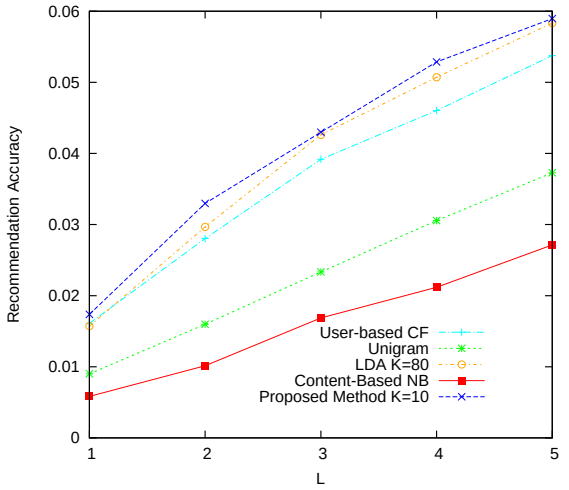

Fig. 2 Plot of the top-L accuracy ($L = 1, \ldots, 5$) of the proposed method and the compared methods on the data

Figure 2 shows the Top-L accuracy of the proposed model in relation to the

baselines. The proposed model slightly outperforms LDA, but the differences are not statistically significant. The proposed model significantly outperforms the other baselines, and outperforms the closest competitor, user-based CF, statistically significantly with $p < 0.01$ at $L = 2, 4$ and $p < 0.05$ $L = 5$ in a chi-squared test.

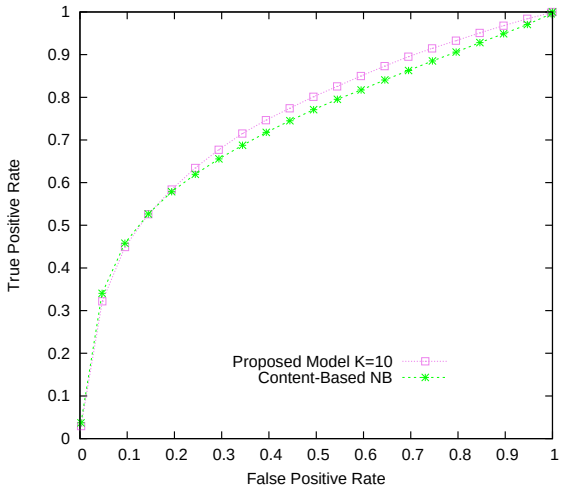

Fig. 3 ROC curves of the proposed model and a content-based naive Bayes classifier

For cold-start recommendations, the best version of the proposed model was chosen, giving $K = 10$. The proposed model was found to be more effective than the content-based naive Bayes classifier baseline. **Figure 3** shows the ROC curves generated in our cold-start experiment. The proposed model can be seen to have a better true positive rate than the content-based NB classifier where there is a higher false positive rate. Both models are steeply curved in the desirable left-hand portion of the graph, indicating far better than random performance.

Table 3 shows the tags modeled as irrelevant by our model when $K = 40$, along with some of the tags modeled as relevant (there are too many to show them all).

4.5 Discussion

The proposed model performs well in normal and cold-start recommedation relative to the baselines, but does not significantly outperform LDA in normal recommendation, despite having access to extra information. We feel that this arises from two things. First, the metadata used was very small in relation to the transaction data. The ratio of about 1:24 between metadata and transactions explains the lack of performance gain by the proposed model. Secondly, during inference, most of the tags were inferred to be relevant. Table 3 shows all the tags that were inferred as irrelevant alongside some that were labeled as relevant. Several of the tags inferred to be relevant are clearly not relevant from a human's perspective. This would contribute to the lack of performance gain by causing the model to consider noisy metadata, which can offset the benefits to recommendation of having the extra information. From the perspective of inference, however, we feel that it is reasonable for our model to have labeled these tags as such. Many of the strange or unhelpful tags labeled as relevant, such as "pg13" or "criterion", were generated in multitudes by the same user. They would therefore clearly be correlated very well with at least that user's purchase preferences. Similarly, some of the tags which would seem relevant to a human, but are labeled as irrelevant by the model are possibly selected as irrelevant due to their excessive generality or specificity. Tags like "cowboy", "los angeles", and "fugitive" are common elements in many movies that transcend genre, and thus are not very descriptive of why a user might choose to watch a movie. Even tags

