Predicting Future Reviews: Sentiment Analysis Models for Collaborative Filtering

Noriaki Kawamae
Tokyo Denki University
2-2 Kanda-Nishiki-cho, Chiyoda-ku, Tokyo, Japan
Japan 101-8457
kawamae@gmail.com

ABSTRACT

This paper presents hierarchical topic models for integrating sentiment analysis with collaborative filtering. Our goal is to automatically predict future reviews for a given author from previous reviews. For this goal, we focus on differentiating author's preference, unlike previous sentiment analysis models which process review articles without this difference. We propose a Latent Evaluation Topic model (LET) that infers each author's preference by introducing novel latent variables into author and his/her document layer. Because these variables distinguish the variety of words in each article by merging similar word distributions, LET incorporates the difference of writers' preferences into sentiment analysis. Consequently, LET can determine the attitude of writers, and predict their reviews based on like-minded writers' reviews in the collaborative filtering approach. Experiments on review articles show that the proposed model can reduce the dimensionality of reviews to the low-dimensional set of these latent variables; it represents a significant improvement over standard sentiment analysis models and collaborative filtering algorithms.

Categories and Subject Descriptors

H.3.3 [**Information filtering**]: Information Search and Retrieval

General Terms

Algorithms, experimentation

Keywords

Topic Modeling, Sentiment Analysis, Latent Variable Modeling, Information Extraction, Collaborative Filtering

WSDM'11, February 9–12, 2011, Hong Kong, China.

1. INTRODUCTION

Collaborative filtering (CF) aims to improve the user's experience and discovery against the potentially overwhelming set of choices. The underlying assumption of CF algorithms is that a good way to find interesting items for a given target user is to find other people who have similar interests, and then recommend items that these people like [18]. CF is used in web-based services, where there are vast numbers of items such as Web pages, digital music, and video content for browsing (download or purchase), and has been regarded as one of the most promising recommendation algorithms.

In this paper, we aim to predict an individual's future reviews by integrating CF and sentiment analysis. We focus on the problem of how to infer each author's preference from consumer-generated content and foretell his/her attitude to unseen content. Since more and more people are expressing their opinions on the Web in various ways, such as customer reviews, forums, discussion groups, and Web logs, people can gain useful information that will influence their behavior. The volume of these reviews is growing so rapidly that it is becoming increasingly difficult for users to wade through all the reviews and find the information that they need. Therefore, more focus is being placed on systems that produce fine-grained sentiment analysis of these articles [5], [17], [22] for extracting the desired information automatically. These systems are designed to detect the aspects discussed in the review articles and predict the sentiment of the writer towards each item. In many cases, the writer's preferences underlie this sentiment and different writers will express different sentiments toward the same item. For example, although all writers praise "Dark Knight", some emphasize "Story" while others highly rate "Casting". If we can elicit the writers' preferences underlying their positive and negative opinions in their reviews, we can predict future reviews based on these preferences in the same way as CF predicts user behavior from user history log data.

Although differentiating user preference is one of the most difficult tasks in CF algorithms, a few studies have attempted to capture the positive and negative sentiments present in Web sources. For example, Topic Sentiment Mixture (TSM) [14] captures the latent topical facets in Web log collections by modeling the mixture of topics and sentiment simultaneously. Multi-grain Topic Model (MG-LDA) [20] extends the conventional topic model by using sliding windows to induce multi-grain topics. These models extract information at the document level and thus process these articles without regard to the writer. Therefore, these models fail to infer the preferences of each writer that underlie these polarity sentiments or indeed to use the uniqueness of the preferences; each item polarity that can be extracted consists of a mixture of various preferences. An analysis of the sentiments associated with groups of like-minded writers would be much more useful than an analysis using the "average" preference of all writers. Generating future reviews of a given writer by integrating this model with CF would be possible if such an analysis model were available.

In this paper, we propose the Latent Evaluation Topic model (LET), a novel sentiment analysis model that simultaneously infers topics, sentiment, and preference in reviews. LET finds conditional relationships between these latent variable representations, and describes the generative process of writing reviews. This model is based on the assumption that a review article is generated by sampling words from a mixture model involving a background topic, a category, and a sentiment, since each article consists of background words, category words, and sentiment words. For distinguishing these differences and describing this generative process, our model defines a category class, a sentiment class, a preference class, and a switch variable as novel latent variables. The category class has a probability distribution over items/topic classes. The sentiment class has a probability distribution over topics/a beta distribution over ratings, and is assigned to each article. The preference class has a probability distribution over category/sentiment classes, and is assigned to each review. The switch variable has a probability distribution over the category topic, sentiment topic and the background topic, and is assigned to each token. Since this switch variable handles other latent variables for generating words in each token, this model explains the generative process by distinguishing these variables and then sampling words.

A key advantage of LET is that it integrates sentiment analysis and CF. LET distinguishes a writers' preference by using the difference in preference classes, predicts which item each writer will select and how he/she will evaluate it, and then recommends items to users based on this predicted reviews. More precisely, since this preference class denotes category and corresponding sentiment in each review document, this class allows LET to predict the items a writer will discuss in his/her article, and the corresponding sentiment, simultaneously. LET can then automatically generate review articles by using like-minded writers' articles, and so is unlike previous CF approaches that merely predict either item or rating by using like-minded history logs.

2. RELATED WORK

Several previous studies have summarized sentiment by extracting and aggregating sentiments over ratable aspects. Rather than modeling the writer's preference and topics simultaneously, all these approaches capture only topics, regardless of the difference in the writers' preferences [14], [20].

On the contrary, collaborative filtering approaches focus on not only distinguishing this difference, but also incorporating the content of items. For example, LISM [10] uses movie data with movie rating history and solves the cold start problem. Accordingly, collaborative filtering could be applicable to even more domains and services by introducing new kinds of data.

Since review articles and blog data are assumed to express their writer's preference, it seems likely that capturing preferences from these documents would be helpful in extending the applicable field of Social Feed. Social Feed is a personalization service based on applying collaborative filtering to contents or documents [2], [11]. For example, this service personalizes the news articles presented to each user based upon the news this user subscribes to and how this user interacts with it: the service helps the user discover relevant news and news sources. Experiments have shown that LET can differentiate various aspects by using the difference in writers' preferences, and thus improves the quality of personalization. Accordingly, LET allows us to process user articles, while conventional collaborative filtering techniques restrict themselves to user history logs. LET will yield highly effective social feed services.

3. PROBLEM FORMULATION

Our work is focused on modeling the latent semantic relationship between review articles and their writers. Table 1 lists the notations used in this paper. In this paper, each article d is associated with a writer, each item m, its corresponding rating v, and description $\mathbf{w}$. This article is represented as the triple of variables(m, v, $\mathbf{w}$). If an article contains multiple regions, we split this article into a single region associated with one item and deal with this region as an article. The aim of analysis is to capture low-dimensional probabilistic relationships among these observed variables and then predict which m will be selected, the value of v given to m, and which words $\mathbf{w}$ are used for describing this selected m in each article.

