A Triple Wing Harmonium Model for Movie Recommendation

Haijun Zhang, *Member, IEEE*, Yuzhu Ji, Jingxuan Li, and Yunming Ye, *Member, IEEE*

***Abstract*—A new triple wing harmonium (TWH) model that integrates text metadata into a low-dimensional semantic space is proposed for the application of content-based movie recommendation. The text metadata considered here include movie synopsis, actor list, and user comments. We develop a new TWH model projecting these multiple textual features into low-dimensional latent topics with different probability distribution assumptions. A contrastive divergence (CD) algorithm is used for efficient learning and inference. Experimental results suggest that the proposed method performs better than the state-of-the-art algorithms for movie recommendation.**

***Index Terms*—Harmonium model, movie recommendation, multiple features, text metadata.**

I. Introduction

THE RAPID development of electronic platforms (such as the Internet) has made massive amount of information available and easy access to people's lives, which leads to a growing demand of developing highly effective systems that are capable of filtering out irrelevant information and selecting content to meet user needs. Recommender systems are just such systems used to facilitate the above process. They have been successfully applied in diverse areas, such as negotiation [1], business process modeling [2], vehicle information systems [3], to name just a few.

The widely used techniques of recommender systems are collaborative filtering [4] and content-based recommending [5]. Collaborative filtering relies on users' own explicit and implicit preferences, the preferences of other users, and the attributes of all users and items. It assumes that a given user's preferences are similar to another user of the system and that a sufficient number of user ratings is available. There are two typical issues that stand out as problematic in collaborative filtering.

1) Items that have not been rated by a sufficient number of users cannot be effectively recommended.
2) A collaborative approach is not able to recommend items that no one has yet rated or purchased.

This is the so-called cold-start problem. Content-based recommending, on the other hand, can help to overcome these issues by inferring similarities between existing and new users, as well as between existing and new items [6]. It recommends items based on content features about the item itself rather than on the preferences of other users. Finding an appropriate function to evaluate the similarity between two items is of utmost importance in a content-based recommender system.

Movie recommendation, one of important recommending applications, has attracted considerable attention for its great promise in digital television domain and online video share platforms [7]–[11]. In this paper, we develop a triple wing harmonium (TWH) model for movie recommendation. The TWH model aims at providing a better latent representation for movies using multiple textual features, namely movie synopsis, actor list, and user comments. Different from previous work that usually considers synopsis, actor list, categories, directors, and keywords, as text description of a movie, this paper incorporates user comments into the recommender system based on two observations.

1) Since user comments express personal opinions, two similar movies may contain similar comments from different users; thus, combining the similarity contributed by user comments into the similarity from pure text description enables us to formulate a more "compact similarity," which is beneficial to recommendation.
2) The variety found in the comments may include controversy, sentiment, and many popular topics; this variety can help to relieve the *serendipity* problem, i.e., the content-based systems tend to produce recommendations with a limited degree of novelty [12].

Incorporating these multiple features into a unified movie representation, however, is difficult. It is possible to use the traditional "bag of words" model [13] to combine these features, but it needs to identify the weights of different features in advance. To overcome this, a TWH model is developed to learn a latent representation for each movie by combining these features in an unsupervised manner. Specifically, all the user comments in a data set are first mapped onto a grid by employing a widely used clustering technique, self-organizing map (SOM) [14]. The comment positions on the grid for each movie are then used to construct a vector, where each component represents the number of occurrences that user comments have been mapped onto a certain position. In the TWH, we model term counts in the movie synopsis with a conditional Poisson distribution, actors present in a movie with a conditional Bernoulli distribution, and the features generated from user comments with another Poisson distribution, respectively. Latent topics are treated as a conditional binomial distribution involving weighted matrices and multiple features. The

Manuscript received July 01, 2015; revised August 14, 2015; accepted August 19, 2015. Date of publication August 31, 2015; date of current version February 02, 2016. This work was supported by the Natural Science Foundation of China under Grant 61300209 and Grant 61572156. Paper no. TII-15-1080.

The authors are with Harbin Institute of Technology Shenzhen Graduate School, Xili University Town, Shenzhen 518055, China (e-mail: hjzhang@hitsz.edu.cn; Nero2074@outlook.com; jing_910307@163.com; yeyunming@hit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2015.2475218

performance of TWH is investigated in the applications of movie recommendation. We show the superiority of TWH with respect to several evaluation measures compared to the state-of-the-art methods. We also investigate the influence of number of latent topics and different learning methods for TWH inference.

This paper is organized as follows. Related work is briefly reviewed in Section II. Multiple features extraction framework is introduced in Section III. In Section IV, a new TWH model is presented in detail. Section V introduces the contrastive divergence (CD) algorithm for TWH learning and inference. Experimental results followed by discussions are presented in Section VI. This paper ends with conclusion and future work propositions in Section VII.

II. Related Work

There is a large body of literatures working on both collaborative filtering and content-based recommender systems. Due to space limitation, we only review the work that belongs to the movie recommending domain. Mak *et al.* [7] proposed a recommender system by using text categorization techniques to learn from movie synopses. In the recommending process, the user was asked to rate a minimum number of movies into six categories: 1) terrible; 2) bad; 3) below average; 4) above average; 5) good; and 6) excellent. In the similar way, Mukherjee *et al.* [8] introduced a Movies2Go system that learns user preferences from the synopsis of movies rated by the user. The innovative aspect of Movies2Go lies in the use of voting strategies to allow multiple individuals with conflicting preferences and to manage them in a single user. In a later study, Szomszor *et al.* [9] described a movie recommendation system built purely on the keywords assigned to movies via collaborative tagging. For an active user, the system produces recommendations relying on the similarity between the keywords of a movie and those of the tag-clouds of movies the user rated. The presented results show that the performance can be improved by combining tag-based profiling techniques compared with traditional content-based recommender strategies. In a recent study, an external source Wikipedia was used to estimate similarity between movies in order to provide more accurate predictions for the Netflix Prize competition [10]. Concretely, the content and the hyperlink structure of Wikipedia articles are exploited to identify similarities between movies. A similarity matrix indicating the degree of similarity of each pair of movies is formulated. A k-nearest neighbor and a pseudo-singular value decomposition (SVD) algorithm are used to predict user ratings. But these methods do not demonstrate significant performance improvement. To exploit the advantages of both collaborative and content-based filtering methods, Lekakos and Caravelas [11] proposed a hybrid approach relying on the monitoring of certain parameters that trigger either a content-based or a collaborative filtering prediction.

