Jointly Modeling Aspects, Ratings and Sentiments for Movie Recommendation (JMARS)

Qiming Diao
Singapore Mgt. University
qiming.ustc@gmail.com

Minghui Qiu*
Singapore Mgt. University
minghuiqiu@gmail.com

Chao-Yuan Wu*
Carnegie Mellon University
cywu@cmu.edu

Alexander J. Smola
CMU and Google
alex@smola.org

Jing Jiang
Singapore Mgt. University
jingjiang@smu.edu.sg

Chong Wang
Carnegie Mellon University
chongw@cs.cmu.edu

ABSTRACT

Recommendation and review sites offer a wealth of information beyond ratings. For instance, on IMDb users leave reviews, commenting on different aspects of a movie (e.g. actors, plot, visual effects), and expressing their sentiments (positive or negative) on these aspects in their reviews. This suggests that uncovering aspects and sentiments will allow us to gain a better understanding of users, movies, and the process involved in generating ratings.

The ability to answer questions such as *"Does this user care more about the plot or about the special effects?"* or *"What is the quality of the movie in terms of acting?"* helps us to understand *why* certain ratings are generated. This can be used to provide more meaningful recommendations.

In this work we propose a probabilistic model based on collaborative filtering and topic modeling. It allows us to capture the interest distribution of users and the content distribution for movies; it provides a link between interest and relevance on a per-aspect basis and it allows us to differentiate between positive and negative sentiments on a per-aspect basis. Unlike prior work our approach is entirely unsupervised and does not require knowledge of the aspect specific ratings or genres for inference.

We evaluate our model on a live copy crawled from IMDb. Our model offers superior performance by joint modeling. Moreover, we are able to address the *cold start* problem — by utilizing the information inherent in reviews our model demonstrates improvement for new users and movies.

Keywords

Collaborative Filtering; Topic Models; Integrated Modeling; Sentiment Analysis

*Equal contribution

KDD'14, August 24–27, 2014, New York, NY, USA.

http://dx.doi.org/10.1145/2623330.2623758.

1. INTRODUCTION

Collaborative filtering is a staple to many business in the internet economy. Data to build good content recommender systems essentially comes in three guises: interactions, ratings, and reviews. First and foremost there is information whether a recommended item was consumed (i.e. viewed, clicked-on, purchased). This is the key source of information in search, ranking and advertising systems [4]. A common approach to processing this data is to try to estimate the probability that a user will interact with a given item, using past interactions as training data. Second, there is rating information regarding whether the user enjoyed the recommended item. This is the traditional domain of collaborative filtering. Its use was popularized through the Netflix contest [3] and it aims to reconstruct a choice set of matrix entries [17] or the entire matrix altogether [5]. Third, there are *reviews*, as provided by the users. This is arguably the most valuable user generated content since in it users not only *rate* items but they also *explain why* they liked or disliked an item. Hence, a system capable of extracting this information automatically should be able to generate more relevant information, and, as a side effect, also allow us to obtain meaningful profiles of the users and objects involved [12]. Our system belongs to the family of *integrated* models that use ratings and reviews to extract a wealth of information. We provide a statistical model and we demonstrate in experiments that our approach excels at recommending movies while simultaneously providing meaningful analysis of the interests and aspects relevant for users and movies.

We begin with an example of the type of analysis we are able to obtain for reviews. In it, positive sentiments are annotated as green, negative ones as red, and blue terms are movie-specific. Below we omit information regarding the *specific* aspect for visualization purposes (see Table 6).

I enjoyed this DVD from the library very much. Daniel Craig plays a believable James Bond. There are some of the older 007 action scenes and similar gimmicks with updates thanks to the younger Quartermaster. Eve plays well with grit and feminism including a surprise revelation at the end. It's touching as well with the final scenes in the mansion and the old Caretaker. Adele's award for best song is well deserved. But the plot was pretty weak and the film dragged on and on and on, probably being 30 minutes too long. The filming is it's usual high quality, but still overall both my wife and I found this boring, something you can't usually level against a Bond film.

1.1 Integrated Modeling

The motivation for our work arises from the task of serving the *right* items to users. This involves a number of challenges ranging from designing an effective user interface to user personalization to solving the cold-start challenge of initializing a recommendation system with meaningful content. Given the wealth of information inherent in review sites, such as IMDb, Netflix and Amazon Prime, it is tempting to extract more than just ratings from the data. After all, we want to understand *why* a user liked a particular movie, what his *preferences* are when it comes to selecting a movie (visual effects, plot, choice of subject matter).

Conventionally, in factorization approaches to recommendation [9] one uses exclusively the information inherent in the ratings. Consequently the latent factors have a certain degree of ambiguity — for instance, if we capture user and movie attributes with vectors v_u and v_m to predict a score $r_{um} \sim \langle v_u, v_m \rangle$, then the parameters are invariant under rotation. That is, replacing v_u with Uv_u and v_m with Uv_m for some rotation U will leave the outcome unchanged, yet it may considerably alter the interpretation of coordinate-wise attributes in v_u and v_m. This is undesirable in several respects: It leads to hard-to-understand factors; The factors may change considerably while leaving the underlying statistical model unchanged.

However, these factors represented as a vector of numbers are usually hard to interpret. For instance, did the movie have good acting but bad story, did the user prefer the director but dislike the genre? At the same time many review sites have textual content *in addition* to the numerical scores. For instance, IMDb is primarily a review site, Netflix allows for comments, YouTube is comments-only, Yelp contains comments and reviews but is lacking in terms of recommendation, and Zagat primarily focuses on curated content. This allows us to solve the above problem. For example, the word "predictable" in a movie review tells us that the user is talking about the "plot" aspect with a "negative sentiment"; likewise, a word "hilarious" tell us that the movie is a comedy and that the user probably likes it.

It is therefore tempting to try extracting additional meaning from textual data. This is valuable, e.g. when building a search and retrieval system since it allows us to identify attractive (and undesirable) aspects. For example, if a user always likes to write reviews talking about the special effects, we should recommend movies with great special effects to her if we can identify these movies. Moreover, we can learn from aspect related specialization which terms are associated with aspect specific sentiments.

1.2 An Overview of the Model

Our model provides a principled extension of the factorization models commonly used for recommendation. That is, we retain the notion that reviews are generated by incorporating user and movie specific features. However, unlike simple vectorial rating models, we use a structured representation to capture the interaction between movie and user. In this sense our model borrows from the tensorial factorization approach of [16] and it extends it from scalars to documents.

More specifically, we assume that each user (and each movie) has an aspect distribution of interest. Reviews are generated by drawing from the product of movie and user aspects. For instance, a review text will likely contain details about special effects, but only if the user is actually interested in them *and* if the movie has special effects worth discussing. Hence, reviews inform both about the content of a movie and also about the interests of a user.

