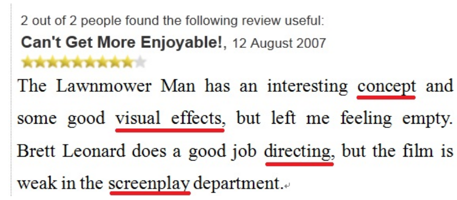
2012 IEEE 12th International Conference on Data Mining

# Collaborative Filtering with Aspect-based Opinion Mining: A Tensor Factorization Approach

Yuanhong Wang*,Yang Liu*, Xiaohui Yu*†
*School of Computer Science and Technology, Shandong University, Jinan, Shandong, China
†School of Information Technology, York University, Toronto, Canada
wyh_1988@163.net, yliu@sdu.edu.cn, xhyu@yorku.ca

***Abstract*—Collaborative filtering (CF) aims to produce user specific recommendations based on other users' ratings of items. Most existing CF methods rely only on users' *overall* ratings of items, ignoring the variety of opinions users may have towards different aspects of the items. Using the movie domain as a case study, we propose a framework that is able to capture users' opinions on different aspects from the textual reviews, and use that information to improve the effectiveness of CF. This framework has two components, an opinion mining component and a rating inference component. The former extracts and summarizes the opinions on multiple aspects from the reviews, generating ratings on the various aspects. The latter component, on the other hand, infers the overall ratings of items based on the aspect ratings, which forms the basis for item recommendation. Our core contribution is in the proposal of a tensor factorization approach for the rating inference. Operating on the tensor composed of the overall and aspect ratings, this approach is able to capture the intrinsic relationships between users, items, and aspects, and provide accurate predictions on unknown ratings. Experiments on a movie dataset show that our proposal significantly improves the prediction accuracy compared with two baseline methods.**

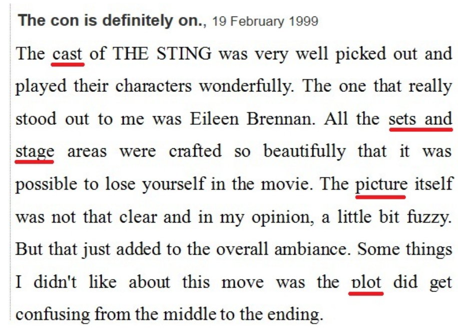
***Keywords*-Opinion Mining, Sentiment Analysis, Collaborative Filtering, Recommendation System, Tensor Factorization**



2 out of 2 people found the following review useful:
**Can't Get More Enjoyable!**, 12 August 2007

The Lawnmower Man has an interesting concept and some good visual effects, but left me feeling empty. Brett Leonard does a good job directing, but the film is weak in the screenplay department.

(a) A review for *The Lawnmower Man*



**The con is definitely on.**, 19 February 1999

The cast of THE STING was very well picked out and played their characters wonderfully. The one that really stood out to me was Eileen Brennan. All the sets and stage areas were crafted so beautifully that it was possible to lose yourself in the movie. The picture itself was not that clear and in my opinion, a little bit fuzzy. But that just added to the overall ambiance. Some things I didn't like about this move was the plot did get confusing from the middle to the ending.

(b) A review for *The Sting*

Figure 1. Two examples of reviews commenting on different aspects.

## I. Introduction

Collaborative filtering (CF) is a widely used technique in recommender systems, which recommend items to a particular user based on other users' ratings. These ratings, often in the form of a scalar such as 1-10 stars, represent people's overall opinions about items. However, the overall ratings cannot provide us more detailed information. For example, a user giving a movie 2-star rating may indicate that the user considers this movie to be bad as a whole. However, it is still possible that he likes some aspects (e.g., storyline, music) of the movie very much.

For example, consider the reviews shown in Figure 1, which are obtained from the IMDB website.

Apparently, both reviews are positive on the topic, but they comment on different aspects. Figure 1(a) provides both numerical ratings and textual reviews. But clearly a simple rating cannot capture the multi-facet nature of the opinion. The problem gets even worse in Figure 1(b), where numerical ratings are not available at all. For conventional CF methods that rely on the numerical ratings, having a lot of missing ratings may lead to the cold start and data sparsity problems, which may greatly affect their performance.

On the other hand, although a lot of research on opinion mining [8] has been conducted to extract subjective opinions from reviews, little has been done in exploring the use of extracting opinions and sentiments from online reviews for more accurate rating estimation in CF.