Table 3 A subset of the relevance modeling results (all inferred-irrelevant tags are listed) for K=40

movie	inferred relevant tags	inferred irrelevant tags
Yojimbo	akira kurosawa	atmospheric
Chicago (2002)	musical	chicago
Amores Perros	dog	hitman
The 39 Steps (1935)	criterion	fugitive
The Queen	pg13	government
The Pink Panther Strikes Again	slapstick	inpector clouseau series
Bambi	disney	orphaned cartoon character
Titanic (1997)	chick flick	oscar best cinematography
Jeremiah Johnson	culture clash	cowboy
Crash (2006)	racism	los angeles
Starter for 10	james mcavoy	on computer

like "oscar best cinematography", which would normally be usefully descriptive of a movie, can be easily seen to be irrelevant to who chooses to purchase a movie as popular as "Titanic". We feel it is reasonable to have label these tags as irrelevant, but the problem remains that our model only inferred 12 irrelevant tags. Future work should investigate the cause of this insensitivity.

5. Conclusion

We proposed a generative probabilistic topic model able to make cold-start recommendations that infers the relevance of the transaction metadata it uses. The experiments we conducted demonstrate our model's use of noisy metadata without performance degradation relative to well-known recommenders from the literature. Additionally, the method proposed achieves better sensitivity and specificity when making cold-start recommendations than an entirely content-based recommender. These results demonstrate the potential utility of our model in the context of recommender systems that may experience cold-start situations.

Our model, however, did not infer the relevance of the metadata as accurately as desired. This motivates further investigation into the cause of this behaviour, as the metadata used in the experiments conducted in this paper contains many obviously useless tags that were not labeled as such by the proposed method. Should this be remedied, we believe the proposed model would attain even greater recommendation accuracy.

References

1) Adomavicious, G. and Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, *IEEE Transactions on Knowledge and Data Engineering*, Vol.17, No.6, pp.734–749 (2005).
2) Blei, D. and Jordan, M.: Modeling Annotated Data, *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.127–134 (2003).
3) Blei, D., Ng, A. and Jordan, M.: Latent Dirichlet Allocation, *Jounal of Machine Learning Research*, Vol.3, pp.993–1022 (2003).
4) Cantador, I., Konstas, I. and Jose, J.: Categorising social tags to improve folksonomy-based recommendations, *Journal of Web Semantics*, Vol.9, No.1, pp. 1–15 (2011).
5) Hofmann, T.: Probabilistic Latent Semantic Analysis, *UAI '99: Proceedings of 15th Conference on Uncertainty in Artificial Intelligence*, pp.289–296 (1999).
6) Iwata, T., Yamada, T. and Ueda, N.: Modeling Social Annotation Data with Content Relevance using a Topic Model, *Proceedings of NIPS 2009*, pp.835–843 (2009).
7) Jeon, J., Lavrenko, V. and Manmatha, R.: Automatic Image Annotation and Retrieval using Cross-Media Relevance Models, *SIGIR '03 Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.119–126 (2003).
8) Nakatsuji, M., Fujiwara, Y., Uchiyama, T. and Fujimura, K.: User Similarity from Linked Taxonomies: Subjective Assessments of Items, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 2305–2311 (2011).
9) Pazzani, M. and Billsus, D.: Content-based Recommendation Systems, *Lecture Notes in Computer Science*, Vol.4321, Springer-Verlag, pp.325–341 (2007).
10) Popescul, A., Ungar, L., Pennock, D. and Lawrence, S.: Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments, *UAI 2001 Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence 2001*, pp.437–444 (2001).
11) Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews, *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pp.175–186 (1994).
12) Schafer, J., Konstan, J. and Reidl, J.: E-Commerce Recommendation Applications, *Data Mining and Knowledge Discovery*, Vol.5, pp.115–153 (2001).
13) Schein, A., Popescul, A., Ungar, L. and Pennock, D.: Methods and Metrics for Cold-Start Recommendations, *SIGIR '02 Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.253–260 (2002).
14) Yoshii, K., Goto, M., Komatani, K., Ogata, I. and H.G., O.: An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model, *IEEE Transactions on Audio, Speech, and Language Processing*, Vol.16, No.2, pp.435–447 (2008).