This differentiation allows us to attribute partial scores to interests, i.e. we assume that the review scores arise from the process of combining partial scores associated with different aspects of the movie. Not only does this improve rating accuracy, it also allows us to attach *sentiments* to aspects. In other words, we can model which terms associate with positive, negative, and neutral aspect specific words within an aspect. We model the following five groups of words:

Background That is, words uniformly distributed in every review are considered background words. For example, in the case of movie reviews, these words include "characters", "movies", etc.

Movie-specific Words such as the name of the characters in a movie, or any term that appears only in the movie are considered movie-specific. These two types of words provide less information about movie quality.

Aspect These are words associated with specific aspects. For example, "music", "sound", and "singing" are all aspect words related to the "music" aspect.

Aspect-Sentiment These words usually come with a specific aspect to express positive or negative sentiments. For example, words such as "bored", "predictable" usually appear with a discussion of the "plot".

General sentiment For example, words such as "great", "bad", or "worse" do not really convey any aspect specific content. We call them general sentiment words.

1.3 Contributions

The key contribution of our model is that it integrates all available data sources, that is, it provides a joint model of user activity, movie content, ratings, reviews, and a detailed language model of the reviews. We show the following:

- Our model outperforms state-of-the-art recommender systems such as matrix factorization [15].
- We obtain an aspect representation of user interests and movie properties.
- We are able to uncover aspect-specific sentiment words.
- We provide an efficient inference algorithm.
- Our experiments are carried out on a real-world snapshot of reviews crawled from IMDb.

In summary, this is the first model tackling the problem set as a whole rather than piecemeal. We begin with an overview of related work in Section 2. This is followed by a description of the model in Section 3. Inference algorithms are provided in Section 4. We then present experimental results in Section 5 and a conclusion in Section 7.

2. RELATED WORK

Collaborative filtering is a fertile area of research and there exists a multitude of techniques which can readily be applied to *subsets* of the problem that we tackle. See e.g. [18, 9] for a review. Specifically, probabilistic matrix factorization methods [15, 17] have proven successful in real world problems [3, 8, 11, 25, 22].

However, probabilistic matrix factorization techniques struggle to generalize to new items, i.e. they fail at the cold-start problem. Regression based latent factor models (RLFM) [1] use attribute features to solve this problem by incorporating observable features into latent factors. Recent research [22, 16] incorporates Latent Dirichlet Allocation (LDA) and uses the topic as features, e.g. for recommending scientific articles. In terms of ratings, [19] use a statistically more appropriate model for capturing the discrete nature of the reviews by formulating an exponential families approach.

Moreover, there is rich literature analyzing reviews, e.g. using LDA [14, 26, 23] to uncover topics and sentiments. These works provide a more fine-grained analysis of review texts by separating sentiment words from neutral aspect words. In light of this, we build a language model component in our integrated model to capture aspects and sentiments in reviews. Different from a semi-supervised component or opinion lexicon used in [14, 26], our sentiments are learnt by building a linkage between user ratings and sentiments.

A recent line of work aims to model multi-aspect ratings from reviews, e.g. [20, 13, 10, 13, 12]. However, it often relies on having aspects readily available, often with aspect-specific ratings. The work of [20] uses LDA based model to identify 'topics' that are correlated with user ratings. Similarly, [13] uses multi-aspect ratings to infer sentiments for predefined aspects. With respect to the importance of mining ratable aspects that contribute to user ratings, as shown in these works, our model also seeks to profile a user's aspect preference when it comes to selecting a movie.

The key difference in our model is that it provides an *integrated* approach to this broad range of problems. Probably Hidden Factors as Topics (HFT) [12] is the closest to our work. HFT jointly models review texts and user ratings by associating each topic dimension with a hidden factor. However, unlike HFT we do *not* have such constraint. This increases the range of applicability. As shown in experiments, our model discovers a more meaningful low-rank representations of aspects, sentiments and movies, and a better recommendation results.

3. MODEL

3.1 Modeling Assumptions

Our task is to predict for a given user u and a movie m both the observed rating r_{um} and also the review w_{um}, as given by a collection of words w_{umi}. In contrast to previous work we model both aspects jointly, using a multitude of observed and latent variables.

The most intuitive way of understanding the model of Figure 1 is to consider how a review is written. Users are assumed to have a given interest distribution θ_u in terms of aspects they write and care about. Moreover, they are also assumed to have biases b_u regarding what can be considered to be a reasonable baseline with regard to their choice. Likewise, movies contain a number of aspects, as indicated by θ_m and a bias b_m.

Whether a user likes a particular movie depends on a number of things. First, it helps if the movie contains aspects the user cares about. Secondly, it is also important that the user's expectations v_u match the movie's properties v_m, when viewed under the angle of a specific aspect, as captured by M_a. These aspect-specific ratings of a movie by a user r_{uma} are then aggregated, based on the user's priorities to obtain an aggregate rating r_{um}.

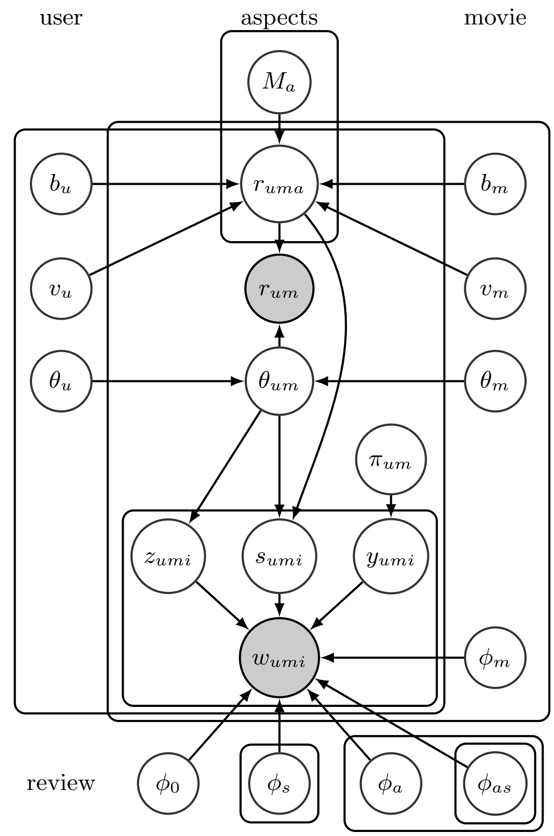

Figure 1: Factorized rating and review model. Note the symmetry between users and movies. The model contains four major plates: aspects, users, movies, and the words within a given review. They are nested and partially overlapping. For convenience we represent the aspect plate as two separate plates (language model and aspect review model are contained in the same plate).