We therefore propose a new CF framework that integrates multi-faceted opinions in the reviews into the CF process, in order to tap the rich sentiment information embedded in the reviews, and to alleviate the cold start/data sparsity problems. In particular, our framework consists of two components, namely (1) opinion mining, and (2) rating inference.

The first component extracts and summarizes the multiple aspects of opinions expressed in the reviews, and generates numerical ratings on the different aspects. Generally, an

Corresponding author: Xiaohui Yu, E-mail:xhyu@yorku.ca.

1550-4786/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDM.2012.76



IEEE computer society

opinionated statement consists of two parts: the opinion word (like "excellent" or "bad") and the opinion aspect (the target object that is being evaluated). For the purpose of mining and summarizing opinions at the aspect level, we first employ a double propagation approach to expand opinion words and extract the aspect terms. We then use Latent Dirichlet Allocation [2] to cluster those aspect terms into latent aspects. The corresponding opinions can then be aggregated to get a user's ratings on each of these aspects. Since each review contains multiple opinion aspects, the result of the opinion mining component is a set of rating matrices, each corresponding to one of the aspects.

The second component uses tensor factorization to infer the overall rating a user may give to an item, forming the basis of item recommendation. We focus on exploring optimization techniques for rating estimation. The rating matrices for different aspects, together with the overall ratings, constitute a tensor, i.e., a 3-dimensional array. We explore the use of tensor factorization to capture the underlying latent structure of the tensor, and the result of the factorization can be used for inferring the unknown ratings. The method can be seen as an extension of matrix factorization techniques widely applied in collaborative filtering; it can preserve the multi-dimensional nature of the data and extract the latent factors along each dimension.

Despite the particular implementation choices made in this paper, we note that the proposed framework is general enough such that other opinion mining methods can also be adopted. To test the effectiveness of our proposal, we conduct extensive experiments to compare our model with several state-of-the-art CF approaches using a movie review dataset.

To summarize, we make the following contributions.

- We propose a CF framework that is able to extract fine-grained, multi-faceted opinions from reviews, and integrate them into collaborative filtering.
- We propose a tensor factorization approach to capture the intrinsic multi-way interactions between users, items, and aspects, and to predict the unknown ratings on items. To the best of our knowledge, we are the first to take this approach.
- We conduct extensive experiments on a movie dataset to verify the effectiveness of our approach.

The rest of this paper is structured as follows. Section II discusses related works. Section III defines the problem and presents the new CF framework. Section IV presents the tensor factorization approach for rating inferences. Section V discusses the methods to obtain the ratings on various aspects. Section VI presents the experimental results. Section VII concludes this paper.

## II. Related Work

### A. Collaborative Filtering

Collaborative filtering brings together the opinions of large interconnected communities and recommends items to individual users by the known rated items. See [3] for an excellent survey of the various CF approaches. An approach closely related to our work is Matrix Factorization (MF) (e.g., matrix factorization[5], and maximum margin matrix factorization(MMMF) [10]). The MMMF is a matrix factorization strategy based only on the known entries in the matrix, but the model is hardly scalable. Different from their work, we introduce the concept of tensor, which extends CF to the N-dimensional case, and provide more freedom to integrate opinion information.

In a CF system, the data sparsity problem happens when the ratio is too small to provide enough information for CF for effective predictions. To address this problem, Moshfeghi et al.[7] attempt to take additional resources, e.g, emotions and semantic features, as external asset to facilitate a better rating estimation. Our work falls into the same group of their study. However, different from their method where a linear comination is taken to capture the effects of various factors to the integrated rating, our proposal does not require a linear assumption.

### B. Opinion Mining & Sentiment Analysis

Opinions are commonly subjective expressions describing people's feelings, sentiments or emotions toward objects or events. The area of mining these opinions is opinion mining or sentiment analysis. A survey [8] summarizes various techniques in this field. Hu and Liu [4] introduce aspect-based opinion mining model. Qiu et al. [9] extracts target aspects and their associated opinion words simultaneously by exploiting syntactic relations between them.

There are few works exploring the use of opinions in CF. A rating inference approach is applied in [6] to incorporate textual reviews into CF. Our idea is that a user opinion is expressed in multiple aspects, all of which contribute to the overall rating. Moshfeghi et al.[7] take additional resources, e.g, emotions and semantic features, as external asset to facilitate a better rating estimation. However, we consider that the relationship between different opinion aspects and the rating can be non-linear, and explore tensor factorization strategies to understand their underlying connection.