As an introduction to our model, we explain the concept of topic models. The basic assumption is that words in a document are generated according to a mixture model where the mixing proportions are document specific and are allowed to vary across the documents. These models represent words that are commonly used across the documents associated with each topic as a latent variable that denotes topic variable z, each with its own probability distribution over words ϕ_z at the document level. For example, terms $\mathbf{w}$ such as "Clint Eastwood", "Robert Patrick", "Roy Scheider", "Sir Thomas Sean Connery", and "Sir Maurice Joseph Micklewhite Jr." appear often in documents mentioning film actors, and so can be represented as the one topic of "film actors" z. Accordingly, these topic models allow each document to be identified using a latent variable representation that has lower dimensionality than the naive representation based on words. More precisely, this approach represents each document as a vector of topic count instead of word count. Consequently, a word in a document is sampled from the mixture of topics. Such models provide useful descriptive statistics for documents and have been applied for information retrieval [21], social network analysis [13] and collaborative filtering [12].

Since reviews contain more kinds of observed variables and these variables' relationships are generated by the writer, we need more types of latent variables and more layers for describing the latent relationship. For example, some reviews highly praise visual effects such as "Special Effects" and "Costume Design" in action movies such as "The Dark Knight" or "Iron Man", while others complain about the "casting" and "story" of the same movies. Among reviews, "Special Effects", "Costume Design", "cast" and "story" are topics that correlate with specific items, whereas "amazing" and "boring" are topics that correlate with items' rating and are reused in very different types of item reviews. Here, writers present their evaluation by rating value v in their reviews, too. Therefore, we define a category class, c, a sentiment class, s, and a preference, e, as novel latent variables to distinguish the differences in topics and explore co-occurrences at both the document and user level.

Definition: Category Class c: The category class is responsible for generating m jointly with the topic occurrence patterns in each document, where each topic is responsible for generating words, $\mathbf{w}$, associated with specific domains in the same way as previous models in each token. The hypothesis is that articles commenting on specific items with similar topics are represented by the same category. In this previous example, "Special Effects", "Costume Design", "cast" and "story" are category class specific words.

Definition: Sentiment Class s: The sentiment class is responsible for generating v jointly with the topic occurrence patterns in each document, where each topic is responsible for generating words. This assumes that articles commenting on specific ratings for similar topics are represented by the same sentiment. For example, sentiment words are generated via topics associated with sentiment class and used mainly to express the writer's sentiments on topics and are reused for very different types of items, whereas topic words are used for expressing a specific theme and are thus reused in the same document. In this previous example, "amazing" and "boring" are sentiment class specific words.

Definition: Preference Class e: The preference class is responsible for generating both category class c and corresponding sentiment class s in the document layer. We hypothesize that articles giving similar sentiment on a specific category are represented by the same preference class. Accordingly, this class generates both an item and its corresponding category specific words via the category class.

Conditional on the values of these latent variables and their hierarchical structure, the feature vectors of articles (m, v, $\mathbf{w}$) are assumed to be distributed over latent classes c, s, and z respectively.

We are interested in models that can perform three tasks: 1) modeling the joint distribution of an item and words, 2) modeling the joint distribution of a category and its rating, and 3) modeling the conditional joint distribution of a category and its rating given by the writers. The first task can be useful for automatic item annotation and text-based item retrieval. The second is useful for clustering and organizing categories in view of preference. The third is useful for collaborative filtering and social feed services.

Table 1: Notations used in this paper

SYMBOL	DESCRIPTION
A	number of writers
G	number of writer classes
E	number of preference classes
S	number of sentiment classes
C	number of category classes
Z	number of topics
D	number of documents
M	number of items
W	number of unique words
D_a	number of documents written by writer a
N_d	number of word tokens in document d
d_a	the document associated with writer a
g_a	the writer class associated with writer a
e_d	the preference class associated with writer a
s_d	the sentiment class associated with document d
c_d	the category class associated with document d
m_d	the item discussed in document d
v_d	the rating in document d
r_{di}	the switch variable associated with the ith token in document d
z_{di}	the topic associated with the ith token in document d
w_{di}	the ith token in document d
ν	the multinomial distribution of writer classes ($\nu\|\iota \sim$ Dirichlet(ι))
ψ_g	the multinomial distribution of preference classes specific to writer class g ($\psi_g\|\alpha \sim$ Dirichlet(α))
θ_e	the multinomial distribution of category classes specific to preference class e ($\theta_e\|\beta \sim$ Dirichlet(β))
λ_s	the beta distribution specific to sentiment class s
χ_e	the multinomial distribution of sentiment classes specific to preference class e ($\chi_e\|\zeta \sim$ Dirichlet(ζ))
ω_c	the multinomial distribution of items specific to category class c ($\omega_c\|\epsilon \sim$ Dirichlet(ϵ))
$\varphi_{c(s)}$	the multinomial distribution of topics specific to category class c(sentiment class s) ($\varphi_{c(s)}\|\gamma \sim$ Dirichlet(γ))
μ_d	the multinomial distribution of the switch variable specific to article d($\mu_d\|\eta \sim$ Dirichlet(η))
$\phi_{z(b)}$	the multinomial distribution of words specific to topic z(background topic b)($\phi_{z(b)}\|\beta \sim$ Dirichlet(δ))
$\alpha, \beta, \zeta, \gamma$ $\delta, \epsilon, \eta, \iota$	hyper parameter

4. LATENT EVALUATION TOPIC MODEL

4.1 Modeling sentiment and preference

In this subsection, we describe our latent variable model in detail; it identifies not only low dimensional, but also interpretable components in review articles for describing how like-minded writers evaluate items. First, we propose simple Latent Evaluate Topic (sLET) which uses preference class e (Only in this model is preference class responsible for generating m jointly with v), and latent switch variable r to model item m with rating v. The discrete latent variable e is used to represent a joint clustering of item m, its corresponding v, and topic z. This preference class, e, is then sampled from ψ_a, and is held fixed during the process of generating item m, its corresponding rating v, and topic z, from ω_e, λ_e and φ_e, respectively, in each article d_a. Since the Beta distribution can take versatile shapes, we use this to describe the rating v associated with the sentiment class in each document, where all ratings are normalized from 0 to 1. Additionally, while generating its components, the preference class indirectly generates words via topics in each token as per previous models. Latent switch variable r acts as a switch to handle words in each token, taking value r=0 if word w is generated via the background topic variable, and r=1 if word w is generated via topic. The background topic is the common topic over almost all reviews regardless of their content and such background topic words can exist together with various topic words in each document. For example, we call "abstract", "figure" or "reference" background topic words, since these words appear over all documents (papers). This model achieves simultaneous dimensionality