As for the actual review text, we assume the following: reviews contain words drawn from a baseline language model of words typically occurring in reviews ϕ_0. Moreover, there are positive and negative sentiment words, as indexed by ϕ_s, where $s \in \{\text{positive}, \text{negative}\}$. Finally, we assume that there are aspect specific word distributions ϕ_a, again colored by sentiment s, i.e. ϕ_{as}. Depending on whether a user appreciates a particular aspect of a movie, as indicated by r_{uma}, he will generate positive or negative sentiment words (or simply neutral ones). Finally, there are also movie-specific words, such as the name of the main protagonists, the title, and other named entities that are bound to occur in a review, regardless of the user. This approach of mixing between five different components summarizes our strategy. Probably most closely related is the model of [2] who use a similar switch construction to distinguish between positive and negative sentiment. The key difference is that we do not have any explicit information regarding attitude and aspects. Instead, we need to extract this from the reviews.

We employ a conventional bag of words representation, paired with a Dirichlet-Multinomial to capture the word distribution of the reviews. Aspect-specific ratings are generated by matrix factorization, i.e. a Gaussian inner-product model. The twist here is that we capture aspect specific preferences via a scaling matrix M_a. This is a strict generalization of regular factorization approaches. Finally, the mixing between these aspects occurs by an exponential linear model which also governs review combination.

3.2 Matrix Factorization with Aspects

As is common in collaborative filtering, only a tiny fraction of matrix entries are present — our dataset contained less than 0.03% observed entries. To infer the missing entries collaborative filtering relies on the assumption that the underlying matrix has fairly low rank and thus, a small number of terms suffice to determine the remainder of the results.

One may argue that this is only part of a solution, since the relative values of the entries matter more than their absolute value [25, 24]. That said, for the purpose of comparison to existing results we adopt the strategy of measuring the least mean square deviation. The matching probabilistic model is that of additive noise relative to an estimated relevance score. We build on the probabilistic matrix factorization (PMF) approach of [17].

As in PMF, we assume that users u and movies m are characterized by latent factor vectors v_u and v_m respectively, that are drawn from zero-mean spherical Gaussian priors

$$v_u \sim \mathcal{N}(0, \sigma_u^2 \mathbf{I}) \text{ and } v_m \sim \mathcal{N}(0, \sigma_m^2 \mathbf{I}). \tag{1}$$

The hyperparameters σ_u^2 and σ_m^2 are user-related and movie-related variances, respectively. In a conventional recommender model one would then assume that

$$r_{um}^{\text{conventional}} \sim \mathcal{N}(v_u^\top v_m + b_u + b_m, \sigma^2)$$

That is, given biases b_u and b_m, we observe a noisy variant of the compatibility. Different from PMF [15], we assume an aspect-specific rating of movie m by user u.

$$r_{uma} = v_u^\top M_a v_m + b_u + b_m + b_0. \tag{2}$$

Here b_u and b_m are biases for users and movies respectively and b_0 is a common bias. The idea is that while v_u and v_m encode the general profile, the matrix M_a emphasizes the aspect specific properties. That is, while movies may be overall good, they may or may not excel quite as much in specific aspects.

We assume Gaussian priors with fixed mean and precision on real-valued parameters. Specifically, we assume that each element of M_a, v_u, v_m, b_u, b_m follows a Gaussian distribution with zero mean and a fixed precision.

3.3 Two Factor Model

One of the challenges in combining user and movie attributes is in the task to *fuse* the respective attributes into a joint model. Our approach borrows from [7] by designing an exponential additive model in terms of θ_u and θ_m. The latter are user and movie specific aspect parameters which jointly generate the aspect distribution of a review. Our assumptions are as follows:

$$\theta_u \sim \mathcal{N}(0, \sigma_{\text{useraspect}}^2 \mathbf{1}) \text{ and } \theta_m \sim \mathcal{N}(0, \sigma_{\text{movieaspect}}^2 \mathbf{1}) \tag{3}$$

Moreover, the joint aspect distribution is given by

$$\theta_{um} \propto \exp(\theta_u + \theta_m) \text{ i.e. } p(a|\theta_u, \theta_m) = \frac{e^{\theta_{ua} + \theta_{ma}}}{\sum_{a'} e^{\theta_{ua'} + \theta_{ma'}}} \tag{4}$$

We also make the (slightly controversial) modeling assumption that the extent of discussion in a review and the relative importance of a aspect coincide. That is, aspects that are discussed at twice the length will contribute twice as much to the aggregate score for a review. This yields

$$\hat{r}_{um} = \mathbf{E}_{a|\theta_u, \theta_m}\left[v_u^\top M_a v_m + b_u + b_m + b_0\right] \tag{5}$$

$$= v_u^\top \left[\sum_a p(a|\theta_u, \theta_m) M_a\right] v_m + b_0 + b_u + b_m \tag{6}$$

Here $\hat{r}_{um}$ is the predicted review rating, and the observed rating r_{um} is generated using $\mathcal{N}(\hat{r}_{um}, \epsilon^{-2})$. This is a strict generalization of the PMF model. The key difference is that the aspect weighting for a given (user,movie) combination is dependent on the aspects they excel in. In other words, the metric is variable in accordance with the content of the movie and the interest of the user. The idea is that, if a movie is a SciFi movie with correspondingly high value of $\theta_{m,\text{SciFi}}$, then the user's review of the movie will likely contain SciFi-related content and moreover, the SciFi quality of the movie will matter in terms of the overall rating. That is, $\theta_{um,\text{SciFi}}$ will likely be large.

A few comments regarding M_a are in order. First and foremost, it does not increase the total number of parameters dramatically, since we only require k terms for each diagonal matrix. In turn it allows us retain *one* joint latent attribute model in v_u and v_m while simultaneously being able to identify individual aspects as needed via $v_u^\top M_a v_m$.

3.4 A Language Model for Reviews

A key in our reasoning is the integration between ratings and reviews. We already established the link between general attributes, aspect-specific ratings and posited a model for the aspect distribution of the a review.

As shown in the sample review in the introduction, when writing a movie review, the user will express his opinions through a set of sentiment words, such as *best*, *weak* or *boring*. Close examination also shows that the user has different opinions on different aspects of the movie. For instance, the user might like the *music* of a movie but dislike the *plot*. This motivates us to model an aspect-specific sentiment for a movie. Overall, we assume that the review language model is given by a convex combination of five components.