## III. Problem Statement and Framework

### A. Problem Statement

Users of a movie review website typically give numerical ratings (e.g., 7 out of 10 stars) and write textual reviews for the movies they have watched. The traditional CF problem can be modeled by a triplet ($U$, $M$, $\mathbf{R}$), where

- $U$ is the user identifier set $\{u_1, u_2, u_3, \ldots, u_I\}$,
- $M$ is the movie identifier set $\{m_1, m_2, m_3, \ldots, m_J\}$,

- $\mathbf{R}$ is the rating matrix of size $I \times J$, where the entry $r_{ij}$ denotes the value with which $u_i$ rates $m_j$.

The goal of CF is to predict the unknown entries $r_{ij}$ in $\mathbf{R}$ based on the known ones. Note that here $r_{ij}$ is for the *overall rating* of the movie, and most existing CF approaches make prediction only relying on those ratings.

However, a review usually contains the user's opinions on several different aspects (e.g., production, plot, and music) of a movie. For a given number ($K$) of aspects, we use $r_{ij}^k$ to denote the *aspect rating* that user $u_i$ gives for the $k$th aspect of the movie $m_j$. Thus we get $K$ aspect rating matrices $\mathbf{R}^1$, $\mathbf{R}^2$, $\ldots$, $\mathbf{R}^K$ similar to $\mathbf{R}$. Since the users' opinions on the different aspects no doubt affect the overall opinion, we assemble the overall rating matrix $\mathbf{R}$, and the $K$ aspect rating matrices $\mathbf{R}^1$, $\mathbf{R}^2$, $\ldots$, $\mathbf{R}^K$, into a 3-dimensional array, i.e., a 3rd-order tensor $\mathcal{R}$. Figure 2 shows the frontal slices of $\mathcal{R}$. The problem of CF with multiple aspects of opinions can then be performed based on the new triplet $(U, M, \mathcal{R})$, where $\mathcal{R}$ is a 3rd-order tensor of size $I \times J \times (K+1)$, and the goal is to predict the overall rating $r_{ij}$ for the movie $j$ not yet rated by the user $i$.

### B. Framework

We propose a framework to exploit the multiple aspects of opinions in the reviews for collaborative filtering. This framework mainly consists of two parts, namely, (1) aspect and opinion mining, and (2) rating inference based on tensor factorization. Figure 3 illustrates the general procedure of the CF process.

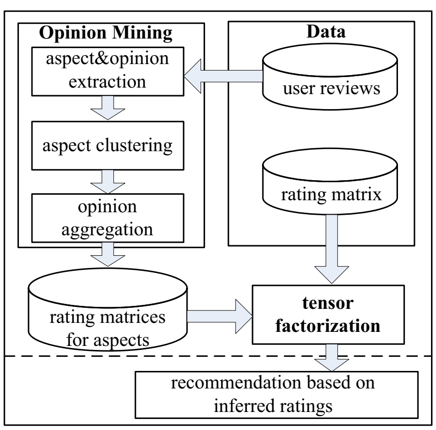



Figure 3. Overview of the recommendation framework

## IV. Tensor Factorization for Rating Inference

For our CF task, tensor factorization is an excellent way of capturing the intrinsic interactions between the three dimensions: users, movies, and aspects. Existing approaches have considered the ratings on different aspects separately and then aggregate them through linear combination (e.g., [12]), but that cannot take into consideration the dependencies between different aspects. In fact, ratings on different aspects are interdependent; for example, a poorly written character may often also leads to a poor rating on acting. It is important to take this into account. In the two-dimensional case, Latent Semantic Indexing (through the mathematical operation of matrix Singular Value Decomposition, or, SVD) proves effective in establishing the associations between different terms in a document. Tensor factorization, the generalization of SVD to multi-dimensional arrays (tensors), can similarly help uncover the complex relationships between aspects, as well as between users and movies, all through one operation.

There are multiple ways of computing the factorization. We adopt the CP-WOPT method [1], which can effectively decompose a high-order tensor to a sum of rank-one tensors in a scalable fashion. Figure 4 illustrates the CP decomposition for the tensor $\mathcal{R}$ in Figure 2. Let us assume the rank of $\mathcal{R}$ is $R$ and $r_{ijk}$ is the entry of $\mathcal{R}$, CP decomposition is defined by *factor matrices* $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ of sizes $I \times R$, $J \times R$, and $(K+1) \times R$ respectively, such that

$$r_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} \tag{1}$$

for all $i = 1, \ldots, I$, $j = 1, \ldots, J$, and $k = 1, \ldots, K+1$.