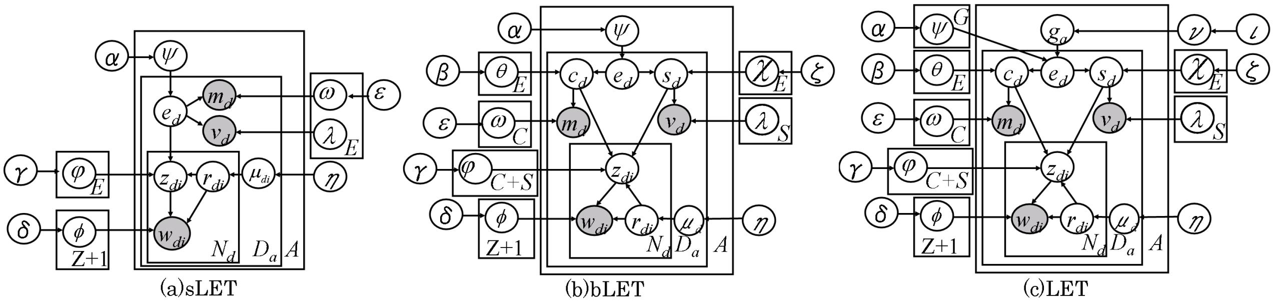

Figure 1: Graphical Model of LET: In this figure, shaded and unshaded variables indicate observed and latent variables, respectively. An arrow indicates a conditional dependency between variables and the stacked panes indicate repeated sampling with the iteration number shown. Since these models are also based on the bag of words assumption, they can represent each document as an unordered collection of words, where the grammar and even word order are disregarded.

reduction in the presentation of an item and its rating, while also modeling the conditional correspondence between their respective reduced representations via the preference class. This preference class allows sLET to generate words via topics and assign ratings to any given item. Consequently, we adopt the hierarchical structure to describe the generative process for distinguishing these variables, and such words by selecting topics in each token.

Second, we extend sLET by introducing both category class c and sentiment class s, and call this model basic Latent Evaluate Topic (bLET). The single discrete latent variable c represents the pairing of item m with category specific topics $\mathbf{z}$. The single discrete latent variable s represents the pairing of rating v with sentiment specific topics $\mathbf{z}$. bLET focuses on classifying words into background topic words, sentiment words, and other words in each token, while sLET classifies words into background topic words or other words. Therefore, we extend r as a switch for handling more kinds of words, taking value r=0 if word w is generated via the background topic variable, r=1 if word w(category words) is generated via the topic variable associated with category class c, and r=2 if word w(sentiment words) is generated via the topic variable associated with sentiment class s. In this model, the preference class generates topics via category class/sentiment class, and then words via these topics.

Finally, we extend bLET to a fully generative model, Latent Evaluation Topic (LET), by introducing writer class [7]. Here, we hypothesize that writer class can be represented as a set of preference classes in the same way that document contents are represented as sets of topics in topic models, and then can be classified by using the differences between these sets. As shown in the figure, writer class g is sampled from ν for each writer a. As a result, LET represents writer associated with similar preference classes via state variable g, while bLET gives them similar distributions ψ.

4.2 Inference and Learning

4.2.1 Generative process

A description of the generative process of LET for parameter estimation and item/its rating/words(m,v,$\mathbf{w}$), explained in the previous subsection and shown in Figure 1, assumes the following generative process:

1. Draw multinomial ν from Dirichlet prior ι;
2. Draw G multinomials ψ_g from Dirichlet prior α, one for each preference g;
3. Draw D multinomials μ_d from Dirichlet prior η, one for each document d;
4. Draw E multinomials θ_e from Dirichlet prior β, one for each preference class e;
5. Draw E multinomials χ_e from Dirichlet prior ζ, one for each preference class e;
6. Draw S beta distributions, one for sentiment class s;
7. Draw C multinomials ω_c from Dirichlet prior ε, one for each category c;
8. Draw $C+S$ multinomials φ_{c+s} from Dirichlet prior γ, one for each sentiment s/category class c;
9. Draw $Z+1$ multinomials ϕ_z from Dirichlet prior δ, one for each topic z or background topic;
10. For each writer a:
    (a) Draw writer class g_a from multinomial ν;
    (b) For each review d:
        i. Draw preference class e_d from multinomial ψ_g;
        ii. Draw category class c_d from θ_e;
        iii. Draw sentiment class s_d from χ_e;
        iv. Draw item m_d from ω_c;
        v. Draw rating v_d from beta λ_s;
        if $r = 0$
        A. Draw word w_{di} from multinomial ϕ_b.
        else if $r = 1$
        A. Draw topic z_{di} from multinomial φ_{c_d};
        B. Draw word w_{di} from multinomial $\phi_{z_{di}}$.
        else if $r = 2$
        A. Draw topic z_{di} from multinomial φ_{s_d};
        B. Draw word w_{di} from multinomial $\phi_{z_{di}}$.

Since the generative process of LET can be described as a Bayesian hierarchical model, we employ Gibbs sampling to perform approximate inference in this model. In this inference, we need to calculate the conditional distribution. We begin with the joint distribution of the entire corpus as follows:

$$
\begin{aligned}
&p(\mathbf{w},\mathbf{z},\mathbf{r},\mathbf{m},\mathbf{c},\mathbf{v},\mathbf{s},\mathbf{e},\mathbf{g},\phi,\varphi,\mu,\omega,\theta,\chi,\psi,\nu|\alpha,\beta,\zeta,\gamma,\delta,\epsilon,\eta,\lambda,\iota)\\
&= p(\mathbf{w},\phi|\mathbf{z},\delta)p(\mathbf{z},\varphi|\mathbf{r},\mathbf{c},\mathbf{s},\gamma)p(\mathbf{r},\mu|\eta)p(\mathbf{m},\omega|\mathbf{c},\varepsilon)p(\mathbf{c},\theta|\mathbf{e},\beta)\\
&\times p(\mathbf{g},\nu|\iota)p(\mathbf{v}|\lambda,\mathbf{s})p(\mathbf{s},\chi|\mathbf{e},\zeta)p(\mathbf{e},\psi|\mathbf{g},\alpha) = \prod_a^A P(g_a|\nu)\\
&\times \prod_d^D [P(c_d|\theta_{e_d})P(s_d|\chi_{e_d})P(e_d|\psi_{g_a}))p(v_d|\lambda_{s_d})P(m_d|\omega_{c_d})]\\
&\times \prod_i^{N_d}[P(w_{di}|\phi_{z_{di}})P(z_{di}|r_{di},\varphi_{c_d},\varphi_{s_d})P(r_{di}|\mu_d)]\prod_d^D p(\mu_d|\eta)\\
&\times\prod_g^G p(\psi_g|\alpha)\prod_e^E p(\theta_e|\beta)p(\chi_e|\zeta)\prod_c^C p(\varphi_c|\gamma)p(\omega_c|\varepsilon)\prod_s^S p(\varphi_s|\gamma)\\
&\times \prod_z^{Z+1} p(\phi_z|\delta).
\end{aligned} \tag{1}
$$