- A background language model covering the default word distribution ϕ_0.
- A background sentiment distribution addressing positive and negative sentiments, i.e. ϕ_{s+} and ϕ_{s-}. They are not document specific.
- A movie-specific word distribution ϕ_m. This is employed to capture the names of actors, movie title, and primarily salient entities in the review.
- An aspect-specific word distribution ϕ_a.
- An aspect-specific sentiment distribution ϕ_{as+} and ϕ_{as-} capturing positive and negative sentiments. Note that the use of words can be highly context specific. For instance, while *brutal* tends to carry a negative connotation, it is associated with positive reviews in the context of war movies. We detect this automatically.

Crucial to the mixture between these models is the use of a switch variable which chooses between the above types. We accomplish this via π, the switching distribution. From it we draw the selector variable y_{umi} for each word and depending on its value we pick one of the above five components. We now go through each of the terms in detail:

Switching distribution π_{um}**:** We draw it from a Dirichlet prior. Subsequently draw y_{umi} from π_{um}, that is

$$\pi_{um} \sim \text{Dir}(\gamma) \text{ for } \pi_{um} \in \mathcal{P}_5 \quad (7)$$

$$y_{umi} \sim \text{Mult}(\pi_{um}) \quad (8)$$

In other words, we infer on a per-review basis what the mixture of generic and specific terms is.

Aspect z_{umi}**:** Whenever we draw an aspect-specific word, we need to decided the aspect. This is accomplished by sampling from θ_{um}, i.e.

$$z_{umi} \sim \text{Mult}(\theta_{um}). \quad (9)$$

Aspect sentiment s_{umi}**:** When s_{umi} is an aspect-specific sentiment, its sentiment is determined by the aspect-specific rating via a logistic link function.

$$p(s_{umi}|r_{uma}, z_{umi} = a) = \frac{1}{1 + e^{-s_{umi}(cr_{uma}-b)}}. \quad (10)$$

In other words, the propensity of picking a positive or a negative sentiment word are related to the aspect specific rating r_{uma}. Note that we identify $s_{umi} = 1$ with positive and -1 with negative sentiment.

Aggregate sentiment s_{umi}**:** When s_{umi} is an general sentiment, this is entirely analogous to above. The only difference is that we draw s_{umi} from the aggregate rating $\hat{r}_{um} = \sum_a \theta_{uma} r_{uma}$. In other words, as before, we use a logistic model to infer general sentiments, employing the predicted review rating $\hat{r}_{um}$.

Language models $\phi_0, \phi_s, \phi_a, \phi_{as}, \phi_m$**:** Each of the language models is a multinomial distribution with a Dirichlet as a conjugate. That is, we assume that

$$\phi_0 \sim \text{Dir}(\eta_0), \qquad \phi_s, \phi_{as} \sim \text{Dir}(\eta_{\text{sentiment}}),$$
$$\phi_a \sim \text{Dir}(\eta_{\text{aspect}}), \qquad \phi_m \sim \text{Dir}(\eta_{\text{movie}})$$

where the value of each element in $\eta_.$ depends the part-of-speech tag of the corresponding word. Adding hierarchy to language models is an obvious direction for improvement, albeit at the expense of a rather more expensive inference problem.

Emission model: The final piece in our approach is to model how the actual words are being generated.

- Based on y_{umi} decide which of the five model types to pick.
- If y_{umi} is aspect specific, select ϕ from the aspect models using aspect z_{umi}.
- If y_{umi} is aspect-sentiment specific, inspect s_{umi} for a matching sentiment for aspect z_{umi}.
- If y_{umi} is sentiment specific, inspect s_{umi} for the corresponding sentiment.

Likewise, we choose the baseline model ϕ_0 or the movie specific model ϕ_m as needed.

By default we choose Gaussian priors for real-valued parameters and Dirichlet conjugate priors for the multinomial distributions. This completes the model specification.

3.5 Properties

Before we delve into details of the inference algorithm, a brief discussion of some properties of the model is in order. The coupling between aspect specific sentiments and ratings allows us to infer such terms without the need for detailed reviews. In fact, it overcomes the problem arising in [19]: there the recommender model could not take advantage of aspect specific ratings to obtain a more refined user model. Moreover, it overcomes the limitation of having only a small number of aspects, such as in [12] since it does not require an explicit formulation of categories. To the best of our knowledge, this is the first integrated model for recommendation.

As byproduct we obtain aspect preferences for both movie and user. Furthermore, we are able to extract movie-specific terms via ϕ_m. This is useful for search and retrieval.

4. INFERENCE AND LEARNING

Our goal is to learn the hidden factor vectors, aspects, and sentiments of the textual content to accurately model user ratings and maximize the probability of generating the textual content. Hence our objective is the negative log posterior, defined as

$$\mathcal{L} := -\log p(\mathcal{R}, \mathcal{W}|\Upsilon, \Omega). \quad (11)$$

where $\mathcal{R}, \mathcal{W}$ denote the ratings and words respectively and Υ and Ω are the Gaussian and Dirichlet hyperparameters.

Unfortunately, inference in this problem is intractable in its direct formulation. Instead, we resort to a hybrid inference procedure combining sampling and variational optimization. That is, we use Gibbs-EM [21], an inference method that alternates between collapsed Gibbs sampling [6] and gradient descent, to estimate parameters in the model. After collapsing out the parameters pertaining to the language model, terms cease to be conditionally exchangeable, hence we cannot decompose $\mathcal{L}$ further. That said, all relevant terms decompose for the purpose of the inference algorithm and we have:

$$\mathcal{L} \text{ “=” } \sum_{r_{um} \in \mathcal{R}} \left[\epsilon^{-2}(r_{um} - \hat{r}_{um})^2 - \log p(w_{um}|\Upsilon, \Omega) \right]. \quad (12)$$

The first term denotes the prediction error on user ratings. The second term denotes the probability of observing the text conditioned on priors. Note that this is *not* a formal equality since each review and score depends on its annotation and, indirectly, on the annotations of all remaining documents. This is simply to convey the intuition of the inference approach that we will pursue.

In the E-step, we perform Gibbs sampling to learn the hidden variables by fixing the values of θ_{um} and $\{r_{uma}\}_{a=1}^{A}$. In the M-step, we perform gradient descent to learn hidden factor vectors by fixing the values of $\{y, z, s\}_{umi}$.

4.1 E-step

In the E-step, we perform Gibbs sampling to learn the hidden variables $\{y, z, s\}_{umi}$ by fixing the values of θ_{um} and all $\{r_{uma}\}_{a=1}^{A}$ updated in the gradient descent step. Dirichlet-Multinomial conjugacy allows Gibbs sampling to work by sampling on the individual tuple of $\{y, z, s\}_{umi}$, collapsing out all the language models ϕ. As this is a conventional step, we omit the detailed derivations and present the derived Gibbs sampling update rules. Interested readers are referred to [6] for more details.