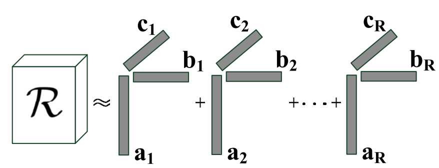



Figure 4. Illustration of an R-component CP for the tensor $\mathcal{R}$

We consider the CP decomposition as a weighted least squares problem, which uses only the known entries of the tensor and minimizes the following objective function

$$f_{\mathcal{W}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \frac{1}{2} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K+1} \left\{ w_{ijk} \left( r_{ijk} - \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} \right) \right\}^2 \tag{2}$$

where $\mathcal{W}$, which has the same size as $\mathcal{R}$, is a non-negative weight tensor defined as

$$w_{ijk} = \begin{cases} 1 & \text{if } r_{ijk} \text{ is known,} \\ 0 & \text{if } r_{ijk} \text{ is unknown.} \end{cases} \tag{3}$$

for all $i = 1, \ldots, I, j = 1, \ldots, J, k = 1, \ldots, K+1$.

We use the gradient descent method to find the factor matrices $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$. By computing the partial derivatives of $f_{\mathcal{W}}$ with respect to each element of the factor matrices, we can get the gradient of (2). Note that the gradient does not need to be computed element-wise because it can be computed efficiently using tensor operations.

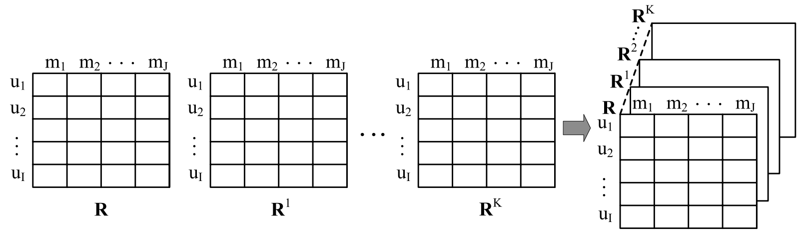



Figure 2. Tensor $\mathcal{R}$ comprised of $\mathbf{R}, \mathbf{R}^1, \mathbf{R}^2, \ldots, \mathbf{R}^K$

For a more detailed description, we first define some notations. Given two 3rd-order tensors $\mathcal{X}$ and $\mathcal{Y}$ of equal size $I_1 \times I_2 \times I_3$, their elementwise product, denoted by $\mathcal{X} * \mathcal{Y}$, is defined as $(\mathcal{X} * \mathcal{Y})_{i_1 i_2 i_3} = x_{i_1 i_2 i_3} y_{i_1 i_2 i_3}$ for all $1 \leq i_n \leq I_N$. In addition, given a 3rd-order tensor $\mathcal{X}$ of size $I_1 \times I_2 \times I_3$, its norm is $\|\mathcal{X}\| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \sum_{i_3=1}^{I_3} x_{i_1 i_2 i_3}^2}$. With three matrices $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ of size $I_1 \times R, I_2 \times R, I_3 \times R$, $[\![\mathbf{X}, \mathbf{Y}, \mathbf{Z}]\!]$ defines an $I_1 \times I_2 \times I_3$ tensor whose elements are $([\![\mathbf{X}, \mathbf{Y}, \mathbf{Z}]\!])_{i_1 i_2 i_3} = \sum_{r=1}^{R} x_{i_1 r} y_{i_2 r} z_{i_3 r}$. Finally, given a 3rd-order tensor $\mathcal{X}$ of size $I_1 \times I_2 \times I_3$, $\mathcal{X}$ can be rearranged as a matrix, which is called matricization. The mode-n matricization of $\mathcal{X}$ is denoted by $\mathbf{X}_{(n)}$. Tensor element $\mathcal{X}_{i_1 i_2 i_3}$ maps to matrix $\mathbf{X}_{(n)}$' element $(i_n, j)$ where

$$j = 1 + \sum_{k=1, k \neq n}^{3} (i_k - 1) J_k \tag{4}$$

with

$$J_k = \begin{cases} 1 & \text{if k = 1, or if k = 2 and n = 1,} \\ \prod_{m=1, m \neq n}^{k-1} I_m & \text{otherwise.} \end{cases}$$