In this eq (1), multinomials ν, ψ_g, θ_e, χ_e, ω_c, φ_c, φ_s, μ_d and ϕ_z can be adapted by the conjugate prior and then integrated out analytically as described in the Appendix. In the Gibbs sampling procedure, we need to calculate the conditional distributions $P(g_a|\mathbf{g}_{\backslash a},\mathbf{w},\mathbf{z},\mathbf{r},\mathbf{m},\mathbf{c},\mathbf{v},\mathbf{s},\mathbf{e},\alpha,\beta,\zeta,\gamma,\delta,\epsilon,\eta,\lambda,\iota)$, $P(e_d|g_a,\mathbf{e}_{\backslash d},\mathbf{w},\mathbf{z},\mathbf{r},\mathbf{m},\mathbf{c},\mathbf{v},\mathbf{s},\alpha,\beta,\zeta,\gamma,\delta,\epsilon,\eta,\lambda,\iota)$, $P(s_d|\mathbf{e}_a,\mathbf{s}_{\backslash d},\mathbf{w},\mathbf{z},\mathbf{r},\mathbf{m},\mathbf{c},\mathbf{v},\alpha,\beta,\zeta,\gamma,\delta,\epsilon,\eta,\lambda,\iota)$,$P(c_d|e_a,\mathbf{c}_{\backslash d},\mathbf{w},\mathbf{z},\mathbf{r},\mathbf{m},\mathbf{v},\mathbf{s},\alpha,\beta,\zeta,\gamma,\delta,\epsilon,\eta,\lambda,\iota)$, and $P(r_{di},z_{di}|c_d,\mathbf{z}_{\backslash di},\mathbf{r}_{\backslash di},\mathbf{w},\mathbf{m},\mathbf{v},\alpha,\beta,\zeta,\gamma,\delta,\epsilon,\eta)$. In these distributions, $\mathbf{e}_{\backslash a}$ represents the preference class assignments for all writers except writer a, $\mathbf{s}_{\backslash d}$ represents the sentiment class assignments for all documents except document d, $\mathbf{c}_{\backslash d}$ represents the category class assignments for all documents except document d, $\mathbf{r}_{\backslash di}$ represents the switch variable assignments for all tokens except r_{di} and $\mathbf{z}_{\backslash di}$ represents the topic assignments for all tokens except z_{di}. A detailed derivation of Gibbs sampling for LAT is given in the following.

4.2.2 Writer class

For each writer, we compute the writer assignment $P(g_a|\mathbf{g}_{\backslash a},\mathbf{w},\mathbf{z},\mathbf{r},\mathbf{m},\mathbf{c},\mathbf{v},\mathbf{s},\mathbf{e},\alpha,\beta,\zeta,\gamma,\delta,\epsilon,\eta,\lambda,\iota)$. We use the chain rule and thus obtain the conditional distribution of writer class to f, is given by $P(g_a=f|\mathbf{g}_{\backslash a},\mathbf{e},\alpha,\iota)$ as

$$
\begin{aligned}
&P(f|\cdots)\\
&\propto \frac{n_{f\backslash a}+\iota_f}{\sum_g^G(n_{g\backslash a}+\iota_g)}\frac{\Gamma(\sum_e^E n_{fe\backslash a}+\alpha_e)}{\prod_e^E\Gamma(n_{fe\backslash a}+\alpha_e)}\frac{\prod_e^E\Gamma(n_{fe}+\alpha_e)}{\Gamma(\sum_e^E n_{fe}+\alpha_e)},
\end{aligned} \tag{2}
$$

where $n_{f\backslash a}$ represents the number of writers placed in f, except a, and $n_{fe\backslash a}$ represents the number of documents associated with e in all documents by writers placed in f, except a. LET represents writers having articles with similar content as the same writer class by merging these writers, where this class is an indicator variable that describes which probability distribution over preference classes each writer class uses in each document.

4.2.3 Preference class

For each document, we compute the preference assignment $P(e_d|\mathbf{e}_{\backslash d},\mathbf{w},\mathbf{z},\mathbf{r},\mathbf{m},\mathbf{c},\mathbf{v},\mathbf{s},\alpha,\beta,\zeta,\gamma,\delta,\epsilon,\lambda,\eta)$. We use the chain rule and thus obtain the conditional distribution of preference class to h, is given by $P(e_d=h|f,\mathbf{e}_{\backslash d},\mathbf{c},\mathbf{s},\alpha,\beta,\zeta)$ as

$$
\begin{aligned}
&P(h|\cdots)\propto \frac{n_{fh\backslash d}+\alpha_h}{\sum_e^E(n_{fe\backslash d}+\alpha_e)}\frac{\Gamma(\sum_c^C n_{hc\backslash d}+\beta_c)}{\prod_c^C\Gamma(n_{hc\backslash d}+\beta_c)}\\
&\times\frac{\prod_c^C\Gamma(n_{hc}+\beta_c)}{\Gamma(\sum_c^C n_{hc}+\beta_c)}\frac{\Gamma(\sum_s^S n_{hs\backslash d}+\zeta_s)}{\prod_s^S\Gamma(n_{hs\backslash d}+\zeta_s)}\frac{\prod_s^S\Gamma(n_{hs}+\zeta_s)}{\Gamma(\sum_s^S n_{hs}+\zeta_s)},
\end{aligned} \tag{3}
$$

where $n_{fh\backslash d}$ represents the number of documents associated with h in all documents by writers placed in f, except d, $n_{hc\backslash d}$ represents the number of documents associated with c in all documents placed in h, except d, and $n_{hs\backslash d}$ represents the number of documents associated with s in all documents placed in h, except d. Since each category, c, and its sentiment, s, are assumed to have been generated conditional on preference class e_a associated with the corresponding writer, a, an item with high probability under a certain category class will likely contain sentiments that express, with high probability, the same preference.

4.2.4 Category class

For each document, the predictive distribution of adding category class c_d, in document d written by writer placed in h, to the category class l, is given by $P(c_d=l|h,\mathbf{c}_{\backslash d},\mathbf{z},\mathbf{m},\beta,\gamma,\epsilon)$ as

$$
\begin{aligned}
&P(l|\cdots)\\
&\propto\frac{n_{hl\backslash d}+\beta_l}{\sum_c^C(n_{hc\backslash d}+\beta_c)}\frac{n_{lm\backslash d}+\epsilon_m}{\sum_m^M(n_{lm\backslash d}+\epsilon_m)}\frac{n_{lz\backslash d}+\gamma_z}{\sum_z^Z(n_{lz\backslash d}+\gamma_z)},
\end{aligned} \tag{4}
$$

where $n_{hl\backslash d}$ represents the number of documents associated with category class l in all documents placed in h, except d, $n_{lm\backslash d}$ represents the number of item m in all documents placed in l, except d, and $n_{lz\backslash d}$ represents the number of tokens assigned to topic z in the documents placed in l, except d. Intuitively, items that have high probability under a certain category class will appear with topics that have a high probability on the same category class.