For the word in the i-th position of the review written by user u for movie m, we jointly sample its switching variable y_{umi}, topic z_{umi} and sentiment s_{umi}, conditioned on its Markov blanket. Let $w = w_{umi}$ and d denote the set of variables $\{umi\}$.

$$
\begin{aligned}
&p(y_d = y, z_d = z, s_d = s | y_{\neg d}, w, \theta_{um}, \Omega) \qquad (13)\\
&\propto \frac{C^y_{\neg d} + \gamma}{\sum_{y'=1}^{5} C^{y'}_{\neg d} + 5\gamma} \cdot \left[\frac{C^w_{y,\neg d} + \eta^w_0}{\sum_{w'=1}^{V} C^{w'}_{y,\neg d} + \eta^{(\cdot)}_0}\right]^{\mathbb{I}(y=0)}\\
&\cdot \left[\frac{C^w_{y,\neg d,s} + \eta^w_{\text{sentiment}}}{\sum_{w'=1}^{V} C^{w'}_{y,\neg d,s} + \eta^{(\cdot)}_{\text{sentiment}}} p(s|\hat{r}_{um})\right]^{\mathbb{I}(y=1)}\\
&\cdot \left[\frac{C^w_{y,\neg d,z} + \eta^w_{\text{sentiment}}}{\sum_{w'=1}^{V} C^{w'}_{y,\neg d,z} + \eta^{(\cdot)}_{\text{sentiment}}} \cdot \theta_{umz} \cdot p(s|r_{umz})\right]^{\mathbb{I}(y=2)}\\
&\cdot \left[\frac{C^w_{y,\neg d,z} + \eta^w_{\text{aspect}}}{\sum_{w'=1}^{V} C^{w'}_{y,\neg d,z} + \eta^{(\cdot)}_{\text{aspect}}} \cdot \theta_{umz}\right]^{\mathbb{I}(y=3)}\\
&\cdot \left[\frac{C^w_{y,\neg d,m} + \eta^w_{\text{movie}}}{\sum_{w'=1}^{V} C^{w'}_{y,\neg d,m} + \eta^{(\cdot)}_{\text{movie}}}\right]^{\mathbb{I}(y=4)}
\end{aligned}
$$

Here $C^w_{y=4,\neg d,m}$ denotes the number of times that w is sampled as a movie-specific word in movie m excluding the current word assignment; all the other Cs are defined in the same way. $\mathbb{I}(\cdot)$ is a indicator function that returns 1 if the statement is true and 0 otherwise. In other words, we effectively have a big switch statement distinguishing 5 cases.

Note that when $y = 3$, the word is an aspect word, and we need to sample an aspect label from θ_{um}, which is a deterministic softmax transformation of the sum of θ_u and θ_m given by (4). The aspect sentiment probability $p(s|r_{umz})$ is based on (10) and the aggregate sentiment $p(s_d = s|\hat{r}_{um})$ uses an analogous logistic function for the predicted general rating $\hat{r}_{um}$ of the movie.

4.2 M-Step

In this step, we use gradient descent to learn the set of parameters $\Theta = [\{v_u, b_u, \theta_u\}_{u=1}^{U}, \{v_m, b_m, \theta_m\}_{m=1}^{M}, \{M_a\}_{a=1}^{A}]$ by fixing the values of $\{y, z, s\}_{umi}$. In this case, our objective function is further modified as follows:

$$
\begin{aligned}
\mathcal{L}' = &\sum_{r_{um}\in\mathcal{R}} \left[\epsilon^{-2}(r_{um} - \hat{r}_{um})^2 - \log p(\{w, y, z, s\}_{um}|\Theta)\right]\\
&- \log p(\Theta|\Upsilon). \qquad (14)
\end{aligned}
$$

The first term remains unchanged from (12). The second goal is to maximize the likelihood of generating all the observed $\{y, z, s, w\}_{u,m}$ variables obtained from Gibbs sampling. The final term is the Gaussian prior of all the parameters. We then seek to minimize the following objective function, decomposed from (14).

Let $\mathcal{L}'_{um}$ be the objective for a single rating and review texts, i.e.: $\mathcal{L}' = \sum_{r_{um}\in\mathcal{R}} \mathcal{L}'_{um} - \log p(\Theta|\Upsilon)$. We expand the likelihood contribution of a given (user,movie) pair $\mathcal{L}'_{um}$ as follows:

$$
\begin{aligned}
\mathcal{L}'_{um} =&\epsilon^{-2}(r_{u,m} - \hat{r}_{u,m})^2\\
&- \log p(\{w, z, s\}_{um} \mid \theta_u, v_u, b_u, \theta_m, v_m, b_m, M_a)\\
=&\epsilon^{-2}(r_{u,m} - \hat{r}_{u,m})^2 - \sum_s N^{y=1}_{u,m,s} \log p(s \mid \hat{r}_{um})\\
&- \sum_a \sum_s N^{y=2}_{u,m,a,s} \log p(s|r_{uma}) - \sum_a N^{y=3}_{u,m,a} \log \theta_{uma}.
\end{aligned}
$$

where $N^{y=1}_{u,m,s}$ is the number of times general sentiment s appears in user u's review in movie m, and $N^{y=2}_{u,m,a,s}$ is the number of times the aspect sentiment s appears under aspect a, and $N^{y=3}_{u,m,a}$ is the number of times aspect a appears in the review. We then compute the first derivatives of $\mathcal{L}'$ with respect to the variables. We optimize $\mathcal{L}'$ using L-BFGS.

4.3 Implementation

We perform 500 runs of Gibbs EM. In each run, we run one iteration for the Gibbs sampling stage and another 10 iterations of gradient descent. We fixed the number of topics and the dimension of the latent factors. For our models and competing baseline models, we use grid search on a development set to select the model hyperparameters. For grid search, we choose latent factor size from $\{5, 10\}$. As our data is sparse, a fairly low rank of factor vectors is sufficient; we also choose a relatively small aspect size from $\{5, 10, 20\}$, so as to leave space for the model to learn a much larger number of movie words. In the following experiments, the regression parameter ϵ^{-2} is set to be 5.0. Aspect distributions θ_u, θ_m have Gaussian priors, with variances being 0.1 and 1.0 respectively. To reflect the fact that more sentiment words should be adjectives, adverbs, or verbs, $\eta_0, \eta_{\text{movie}}$, and η_{aspect} is 0.001 on adjectives, adverbs, and verbs, and 0.01 for other words. On the other hand, $\eta_{\text{sentiment}}$ is 0.01 on adjectives, adverbs, and verbs, and 0.001 for other words.