With the notation, equation (2) can be rewritten as

$$f_{\mathcal{W}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \frac{1}{2} \|\mathcal{W} * (\mathcal{R} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!])\|^2 \tag{5}$$

Equation (5) is equivalent to

$$f_{\mathcal{W}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \frac{1}{2} \|\mathcal{S} - \mathcal{T}\|^2 \tag{6}$$

where $\mathcal{S} = \mathcal{W} * \mathcal{R}$ and $\mathcal{T} = \mathcal{W} * [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$. Both $\mathcal{W}$ and $\mathcal{R}$ do not change during the iterations, so we pre-compute $\mathcal{S}$.

In matrix notation, the partial derivatives of $f_{\mathcal{W}}$ are

$$\begin{aligned} \frac{\partial f_{\mathcal{W}}}{\partial \mathbf{A}} &= \left(\mathbf{T}_{(1)} - \mathbf{S}_{(1)}\right)(\mathbf{C} \odot \mathbf{B}) \\ \frac{\partial f_{\mathcal{W}}}{\partial \mathbf{B}} &= \left(\mathbf{T}_{(2)} - \mathbf{S}_{(2)}\right)(\mathbf{C} \odot \mathbf{A}) \\ \frac{\partial f_{\mathcal{W}}}{\partial \mathbf{C}} &= \left(\mathbf{T}_{(3)} - \mathbf{S}_{(3)}\right)(\mathbf{B} \odot \mathbf{A}) \end{aligned} \tag{7}$$

$\mathbf{S}_{(n)}$ and $\mathbf{T}_{(n)}$ can be obtained by equation (4), and the symbol $\odot$ denotes the Khatri-Rao product and is defined as follows for $\mathbf{C}$ and $\mathbf{B}$:

$$\mathbf{C} \odot \mathbf{B} = [\mathbf{c_1} \otimes \mathbf{b_1} \quad \mathbf{c_2} \otimes \mathbf{b_2} \quad \cdots \quad \mathbf{c_R} \otimes \mathbf{b_R}] \tag{8}$$

where $\otimes$ denotes the vector Kronecker product.

After getting the gradient, we can use a standard first-order optimization method such as nonlinear conjugate gradient (NCG) or limited-memory BFGS to get the factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. The predicted rating value that $u_i$ will give for $m_j$ is

$$\hat{r}_{ij} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{1r} \tag{9}$$

## V. Inferring Aspect Ratings

Our approach to infer the aspect ratings involves three steps, 1) extracts the aspect terms and the corresponding opinion words from the reviews; 2) clusters aspect terms into $K$ latent aspects; 3) computes the ratings on those aspects based on the opinions expressed towards the constituent aspect terms. In what follows, we describe each step in more detail.

### A. Extracing Aspect Terms and Opinions

The movie aspect terms and opinion words in a user's review can be very different from each other's. Manual extraction approaches are not proper because they are time consuming and tend to have a poor recall. Thus we adapt a semi-supervised method called double propagation [9] to perform the extraction task.

Double propagation makes use of the syntactic relations between opinion/aspect terms and other opinion/aspect terms to expand opinion lexicon and extract aspect terms simultaneously. It makes use of dependency grammar to describe syntactic relations. The basic idea of double propagation is to extract opinion words (or aspects) iteratively using known and extracted (in previous iterations) opinion words and aspects through the identification of direct dependencies. As an example, in a movie review sentence *The film has a great plot*, the adjective *great* is parsed as directly depending on the noun *plot*. If we know *great* is an opinion word and have a rule like "a noun on which an opinion word directly depends by the dependency type *amod (adjectival modifier)* is regarded as an aspect term", we can take *plot* as a movie aspect term. Similarly, we could extract the adjective *great* as an opinion word if we know *plot* is a movie aspect.

In our experiment, we employ Stanford CoreNLP[1] to do

[1] http://nlp.stanford.edu/software/corenlp.shtml

POS tagging and get dependency parses and use opinion lexicon[2] as seed. Note that the double propagation method can detect the negations/contrary words in the sentences. We assign sentiment polarities to the newly extracted opinions by contextual environment.