4.2.5 Sentiment class

Like eq (4), the predictive distribution of sentiment class s_d in document d placed in h to sentiment class j, is given by $P(s_d=j|h,\mathbf{s}_{\backslash d},\mathbf{z},\mathbf{v},\zeta,\gamma)$ as

$$
\begin{aligned}
&P(j|\cdots)\\
&\propto\frac{n_{hj\backslash d}+\zeta_s}{\sum_s^S(n_{hs\backslash d}+\zeta_s)}\frac{(1-v_d)^{\lambda_{j1}-1}v_d^{\lambda_{j2}-1}}{B(\lambda_{j1},\lambda_{j2})}\frac{n_{jz\backslash d}+\gamma_z}{\sum_z^Z(n_{jz\backslash d}+\gamma_z)},
\end{aligned} \tag{5}
$$

where $n_{hj\backslash d}$ represents the number of documents associated with sentiment class j in all documents placed in h, except d, $n_{jz\backslash d}$ represents the number of tokens assigned to topic z in all documents placed in j, and $B(\cdot)$ is the beta function. Intuitively, the high probability rating value under the sentiment class suggests which words this class captures via topics associated with this sentiment class. In a similar way, a topic with high probability under a sentiment class will likely receive rating with high probability in the same sentiment class category.

4.2.6 Switch variable & Topic

For each token, the predictive distribution of adding word w_{di} in a region of article d to background topic is $P(r_{di} = 0|\mathbf{z}_{\setminus di}, \mathbf{r}_{\setminus di}, \mathbf{w}, \delta, \eta)$ as

$$P(0|\cdot\cdot\cdot) \propto \frac{n_{d0\setminus di} + \eta_0}{\sum_r^R (n_{dr\setminus di} + \eta_r)} \frac{n_{bv\setminus di} + \delta_v}{\sum_w^W (n_{bw\setminus di} + \delta_w)}, \quad (6)$$

where $n_{d0\setminus di}$ represents the number of tokens assigned to switch variable $r = 0$(background topic) in document d, except di, and $n_{bv\setminus di}$ represents the number of words, v, in background topic, except token di.

Similarly, the predictive distribution of adding word w_{di} in document d to topic k is $P(r_{di} = 1, z_{di} = k|l, \mathbf{z}_{\setminus di}, \mathbf{r}_{\setminus di}, \mathbf{w}, \gamma, \delta, \eta)$ as

$$P(1, k|\cdot\cdot\cdot) \propto \frac{n_{lk\setminus di} + \gamma_k}{\sum_z (n_{lz\setminus di} + \gamma_z)} \frac{n_{d1\setminus di} + \eta_1}{\sum_r (n_{dr\setminus di} + \eta_r)} \frac{n_{kv\setminus di} + \delta_v}{\sum_w (n_{kw\setminus di} + \delta_w)}, \quad (7)$$

where $n_{d1\setminus di}$ represents the number of tokens assigned to switch variable $r = 1$ (topic associated with category class) in document d, except d_i, $n_{lk\setminus di}$ represents the number of tokens assigned to topic k in documents associated with l, except d_i, and $n_{kv\setminus di}$ represents the number of word v in tokens associated with topic k, except token d_i.

Like eq (7), the predictive distribution of adding word w_{di} in document d to topic k is $P(r_{di} = 2, z_{di} = k|j, \mathbf{z}_{\setminus di}, \mathbf{r}_{\setminus di}, \mathbf{w}, \gamma, \delta, \eta)$ and is written as

$$P(2, k|\cdot\cdot\cdot) \propto \frac{n_{jk\setminus di} + \gamma_k}{\sum_z (n_{jz\setminus di} + \gamma_z)} \frac{n_{d2\setminus di} + \eta_2}{\sum_r (n_{dr\setminus di} + \eta_r)} \frac{n_{kv\setminus di} + \delta_v}{\sum_w (n_{kw\setminus di} + \delta_w)}, \quad (8)$$

where $n_{d2\setminus di}$ represents the number of tokens assigned to switch variable $r = 2$ (topic associated with sentiment class) in document d, $n_{jk\setminus di}$ represents the number of tokens assigned to topic k in documents associated with j, except d_i, and $n_{kv\setminus di}$ represents the number of word v in tokens associated with topic k, except token d_i.

Accordingly, the review associated with same category class is likely, with high probability, to exhibit similar items, corresponding sentiment classes, and words via topics associated with the same category/sentiment class. Therefore, this model is appropriate for predicting the rating given to the item in an article.

5. EXPERIMENTS

5.1 Experiments Design and Data

We focus here on detecting groups of like-minded writers and determining the attitude of each group from a given review article; both qualitative and quantitative evaluations of the proposed model are presented. In the qualitative analysis, we show that sentiment words, words associated with topics as inferred by LET correspond to the user's evaluation of each item in each review. In the quantitative analysis, we show that the sentiment class generated from review articles by our model can significantly improve the quality of generated reviews. We used two data sets: (1) Data1: We extracted 5000 movie review articles from MSN movie [3] as the corpus. Each article consisted of reviewer name or id, movie title, its rating(0-100) and review text. After removing stop words, numbers, and the words that appeared less than five times in the corpus, we obtained a total set of 212 reviewers, 23564 documents, and 52349 unique words. (2)Data2: We used Last.fm data set [4]. Each play list consisted of user id, music title, its play count and tag. After splitting data, we obtained a total set of 10000 music titles, 3344 users, and 68335 unique words.

In these experiments, we set the number of sentiment classes, S, to 3, since pre experiments confirmed that this separation, (high/medium/low), was enough for the data used. In each review(playlist), about 83% of all tokens are occupied by a few kinds of topic class and many of these topics are reused in other reviews(playlist). Consequently, we do not need many category/sentiment classes, and we confirmed, from pre experiments, that setting $|C| \approx |Z|$, $|G| \approx |E|$ was sufficient for tracking category/sentiment classes and writer/preference,. The smoothing parameters α, β, ζ, γ, δ, ϵ, η, ι were set at 1/E, 1/C, 1/S, 1/(C+S), 1/W, 1/M, 1/3, 1/G, respectively, i.e. all symmetric hyper parameters as in previous works. We performed experiments on machines with Dual Core 2.66 GHz Xeon processors and iterated each single Gibbs sampling chain 10000 times.

Table 2: The ratio of switch variable r comparison on data sets: We learned these ratios under LET using the number of topics $Z = 100$ and preference class $E = 30$.

Data	$\{r = 0\}$	$\{r = 1\}$	$\{r = 2\}$
Data1	0.11	0.64	0.25
Data2	0.23	0.71	0.06

5.2 Qualitative Experiments

Table 2 shows the results of ratio r for the two data sets. This result shows that LET can distinguish the role of words in accordance with the difference in data. Last-fm data consists of only tags in documents, while MSN review articles are text with words. For example, a song is tagged by "indie->rock", "alternative rock->rock","alternative->indie" and "alternative->rock". Since these tags represent the song and are used for retrieval, they have a strong relationship with the song (item) in each document. Therefore, LET weights $r = 2$(sentiment class) close to 0.

Table 3 provides examples of movies and words under the category class learned by LET to compare the differences between the word distribution associated with category class and that distribution associated with sentiment class. These two selected two category classes have the most similar item distribution, as measured by KL divergence, among all pairs of category classes learned from the corpus. We manually assigned labels to category and topics to reflect our interpretation of their meaning. As we can see, these category classes commonly refer to similar movies and praise them. Interestingly, the difference between these classes is observed in the difference of their topics. The reviews assigned to

Table 3: Item and Topic distribution of category class and topics learned from the corpus: We list 10 movies by the learned probability from the category class under LET, and list 10 words by the learned probability from the topic under LET using the number of topics Z = 100, category class C = 100, and preference class E = 30. Each topic is decided by the highest value of topic distribution associated with each sentiment/category class. The value of each column is the corresponding probability.