5. QUANTITATIVE EVALUATION

Having defined our model mathematically we now proceed to evaluating it. We begin with a quantitative evaluation in the present section. A qualitative discussion of the results follows. Our experiments show that:

- Our model outperforms state-of-the-art methods in terms of MSE on recommendation.
- Our model has better predictive power in terms of perplexity on new reviews.
- Our model is able to model review texts effectively and distinguish between words associated with aspects and sentiments.

5.1 Protocol

We use a dataset compiled from IMDb. We randomly select 50k movies and crawl all their reviews. We only keep those reviews with user ratings (scaled from 0 to 10). We remove users who have less than two reviews and then remove movies with less than two reviews. Note that despite this simple cleaning, our data is much more sparse with only 0.03% entries present, than, say Netflix [3] or the datasets studied in HFT [12]. Table 1 displays some statistics.

# users	54,671
# movies	22,380
# user reviews	348,415 (density 0.03%)
# unigram (after pruning)	118,616

Table 1: IMDb data set. Unigrams containing stop words or punctuations, as well as infrequent unigrams that appear less than five times in the corpus are removed during pruning.

We present histograms over different numbers of reviews for movies and user in Figure 2. Clearly, the majority of users only write a small number of reviews and the majority of movies only receive a few reviews. This is not too surprising, given that IMDb aims to catalogue all movies, including obscure works dating from the 19th century. This

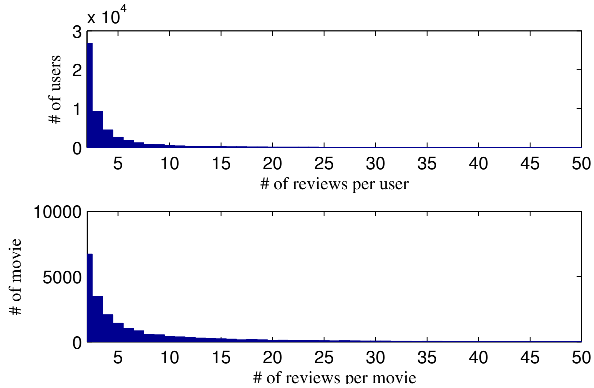

Figure 2: Histograms for reviews for movie and user.

sparsity underscores the importance of a method that can handle 'cold-start' for users or movies with few reviews.

We randomly split our data set into training, validation and test sets. Similar to [12], we use 80% of our dataset as training data, 10% for validation, and 10% for testing. We evaluate the following competing models for comparison: offset only, two state-of-the-art methods, and our model.

Offset only Predict the rating as the average of past ratings. This is the best constant predictor we can get.

PMF Probabilistic matrix factorization [15]. This model is designed for numerical ratings while ignoring all the review texts. By comparing to it, we evaluate the importance of jointly modeling ratings and reviews.

HFT Hidden factors with topics [12]. This work also models both review texts and ratings. It shows state-of-the-art performance on a variety of review data sets. By comparing with HFT, we examine which of them provides a better modeling of movie reviews.

JMARS Jointly modeling aspects, ratings and sentiments. This is the full model discussed in Section 3.

5.2 Perplexity

We analyze the perplexity of all the competing models. Perplexity is a standard measure to evaluate the quality of probabilistic models. The performance in terms of perplexity shows the prediction power of the model on unseen reviews, where a lower perplexity means a better performance.

Since PMF does not use review texts, it is not considered in this evaluation. For HFT and our model, we define perplexity as follows:

$$\log \mathrm{PPX}(D_{\mathrm{test}}) = -\frac{1}{N_w} \sum_{\substack{u,m \\ d_{um} \in D_{test}}} \sum_i \log p(w_{umi}). \quad (15)$$

Here $p(w_{umi})$ denotes the likelihood of generating the i-th word in the review written by user u for movie m in D_{test}, and N_w is the total number of words in the test data. In the following formulas, we use w and y to refer to w_{umi} and y_{umi} whenever indices are obvious.

In HFT, the word likelihood $p(w_{umi})$ is defined as:

$$p(w) = \sum_a \hat{\phi}_{a,w} \hat{\theta}_{m,a}. \quad (16)$$

where $\hat{\phi}_{a,w}$ is the estimated word distribution of topic a, and $\hat{\theta}_{m,a}$ is the estimated topic distribution of the movie m.

In our model, $p(w_{umi})$ is defined as:

$$\begin{aligned} p(w) = {} & p(y=0 \mid \pi_{um})\hat{\phi}_0^w + p(y=1 \mid \pi_{um}) \sum_s p(s \mid \hat{r}_{um})\hat{\phi}_s^w \\ & + p(y=2 \mid \pi_{um}) \sum_a \sum_s p(s \mid r_{uma})\hat{\theta}_{uma}\hat{\phi}_{as}^w \\ & + p(y=3 \mid \pi_{um}) \sum_a \hat{\theta}_{uma}\hat{\phi}_a^w + p(y=4 \mid \pi_{um})\hat{\phi}_m^w. \end{aligned} \quad (17)$$

Here we use the word distributions ϕ, user parameters $\{v_u, b_u, \theta_u\}$, movie parameters $\{v_m, b_m, \theta_m\}$ and M_a learned in the training step. In this case, we can calculate all terms in Eqn. 17 except π_{um}. Then we run Gibbs sampling on the testing data for 50 iterations to estimate π_{um}.

We vary aspect size and latent factor size to test model performance. Note that HFT enforces topics and latent factors with the same dimension, but our model allows them to have different dimensions. To evaluate the sensitivity of model performance in terms of aspect size, we vary aspect size for each latent factor size. Results are shown in Table 2.

Factor size	HFT	JMARS		
		A=20	A=10	A=5
5	8,247	**3,348**	3,369	3,399
10	7,459	**3,335**	3,359	3,379

Table 2: Comparison of models in terms of perplexity on held-out data in terms of different topic and latent factor size.

Consistently, our model achieves better performance than HFT in terms of different factor size. The main difference between our model and HFT lies in the way of modeling review texts, where our model uncovers underlying rich information, e.g.: aspect, sentiment and movie-specific contents. This shows a carefully designed language model for review texts could have better predictive power for unseen data. Furthermore, our model's performance varies more in terms of different aspect size A instead of factor size K. This shows that the latent factor dimension in probabilistic matrix factorization has minor effect, compared to the aspect dimension in topic modeling.

5.3 Movie recommendation

Factor size	Offset	PMF	HFT	JMARS		
				A=20	A=10	A=5
5	7.07	5.99	5.21‡	**4.97**†	5.11	5.23
10		5.92	5.14‡	**5.05**†	5.18	5.28

Table 3: Comparison of models in terms of MSE on held-out data. † and ‡ mean the result is better than the method in the previous columns at 1% and 0.1% significance level, measured by McNemar's test.