### B. Opinion Summarization

The movie aspects that a user can comments on in his or her review are not restricted, so there will be many different aspect terms referring to the same aspect group. In order to aggregate the sentiment for these different aspect terms, we need to map the extracted movie aspect terms to one or more latent aspects. We use Latent Dirichlet Allocation [2], which has been successfully applied in numerous settings, for this task. The input of the LDA procedure is the set of reviews and the aspect terms, and the output is a set of aspects, each consisting of a set of aspect terms. The number of aspects can be determined experimentally.

### C. Computing Aspect Rating Matrices

In Algorithm 1, we describe how to compute $\mathbf{R}^1, \mathbf{R}^2, \ldots, \mathbf{R}^K$ defined in Section III.

We first calculate the numbers of positive and negative opinion words about one aspect, then use the ratio (number of positive words)/(total number of opinion words) as the aspect rating value for this aspect. As we would like to give equal weight to the overall ratings and aspect ratings, we normalize the aspect rating value so that they fall in the same range. If there are no opinion words about an aspect, its rating is set to zero naturally. It is worth noting that an aspect term can belong to several different aspects, and correspondingly, the opinions expressed on that aspect term are counted toward the ratings of multiple aspects. The end result of this step are then used to construct the tensor in Section IV.

**Algorithm 1** algorithm for calculating $r_{ij}^k$

**Require:**
The clusters of movie aspects, $C_1, C_2, \ldots, C_K$;
The set of aspect terms, *A*;
The set of opinion words, *O*;
The review user $u_i$ wrote for movie $m_j$, $rev_{ij}$;
**Ensure:**
The aspect rating value $r_{ij}^k (k = 1, 2, \ldots, K)$;
**for** each $k \in [1, K]$ **do**
  initialize variables $p_k = 0, n_k = 0$;
**end for**
//find all aspect and opinion terms in $s$
**for** each sentence $s$ in $rev_{ij}$ **do**
  initialize empty sets $S_1, S_2$;
  **for** all aspects $a \in A$ and opinion words $o \in O$ **do**
    **if** $a$ appears in $s$ **then**
      $S_1 = S_1 \cup \{a\}$; //aspect terms set
    **end if**
    **if** $o$ appears in $s$ **then**
      $S_2 = S_2 \cup \{o\}$; //opinion terms set
    **end if**
  **end for**
  //aggregate opinions for $C_k$
  **for** each $a \in S_1$ **do**
    **for all** $C_k$ such that $a \in C_k$ **do**
      **for all** $o \in S_2$ **do**
        **if** the opinion polarity of $o$ is positive **then**
          $p_k = p_k + 1$;
        **else**
          $n_k = n_k + 1$;
        **end if**
      **end for**
    **end for**
  **end for**
  **for** $k \in [1, K]$ **do**
    **if** $p_k + n_k > 0$ **then**
      $r_{ij}^k = 10 * p_k/(p_k + n_k)$;
    **else**
      $r_{ij}^k = 0$;
    **end if**
  **end for**
**end for**

## VI. Experiments

### A. IMDB Data Set

Several datasets (e.g. Jester Joke[3], MovieLens[4]) for recommender systems only provide star or numerical ratings. So we collected a dataset containing the star rating values(1-10 stars) and corresponding textual reviews of about 1600 movies from IMDB[5]. In order to have a more robust recommendation performance, we remove from the raw dataset all users who rate less than 20 movies.

Table I shows some statistics on the raw and reduced datasets. We can observe the sparsity is improved from 0.14% to 3.70%. The data sparsity is defined as

$$sparsity = \frac{\#reviews}{\#users * \#movies} \tag{10}$$

Table I
DATA SET STATISTICS

| | raw | reduced |
|---|---|---|
| #movies | 1606 | 1525 |
| #users | 83585 | 946 |
| #reviews | 193266 | 53353 |
| sparseness | 0.14% | 3.70% |

### B. Experimental Setup

To evaluate recommendation quality of our framework, we use RMSE (Root Mean Squared Error) as evaluation metrics. We first divide the data set into a training and a testing set. To determine the parameter sensitivity, we further divide the training data set into a training portion and a testing portion. We perform a 10-fold cross validation. For each testing fold, we calculate the RMSE between the predicted ratings of each model and the actual ratings, and then take an average

[2]http://www.cs.uic.edu/∼liub/FBS/sentiment-analysis.html
[3]http://www.ieor.berkeley.edu/∼goldberg/jester-data/
[4]http://www.grouplens.org/node/12
[5]http://www.imdb.com

over the ten fold to measure the model's performance.