Item	Sentiment	Category			
id 12	ID 3	ID 22	ID 39	ID 73	ID 87
The Dark Knight 0.32	much	opening	joker	imax	making
Iron ma 0.24	thriller	nolan	heath	texture	theme
Batman Begins 0.22	unbelievable	united	actor	image	ethical
Watchmen 0.13	theater	warner	ledger	photo	representation
Cat woman 0.12	criminal	counterpart	performance	3D	performance
X-men 0.11	interesting	christopher	action	really	friendship
Batman 0.11	best	ticket	robin	superhero	laws
Paychec 0.10	amazing	comic	eckhart	screen	escalation
The Avengers 0.09	warrior	DC	uma	quality	critical
Pulp Fiction 0.08	technology	popular	Bale	wide	midnight
Batman Returns 0.08	judge	New York	sound	chaos	commercial
id 27	ID 11	ID 39	ID 55	ID 87	ID 98
The Dark Knight 0.29	dream	joker	performed	making	oldman
Batman Begins 0.26	spectacular	heath	academy	theme	police
Terminator 4 0.22	brooding	actor	starring	ethical	role
Mask 0.15	technology	ledger	nominated	representation	freeman
Watchmen 0.14	experiment	performance	muscle	performance	together
The Avengers 0.14	incredible	action	demos	friendship	heroic
Batman 0.13	masterpiece	robin	personality	laws	costar
X-men 0.11	major	eckhart	role	escalation	childhood
Star trek 0.11	future	uma	formulating	critical	fatherly
Iron man0.10	impossible	Bale	hero	midnight	advice
Paycheck 0.08	excited	sound	images	commercial	rachel

category class id 12 are correlated to the topics of "cast(ID 39)", "IMAX effect(ID 73)" and "story & theme(ID 87)", and praise movies with regard to casts, effects and story, where these titles are our interpretation from the most likely words associated with each topic. On the contrary, the reviews assigned to category class id 27 are correlated with the topics of "cast(ID 89 & ID 98)", "acting(ID 55)", and praise movies with regard to these topics. Consequently, these tables show that LET can detect both the sentiment class and the category class from the mixture of words by using items and their value.

5.3 Quantitative Evaluation

To measure the ability of sLET, bLET and LET to act as generative models, we computed test-set perplexity under the estimated parameters and compared the resulting values, and KL divergence over gained topics.

5.3.1 *Perplexity*

First, we compared the models by perplexity, which is widely used in the language modeling community to assess the predictive power of a model; it is algebraically equivalent to the inverse of the geometric mean per-word likelihood (lower numbers are better). The perplexity of LET was computed for all models using 100 samples from 100 different chains using

$$PPX = \exp(-\frac{1}{W}\sum_{d\in D_{\text{test}}}\sum_{v\in d}\frac{1}{Y} \times \log(\sum_{z}^{Z}\sum_{c}^{C}\sum_{s}^{S} p(w|z,\delta)p(z|c,s,\gamma)p(c,s|e,\beta,\eta))), \tag{9}$$

where W is the number of test words and Y is the number of samples (from Y different chains).

We computed perplexity as follows. First, we randomly took 20% of each document as the test part and the remainder as the learning part. For every document, the test was held out to compute perplexity. Second, the learning part was used for estimating the parameters by Gibbs sampling. Finally, a single set of topic counts was saved when a sample was taken; the log probability of test words that had never been seen before was computed in the same way as the perplexity computation of previous works.

We compared sLET, bLET and LET against TSM and MG-LDA. This comparison is over only word distributions, since neither TSM nor MG-LDA can differentiate the type of words or indeed generate items and ratings. Table 4 shows the results of the perplexity comparison. These results are the averages over ten-fold cross validation.

From this table, we observe the following: (1)Variety: Under the same number of iterations, the value of perplexity decreases in inverse proportion to the number of sentiment

Table 4: Perplexity comparison of TSM, MG-LDA, sLET, bLET and LET: All models were learned by using the number of topics $Z = 100$, and the number of preference classes $E = 30$(Data1), 80 and 10(Data2), respectively . The numbers in the second row for MG-LDA and LET are the numbers of local topics(MG-LDA)/category class(bLET and LET), respectively. The value of Avg means the average of the computing time for each iteration in Gibbs sampling (seconds). The difference shown between TSM, MG-LDA and LET is significant according to the one-tailed t-test.

Data	Iteration	TSM	sLET	MG-LDA			bLET			LET		
				25	50	100	25	50	100	25	50	100
Data1	1000	1964	1732	1723	1705	1698	1679	1658	1612	1652	1625	1588^{**}
	5000	1756	1622	1665	1653	1632	1651	1621	1595	1631	1582	1465^{**}
	10000	1658	1532	1582	1577	1535	1493	1475	1422	1462	1406	1383^{**}
	Avg	33.2	25.8	27.2	29.4	31.3	32.5	34.6	36.1	33.3	35.6	37.2
				20	40	80	20	40	80	20	40	80
Data2	1000	321	334	313	311	307	302	298	296	298	297	296
	5000	311	310	309	302	297	292	291	288	287	285	285
	10000	303	296	298	295	293	291	289	285	285	283	279^{*}
	Avg	18.1	11.4	12.4	14.2	16.2	17.3	19.3	21.2	18.5	19.9	21.4

classes rather than that of topics. Moreover, LET tends to achieve low perplexity faster than TSM and MG-LDA. This implies that there are various review points(topics) even for the same item, as shown in the qualitative experiments. (2)Noise reduction: LET groups articles under the co-occurring topic distributions rather than permitting various topic distributions on each document. This implies that LET assigns fewer iterations to topics that are correlated to each other in each document than the other topics. Consequently, clustered documents contain less noise than would be otherwise true.

5.3.2 KL divergence

We also measured average topic distribution separations between different pairs of latent classes; the effect of the proposed latent classes is discussed. We measure this distance between classes by the average KL-Divergence. That is, a higher score implies that the distributions over topics are more distinct. This score allows us to measure how distinct the topics are for each model. Table 5 shows the results of the distance comparison.