We compare our model with baseline models on the movie recommendation task, measured by Mean-Square-Error (MSE) on the held-out test data. Results are shown in Table 3. Similarly we vary topic size and latent factor size to test model performance. Our observations as follows:

- The offset baseline does not perform well compared to all other methods, which shows that our rating data has a relatively large variance.
- HFT significantly outperforms PMF at 0.1% significance level. Hence adding review texts can significantly improve the matrix factorization model.
- Our model achieves the best performance in terms of different factor size when the size of aspect is 20.
- Different from HFT where each topic is associated with a hidden factor dimension in matrix factorization, our model learns aspect-specific ratings and use aspect preference of reviews to aggregate these ratings to account for the final rating. This allows us to diverge aspect size from hidden factor size. For example, in our data set, the ratings are sparse (density 0.03%) but with rich user generated textual contents. Therefore, with small factor size and relatively large aspect size, our model can better fit the data and achieve better results in terms of MSE.
- Our results also suggest that a lower size of aspects may not be sufficient to capture distinct aspects and aspect sentiments in our data, which is an important premise for modeling aspect-specific ratings in our model. A relatively large aspect size has better performance and clean aspect words. We will present detailed aspect words in Section 6. We have also tried a larger size of aspect, but the improvement is minor.

5.4 'Cold-start' recommendation

Making recommendations for new users or items which do not have enough rating data is a common issue in recommendation systems. For our model and HFT, although the training data for an item is scarce, the review associated with it can still provide important textual information. HFT clusters the review words into topics, which are tied with item factor vector. Our model identifies aspect distribution and aspect sentiment within the review, and associates the sentiment words with matrix factorization. Therefore, both models can potentially help to better deal with 'cold-start' users and items.

We compare the performance of our model with HFT in terms of relatively improvement over PMF. Performance is evaluated on movies/users with different amount of reviews in training data, as shown in Figure 3. Our findings are as follows:

- In the comparison of different numbers of training ratings for movies, both our model and the HFT consistently outperform PMF, ranging from 10% to 34% relative improvement over PMF. This shows the benefit of modeling review texts for recommendation. Compared with HFT, our model's performance is similar when the number of reviews for movie is small, which suggests that it is difficult to learn the user's aspect taste and movie's aspect property given a few reviews. However, our model outperforms HFT when the number of reviews for movie is relatively large. It suggests that our model can better utilize the textual information (e.g. aspects, sentiments, aspect-sentiments) within user reviews, while the HFT only cluster review contents as topics.
- Both our model and HFT consistently outperform PMF under different numbers of training ratings for users. Similarly, we also observe that our model outperforms HFT when the numbers of training ratings for users is relatively large, which suggest that our model can better fit the textual information.

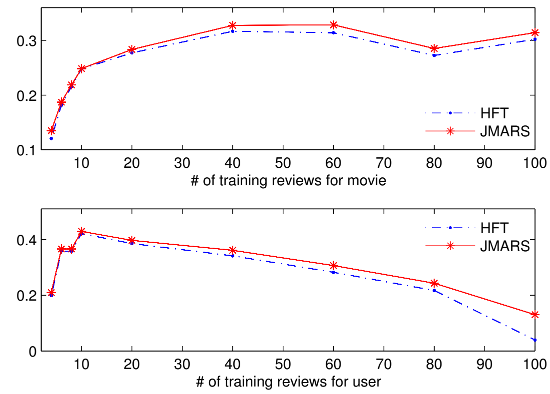

Figure 3: Improvement in MSE compared to PMF for 'cold-start' movies and users.

6. QUALITATIVE EVALUATION

6.1 Aspect rating

Aspect	Director	History	War	Life	Character
Rating	9.36	8.55	8.51	9.20	9.50
Prob	0.12	0.10	0.09	0.09	0.08

... what an excellent piece of cinema ... the actors are great and directing incredible ... in 300, Gerard Butler dominates the screen ... battle scenes are incredible ...

Table 4: The learnt aspect-specific ratings and latent sentiment identified by our model for a review.

To evaluate whether our model is capable of interpreting the reviews correctly, we examine the learned aspect ratings of our model. We present one review in our training set along with the learned aspect ratings and sentiments of the top 5 aspects in the table above. As we can see, the high aspect probability in 'director' aspect reflects the fact that positive sentiment has been expressed towards the director, e.g.: "directing incredible" in the review. Commonly one would assume that the *War* topic would dominate in anything written about the movie *300*, whereas here we are able to infer that it is the directing that is being reviewed.

6.2 Background and sentiment words

Background-word and sentiment-word distributions, ϕ_0 and ϕ_S, are presented in Table 5.

Not surprisingly, the top three background words are 'film', 'story', and 'character', all of which provide little information about aspects or sentiments. Positive sentiment words such as 'great' and 'good', and negative sentiment words such as 'bad' and 'boring', are all sentiment words which are not aspect-specific. Note that we do not handle negation, hence "not good" will be split into "not" and "good", which makes "good" appear in negative word distribution.

Type	Words
Background	film, story, character, films, characters, movies, scenes, scene, real, good
Positive	great, good, love, acting, fun, funny, excellent, lot, thought, perfect
Negative	bad, good, pretty, acting, plot, boring, worst, watching, minutes, stupid

Table 5: Top background words from ϕ_0 and sentiment words from ϕ_s.

6.3 Aspect and sentiment

Aspect words and aspect-sentiment words from three popular aspects are shown in Table 6. These words are easily interpretable. For example, for the aspect 'Adventure', the top words are "earth," "human" and "space". Aspect-sentiments contain sentiment words specific to aspects, e.g. "spectacular" of "Adventure" aspect, "sharp" of "Social" aspect, and "nasty" of "Violence" aspect. These words emphasize the importance of discriminating sentiment words for different aspects. Note that the word 'nasty' is classified as both positive and negative in the context of 'Violence'. In our opinion, this is not a mistake, as the word 'nasty' can indeed convey positive or negative connotations for different users at the same time.

6.4 Movie specific words

We present movie-specific words in Table 7. These are words that do not convey sentiment or genre information and are particular to the movie. They typically correspond to names of places, actors, and other entities. For example, character names like "Bond" and "James" pertaining to the movie "Casino Royale" and words like "Neo" to the movie "The Matrix Reloaded" . These words also provide a list of interpretable keywords specific to the movies.

Movie	Words
Casino Royale	bond, craig, james, casino, royale
Batman Begins	batman, bruce, wayne, bale, begins
The Matrix Reloaded	matrix, neo, reloaded, action, flight
American Beauty	beauty, american, spacey, lester, kevin

Table 7: Top movie-specific words from ϕ_m.

In summary, our model performs well at distinguishing different types of words: background, aspect, sentiment, aspect-sentiment and movie specific words. The resulting word distributions provide a low-rank representations of aspects, sentiments and movies, which give a great insight to understand them.