In order to study the effect of data sparsity, we randomly remove some ratings from the data set so that the data sparsity varies from 3.7% to 1.7%. To determine the impact of parameter K (number of clusters), we then vary the value of K from 2 to 20. We notice that the values of RMSE fluctuate slightly between 2.05 to 2.15, and the best result is observed at $K = 6$ for our dataset. Therefore, we take $K = 6$ for the rest of our experiments.

We use two baselines in our experiments. First, since we want to investigate the effect of opinions in CF, we use a baseline only relying on the star ratings. Because our model (identified as TF) can be considered as an extension of matrix factorizations in multi-dimensional arrays, we choose a widely used matrix factorization approach (identified as MF) [11] for comparison. Second, we report the performance of the framework (identified as MR) by Moshfeghi et al. [7], which has been shown to outperform other state-of-the-art approaches. The original MR framework considers three types of feature spaces and can incorporate emotion information. For a fair comparison, we implement an MR framework that only makes use of the movie space and the emotion space which are also used in our model.

### C. Main Result & Discussion

The results of our experiments are shown in Figure 5. We performed a paired t-test between the RMSE measures to check the significance of difference from the baselines (MF and MR in Figure 5), which indicates that our model improves the prediction accuracy with the confidence level $p < 0.05$.

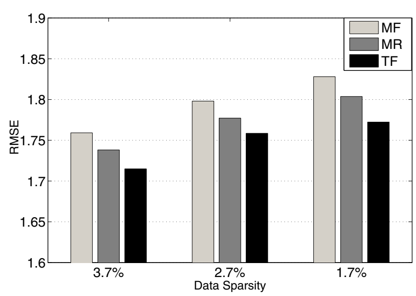



Figure 5. Experiment results for different sparsities (smaller RMSE is better)

In Figure 5, we observe that both our model (TF) and the second baseline(MR) have a better result than the first baseline (MF), though they respectively exploit different ways to combine opinion information with CF. Since our model and MR make use of both star ratings and extracted opinion ratings, while MF uses star ratings only. This coincides with our intuition that incorporating the opinion information extracted from reviews can improve the precision of recommender systems. In addition, our model outperforms the two baselines for different levels of data sparsity. Figure 5 also shows that the prediction accuracy of each model decreases as the data becomes sparser. But the rate of accuracy change is different. When the data sparsity changes from 3.7% to 1.7%, the RMSE of our model increases at a lower rate than the baselines. This shows our model can adapt better to lower data sparsity.

## VII. Conclusions

In this paper we have shown the opinions extracted from movie reviews can be integerated into CF to improve recommendation quality. Our fundamental idea is that a user's overall ratings on a movie can be reflected by his or her opinions on different movie aspects (e.g. direction, animation, plot). To model this relationship, we organize both the overall rating and the aspect ratings for a movie into a tensor, and employ a tensor factorization approach to explore the latent structure of the tensor. This approach preserves the multi-way nature of the data and captures the underlying factors in each dimension of a tensor. Our experiment results indicate the extracted opinions indeed play a role in improving the prediction accuracy compared with the baselines. The effectiveness of the proposed approach is even more apparent when the ratings are sparse.

## References


[1] E. Acar, D. Dunlavy, T. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.

[2] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[3] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[4] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

[5] N. Jakob, S. Weber, M. Müller, and I. Gurevych, "Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations," in *Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*. ACM, 2009, pp. 57–64.

[6] C. Leung, S. Chan, and F. Chung, "Integrating collaborative filtering and sentiment analysis: A rating inference approach," in *Proceedings of The ECAI 2006 Workshop on Recommender Systems*. Citeseer, 2006, pp. 62–66.

[7] Y. Moshfeghi, B. Piwowarski, and J. Jose, "Handling data sparsity in collaborative filtering using emotion and semantic based features," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*. ACM, 2011, pp. 625–634.

[8] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[9] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Computational Linguistics*, vol. 37, no. 1, pp. 9–27, 2011.

[10] N. Srebro, J. Rennie, and T. Jaakkola, "Maximum-margin matrix factorization," *Advances in neural information processing systems*, vol. 17, no. 5, pp. 1329–1336, 2005.

[11] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Major components of the gravity recommendation system," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 80–83, 2007.

[12] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: a rating regression approach," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 783–792.