Table 5: Average KL divergence between classes, topic distributions: All models were learned with the number of topics Z and the number of preference classes E set at 100 and 30(Data1), 80 and 10(Data2), respectively.

		sLET	bLET	LET
Data1	$D_{KL}(\varphi_c\|\varphi_{c'})$	-	2.6	2.7
	$D_{KL}(\varphi_s\|\varphi_{s'})$	1.6	4.7	4.8
	$D_{KL}(\varphi_c\|\varphi_s)$	-	3.9	4.1
Data2	$D_{KL}(\varphi_c\|\varphi_{c'})$	-	1.5	1.7
	$D_{KL}(\varphi_s\|\varphi_{s'})$	1.4	1.5	1.5
	$D_{KL}(\varphi_c\|\varphi_s)$	-	1.6	1.6

From this table, we observe the following: (1)Category and Sentiment class: The distance among topic distributions φ_z learned by LET is larger than that learned by sLET. This implies that the generative model of review articles requires both sentiment specific topics and category specific topics to be distinguished, since each article consists of item description and writer's impression. Accordingly, sentiment analysis models must allow for this difference. (2)Separation of topic distributions: The distance among topic distributions on different type classes(e.g., $D_{KL}(\varphi_{c_1}|\varphi_{s_1})$) is larger than that on same type classes(e.g., $D_{KL}(\varphi_{c_1}|\varphi_{c_2})$). Since LET achieves the highest KL-Divergence and lowest perplexity as shown in the above experiments, LET models review articles as a generative process by sampling words from a mixture model involving the background topic, sentiment and category topics. As shown in Table 2, Data2 has few topics associated with sentiment classes in each article, since these articles consist of tags that are annotations of songs rather than user evaluations. Consequently, the sentiment specific topic distribution became distinct and the KL-based distance was lowered.

5.3.3 Collaborative Filtering

To measure model quality in the task of capturing an individual's preference, we computed performance on the collaborative filtering task using the MSN data set. We conducted simulations to evaluate the predictive performance of recommendation via K cross-validation where the original data is partitioned into K sub samples at random. Each sub sample is set to test data and the remaining sub samples are set to learning data. In the simulations, we treated each user in the test data as a target user to whom we applied each of the recommendation methods by using user logs collected from the learning data. We then presented the top N ranked items to the target user according to the probability that each algorithm assigns to candidate items, and confirmed that these recommended items existed in the test data.

To evaluate the quality of the proposed method, we used both top N precision and mean absolute error (MAE), a measure commonly used for evaluating the performance of CF systems. We applied the proposed method, a baseline method and 5 conventional recommendation methods to the data sets and compared the precision of their top-N recommendations (N values were 1, 5 and 10). The baseline method is *Popular* which recommends the most popular

Table 6: Comparison of various methods: The number of topics Z, preference classes E, and writer classes G are fixed at 100, 30, and 15, respectively, all with 10000 iterations. Results that differ significantly t-test $p < 0.01$, $p < 0.05$ from the conventional methods are marked with '', '*' respectively.**

Evaluation	Popular	Item	MEA	IFD	sLET	bLET	LET
Top-1	9.53	7.79	8.67	9.72	10.69	9.64	9.71
Top-5	9.77	8.23	8.99	10.51	11.82^{*}	9.76	10.92
Top-10	10.23	8.87	9.22	11.03	13.22^{*}	11.36	12.53
MAE	12.16	12.32	13.62	9.35	11.22	9.31	9.22
User coverage	100	88.3	88.3	88.3	100	100	100
Item coverage	4.78	4.46	4.65	4.68	7.32	8.22	8.68^{**}
Gini coefficient	0.72	0.68	0.66	0.66	0.60	0.58	0.53^{**}

items during the last one month of the learning period and thus it is not personalized to the user. The conventional methods are as follows: *Pearson* and *Cosine* are based on user similarity, as measured by Pearson's correlation coefficient and cosine similarity, respectively. On the other hand, *Item* is based on content similarity measured by Pearson's correlation coefficient as proposed in [1]. *MEA* is the maximum entropy approach proposed in [16]. *IFD* is the collaborative filtering for implicit feedback data sets [6]. Since these models require history logs, we extracted the triple of variables(user, movie and corresponding rating) from the data. sLET, bLET and LET ranked items by the high order of estimated ω on a given category class c that is learned from the history log on condition that rating v exceeds the average of each user rating, while the other methods ranked items by high order of estimated rating.

Additionally, we also evaluated the performance in terms of "Item coverage" (the percentage of the number of unique items appearing in the top-N recommendation list over the number of total unique items), "User coverage" (the percentage of the number of users to whom each method can recommend any item over the number of all users who purchased any item in the test period), and the Gini coefficient (a measure of the statistical dispersion of the distribution of users over items) and show these results in Table 5.3.2. A high User/Item indicates that many users/items are recommended/presented, respectively. The Gini coefficient is a measure of the statistical dispersion of the distribution of users over items and is defined as the ratio of the areas on the Lorenz curve diagram; it takes values between 0 and 1. A low Gini coefficient indicates that the distribution is flat, while a high Gini coefficient indicates that the distribution is extremely biased: 0 corresponds to perfect equality (every item has been purchased by exactly the same number of users) and 1 corresponds to perfect inequality (where one item is purchased by all users, while none of the other items are purchased by any user). In other words, a result with a high Gini coefficient means that a few particular items tend to be ranked highly for most users and thus recommendations are not personalized.

We can see that LET exhibits the highest item coverage and the lowest Gini coefficient, closer to 0 than the others. This result indicates that LET can rank and recommend a wide range of items with much less bias than the other methods. In fact, those conventional algorithms do not consider which attributes each user refers to in praising each item, so that achieve lower item coverage and higher Gini coefficients, closer to 1 than the others. For example, popular movies such as "The Day After Tomorrow" and "Miss Congeniality" are not good indicators for distinguishing the differences of these attributes. Since there exists users(items) that have no common items(users), the similarity based methods can not gain 100% in User/Item coverage. LET focuses on the difference of sentiment in like-minded users, which means the item attributes discussed in each review, and such ranks items that match both the preferences and the sentiment. Although sLET gained high precision, it could recommend trivial/popular items. Consequently, LET allows CF to be performed on review content instead of user history logs.

6. DISCUSSION

Our introduction of latent variable "category class", "sentiment class" and "preference class" is indispensable for inferring writer's preference and can reduce the total calculation cost. The biggest advantage of incorporating these latent classes is in merging similar distributions into one distribution. Previous topic models [15], [19] consider topics at the document level and the writer level separately when inferring document topics and writer's preference. Since many similar reviews and writers with similar preferences exist in the corpus, this learning is inefficient and verbose. Instead of learning topics and preferences independently, we incorporate novel latent classes that are indicator variables by merging variables having similar distribution at each layer(review and writer) and can be shared among all reviews and writers to learn across the groups of writer's preference using distributions over topics.

Second, we discuss the calculation overhead of LET. In fact, LET consumes more time the other models in each iteration of Gibbs sampling. Nevertheless, adding the latent variables results in reducing the number of parameters by merging parameters in the same way that topic models represent co-occurrence words as the same topic variable; this reduces the number of possible parameters without losing generality and lowers the number of iterations that Gibbs sampling requires for estimation. Therefore, LET gains lower perplexity with fewer iterations than the others, see Table 4. Moreover, LET doesn't need to calculate the similarity after learning, because this model groups, simultaneously, writers and documents by assigning those classes. Consequently, the overall calculation cost of LET is not impractical for Web services based on user-generated content. LET suits the prediction of user reviews of a given user, while sLET suits the prediction of both item and corresponding rating of a given user like previous CF.