6.5 Failure Modes

After examining the cases which have higher prediction error rates, we find that one source of errors is the inconsistency of ratings and review words in reviews.

> Score: 1/10
>
> *I am a teenager, and I never thought of finding The Godfather so interesting! It shows a vivid and perfect example of the words Classic and Timeless in a movie...*

The reviewer expresses clearly positive opinions in the review yet gives a low rating. This is an observation that most systems would like to rule out since it may harm the whole system. One possible solution is to perform database cleaning by examining the inconsistency between sentiment words and ratings and rule out such cases. Our system can detect this case by observing the inconsistency between word probability and rating accuracy. This technique can then be applied to anomaly detection or database cleaning, which removes reviews with less meaningful information.

7. CONCLUSION

In this paper we proposed JMARS which provides superior recommendations by exploiting all the available data sources. Towards this end, we involve information from review and ratings. In fact our model is able to capture the sentiment in each aspect of a review, and predict partial scores under different aspects. Additionally the user interests and movie topics can also be inferred with the integrated model. We showed that our model outperforms state-of-the-art systems in terms of prediction accuracy and the language model for reviews is accurate. Future work includes capturing the hierarchical nature of movie topics and incorporating non-parametric models to increase flexibility. Moreover, a fast inference algorithm is required to further increase the scalability of this model.

8. ACKNOWLEDGEMENTS

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

9. REFERENCES

[1] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In J. Elder, F. Fogelman-Soulié, P. Flach, and M. Zaki, editors, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 19–28. ACM, 2009.

[2] A. Ahmed and E. P. Xing. Staying informed: supervised and semi-supervised multi-view topical analysis of ideological perspective. In *Conference on Empirical Methods in Natural Language Processing*, pages 1140–1150. ACL, 2010.

[3] R. M. Bell and Y. Koren. Lessons from the Netflix prize challenge. *SIGKDD Explorations*, 9(2):75–79, 2007.

[4] A. Z. Broder. Computational advertising and recommender systems. In P. Pu, D. G. Bridge, B. Mobasher, and F. Ricci, editors, *Conference on Recommender Systems*, pages 1–2. ACM, 2008.

[5] J.-F. Cai, E. J. Candés, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[6] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004.

[7] L. Hong, A. Ahmed, S. Gurumurthy, A. Smola, and K. Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *International Conference on World Wide Web*, 2012.

"Adventure"			"Violence"			"Social"		
Aspect	Negative	Positive	Aspect	Negative	Positive	Aspect	Negative	Positive
earth	effects	effects	murder	nudity	violence	social	attempts	ultimately
human	special	visual	police	exploitation	cinematography	moral	result	level
space	cgi	adventure	killer	female	genre	society	sense	dramatic
world	sci-fi	exciting	crime	pace	solid	point	material	intelligent
planet	mess	sci-fi	cop	violent	gritty	question	superficial	contemporary
fight	monsters	human	mystery	tension	stylish	human	shallow	strength
action	science	impressive	detective	nasty	highly	god	fails	essentially
alien	lack	sets	case	gratuitous	sinister	act	trite	complexity
science	cg	epic	death	naked	inspired	nature	ugly	sympathetic
humans	effort	spectacular	dead	sleazy	engaging	issues	one-dimensional	genuinely
save	giant	cool	thriller	brutal	macabre	men	grotesque	compelling
kill	mindless	evil	victim	hilarious	blood	personal	banal	understated
battle	silly	elements	murdered	murderous	vicious	culture	awkward	sharp
crew	fairly	set	murders	low-budget	nasty	behavior	satirical	equally
attack	sci	created	criminal	trashy	compelling	conflict	excessive	thoughtful

Table 6: Top topic words from ϕ_a for three topics measure by aggregating all $\theta_{u,m}$ across reviews. The aspect labels (adventure, violence, social) are manually assigned.

[8] Y. Koren. Collaborative filtering with temporal dynamics. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 447–456, 2009.

[9] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[10] A. Lazaridou, I. Titov, and C. Sporleder. A bayesian model for joint unsupervised induction of sentiment, aspect and discourse representations. In *Annual Meeting of the Association for Computational Linguistics*, pages 1630–1639, 2013.

[11] H. Ma, H. Yang, M. R. Lyu, and I. King. SoRec: Social Recommendation Using Probabilistic Matrix Factorization. In *Conference on Information and Knowledge Management*, pages 931–940, 2008.

[12] J. McAuley and J. Leskovec. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Conference on Recommender Systems*, pages 165–172, 2013.

[13] J. J. McAuley, J. Leskovec, and D. Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *International Conference on Data Mining*, pages 1020–1025, 2012.

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

[15] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Neural Information Processing Systems Conference*, pages 1257–1264, 2007.

[16] I. Porteous, E. Bart, and M. Welling. Multi-HDP: A non parametric bayesian model for tensor factorization. In D. Fox and C. Gomes, editors, *Conference on Artificial Intelligence*, pg. 1487–1490. 2008.

[17] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In W. Cohen, A. McCallum, and S. Roweis, editors, *International Conference on Machine Learning*, volume 307, pages 880–887. ACM, 2008.

[18] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009 4:2, Jan. 2009.

[19] C. Tan, E. H. Chi, D. Huffaker, G. Kossinets, and A. J. Smola. Instant foodie: Predicting expert ratings from grassroots. In *Conference on Information and Knowledge Management*, 2013.

[20] I. Titov and R. Mcdonald. A Joint Model of Text and Aspect Ratings for Sentiment Summarization. In *Annual Meeting of the Association for Computational Linguistics*, pages 308–316, Columbus, Ohio, 2008.

[21] H. M. Wallach. Topic Modeling: Beyond Bag-of-words. In *International Conference on Machine Learning*, pages 977–984, 2006.

[22] C. Wang and D. M. Blei. Collaborative Topic Modeling for Recommending Scientific Articles. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 448–456, 2011.

[23] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis without aspect keyword supervision. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 618–626, 2011.

[24] M. Weimer, A. Karatzoglou, Q. Le, and A. J. Smola. Cofi rank - maximum margin matrix factorization for collaborative ranking. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, 2008.

[25] S.-H. Yang, B. Long, A. Smola, H. Zha, and Z. Zheng. Collaborative competitive filtering: learning recommender using context of user choice. In W.-Y. Ma, J.-Y. Nie, R. A. Baeza-Yates, T.-S. Chua, and W. B. Croft, editors, *Research and Development in Information Retrieval*, pages 295–304. ACM, 2011.

[26] X. Zhao, J. Jiang, H. Yan, and X. Li. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Conference on Empirical Methods in Natural Language Processing*, pages 56–65, 2010.