7. CONCLUSION

In this paper, we proposed a hierarchical topic model that simultaneously captures topics, sentiment at the document level, and preference at the writer level. A novel feature of our model is that it explores the generative process of review articles by sampling words from a mixture model involving a background topic, sentiment specific topics, and category specific topics. Experiments on various data sets showed that LET can determine the attitudes of writers, that lead to predict their reviews on these attitudes. In future work, we will extend the LET model by incorporating other metadata such as time [8], trends [9], social network membership, and affiliation.

8. REFERENCES

[1] J. K. B. Sarwa, G. Karypis and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[2] J. S. Breese, D. Heckerman, and C. M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *The 14th Annual Conference on Uncertainty in Artificial Intelligence*, pages 43–52, July 1998.

[3] http://movies.msn.com.

[4] http://mtg.upf.edu/node/1671.

[5] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD*, pages 168–177, 2004.

[6] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[7] N. Kawamae. Latent interest-topic model:finding the conditional relationships between author-document. In *CIKM*, pages 649–658, 2010.

[8] N. Kawamae. Serendipitous recommendations via innovators. In *SIGIR*, pages 218–225, 2010.

[9] N. Kawamae. Trend analysis model: Trend consists of temporal words, topics, and timestamps. In *WSDM*, page To appear, 2011.

[10] N. Kawamae and K. Takahashi. Information retrieval based on collaborative filtering with latent interest semantic map. In *ACM KDD*, pages 618–623, 2005.

[11] J. Konstan, B. Miller, J. H. D. Maltz, L. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[12] B. Marlin. Modeling user rating profiles for collaborative filtering. In *NIPS*, 2003.

[13] A. McCallum, A. Corrada-Emanuel, and X. Wang. Topic and role discovery in social networks. In *IJCAI*, pages 786–791, 2005.

[14] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *WWW*, pages 171–180, 2007.

[15] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *KDD*, pages 500–509, 2007.

[16] D. Pavlov and D. Pennock. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. In *NIPS*, pages 1441–1448, 2002.

[17] A. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *EMNLP*, 2005.

[18] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. An open architecture for collaborative filtering of netnews. In *CSCW*, pages 175–186. ACM Press, 1994.

[19] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. L. Griffiths. Probabilistic author-topic models for information discovery. In *KDD*, pages 306–315, 2004.

[20] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *WWW*, pages 111–120, 2008.

[21] P. Wei and S. Xiaotong. Penalized model-based clustering with application to variable selection. *J. Mach. Learn. Res*, 8:1145–1164, 2007.

[22] L. Zhuang, F. Jing, and X. Zhu. Movie review mining and summarization. In *CIKM*, pages 43–50, 2006.

APPENDIX

As shown in eq (1), multinomials ν, ψ_g, θ_e, χ_e, ω_c, φ_c, φ_s, μ_d and ϕ_z can be adapted by the conjugate prior and then integrated out analytically as follows:

$$
\begin{aligned}
&P(\mathbf{w},\mathbf{z},\mathbf{r},\mathbf{m},\mathbf{c},\mathbf{v},\mathbf{s},\mathbf{e},\mathbf{g}|\alpha,\beta,\zeta,\gamma,\delta,\epsilon,\eta,\lambda,\iota)=\int p(\mathbf{w},\phi|\mathbf{z},\delta)d\phi\\
&\times\int p(\mathbf{r},\mu|\eta)d\mu\int p(\mathbf{z},\varphi|\mathbf{r},\mathbf{c},\mathbf{s},\gamma)d\varphi\int p(\mathbf{m},\omega|\mathbf{c},\varepsilon)d\omega\int p(\mathbf{g},\nu|\iota)d\nu\\
&\times\int p(\mathbf{s},\chi|\mathbf{e},\zeta)d\chi p(\mathbf{v}|\lambda,\mathbf{s})\int p(\mathbf{c},\theta|\mathbf{e},\beta)d\theta\int p(\mathbf{e},\psi|\mathbf{g},\alpha)d\psi\\
&=[\frac{\Gamma(\sum_g^G\iota_g)}{\prod_g^G\Gamma(\iota_g)}][\frac{\Gamma(\sum_e^E\alpha_e)}{\prod_e^E\Gamma(\alpha_e)}]^G[\frac{\Gamma(\sum_c^C\beta_c)}{\prod_c^C\Gamma(\beta_c)}]^E[\frac{\Gamma(\sum_s^S\zeta_s)}{\prod_s^S\Gamma(\zeta_s)}]^E\\
&\times[\frac{\Gamma(\sum_m^M\epsilon_m)}{\prod_m^M\Gamma(\epsilon_m)}]^C[\frac{\Gamma(\sum_z^z\gamma_z)}{\prod_z^Z\Gamma(\gamma_z)}]^{C+S}[\frac{\Gamma(\sum_r^R(\eta_r))}{\prod_r^R\Gamma(\eta_r)}]^D[\frac{\Gamma(\sum_w^W\delta_w)}{\prod_w^W\Gamma(\delta_w)}]^{Z+1}\\
&\times\frac{\prod_g^G\Gamma(n_g+\iota_g)}{\Gamma(\sum_g^G(n_g+\iota_g))}\prod_g^G[\frac{\prod_e^E\Gamma(n_{ge}+\alpha_e)}{\Gamma(\sum_e^E(n_{ge}+\alpha_e))}]\\
&\times\prod_e^E\frac{\prod_c^C\Gamma(n_{ec}+\beta_c)}{\Gamma(\sum_c^C(n_{ec}+\beta_c))}\frac{\prod_s^S\Gamma(n_{es}+\zeta_s)}{\Gamma(\sum_s^S(n_{es}+\zeta_s))}\\
&\times\prod_c^C\frac{\prod_m^M\Gamma(n_{cm}+\epsilon_m)}{\Gamma(\sum_m^M(n_{cm}+\epsilon_m))}\prod_d^D p(v_d|\lambda_{s_d})[\frac{\prod_r^R\Gamma(n_{dr}+\eta_r)}{\Gamma(\sum_r^R(n_{dr}+\eta_r))}]\\
&\times\prod_c^C[\frac{\prod_z^Z\Gamma(n_{cz}+\gamma_z)}{\Gamma(\sum_z^Z(n_{cz}+\gamma_z))}]\prod_s^S[\frac{\prod_z^Z\Gamma(n_{sz}+\gamma_z)}{\Gamma(\sum_z^Z(n_{sz}+\gamma_z))}]\\
&\times\prod_z^{Z+1}\frac{\prod_w^W\Gamma(n_{zw}+\delta_w)}{\Gamma(\sum_w^W(n_{zw}+\delta_w))},
\end{aligned}
\tag{10}
$$

where n_g represents the number of writers assigned to preference class, g, and $n_{ge}(n_{ec}$, n_{cm}, n_{cz}, n_{dr}, $n_{zw})$ represent the number of documents associated with writers placed in g that is assigned to e, the number of categories, c, in documents associated with e, the number of items, m, in documents associated with c, the number of topics, z, in documents associated with c, the number of switch variables, r, in document d and the number of words w in tokens associated with z, respectively.