From the aspect of item representation, a movie can be represented by a set of features, e.g., actors, directors, genres, synopsis, plot summaries, reviews, and short abstract. When the movie is described by the same set of attributes, it can be represented by a word vector [12]. Thus, many bag of words models, widely used in documents, can be employed to learn a movie representation. The between-movie similarity can be measured by the widely used *cosine* distance. However, the features extracted from a movie may play different roles in the recommender system. Automated capturing the weights of different features for recommendation is critical. Traditional methods, such as the vector space model (VSM) [13], the latent semantic indexing (LSI) [15], the probabilistic LSI (PLSI) [16], the latent Dirichlet allocation (LDA) [17], and the rate adapting Poisson (RAP) model [18], mainly consider only one type of features, for instance, the *term frequency* (*tf*) extracted from movie synopsis. In line with the item representation, the TWH model presented in this paper aims at learning a unified latent representation with different movie features. The resulting representation can be used in other content-based or hybrid recommending methods.

III. Feature Encoding

The features used in this paper include movie synopsis, actors, and user comments. We believe that these three types of features can provide sufficient information to discriminate the movie between-similarity. It is worth pointing out that other features, such as titles, genres, directors, and short abstracts can be easily grouped into these three types of features.

A. Movie Synopsis

The synopsis can be viewed as a text description of movie content. Similar to document modeling, words from all the synopses are first extracted in the given data set and stemming was subsequently applied to each word. Stems are often used as basic features instead of original words. We then remove the stop words, a set of common words like “a,” “the,” “are,” etc., and store the stemmed words together with the information of the *tf*. A vocabulary is then constructed for forming a histogram vector for each synopsis. Thus, each synopsis can be represented by a vector $\boldsymbol{X}$, in which each component represents the *tf* in a synopsis.

B. Actor List

Another factor affecting on user preference is the actor list, because users may be fans of some actors. For feature representation, we can extract all the actors from the data set. Then, the actor list for each movie can be described by a binary vector $\boldsymbol{Y}$ in which each component indicates the presence or absence of an actor. Thus, this binary state of the actor will capture the movie similarity from the aspect of actor list.

C. User Comment

With the rise of Internet, users can easily share their opinions on some movies through various online platforms. A popular movie may receive hundreds or even thousands of comments from different users. One possible way to represent these features is grouping all the comments for a movie into one single document and describing it with a term vector similar to the synopsis. This simple strategy, however, cannot capture the discriminative information of different user opinions. In this paper, we introduce a new method to project these user comments into a vector. Specifically, we first extract the words from all the comments in a data set. After stemming and removing stop words, a vocabulary is constructed. Each comment can be represented by a histogram vector. But each vector is a sparse vector due to the short length of the comment. In order to make the framework computationally efficient, a typical principle component analysis (PCA) was used to project each histogram vector, associated with a comment, into a d-dimensional feature space (d is much smaller than the original feature dimension, in this paper, we set $d = 100$). It is noted that other advanced dimensionality reduction methods can be used, instead of PCA. We then employ the SOM [14] algorithm to map all the reduced feature vectors onto an $m \times m$ two-dimensional (2-D) grid.

SOM is a versatile unsupervised neural network used for dimensionality reduction, vector quantization, and visualization [14]. It is able to preserve a topologically ordered output map, where input data are mapped into a small number of neurons.

A basic SOM consists of m^2 neurons located on a regular low-dimensional grid, which is usually a 2-D grid. The lattice of the grid is either hexagonal or rectangular. The SOM algorithm is iterative. Each neuron i has a d-dimensional feature vector $\boldsymbol{w_i} = [w_{i1}, \ldots, w_{id}]^T$. At each training step t, a sample data vector $\boldsymbol{x}(t)$ is randomly selected from a training set. The distances between $\boldsymbol{x}(t)$ and all the feature vectors $\{\boldsymbol{w_i}\}$ are calculated. The winning neuron, denoted by c, is the neuron with the feature vector closest to $\boldsymbol{x}(t)$ given by

$$c = \arg\min_i \left(S\left(x(t), w_i\right)\right), \qquad i \in \left\{1, 2, \ldots, m^2\right\} \tag{1}$$

where $S\left(\boldsymbol{x}(t), \boldsymbol{w_i}\right)$ is a distance function between $\boldsymbol{x}(t)$ and $\boldsymbol{w_i}$. In the sequential SOM algorithm, the winner neuron and its neighbor neurons are updated according to the weight-updating rule in a form of

$$w_j(t+1) = \begin{cases} w_j(t) + \eta(t) r_{jc}(t)\left(x(t) - w_j(t)\right) \; \forall j \in N_c \\ w_j(t), \text{ otherwise} \end{cases} \tag{2}$$

where N_c is a set of neighboring neurons of the winning neuron. $\eta(t)$ is the learning rate which decreases monotonically with iteration t in the form of

$$\eta(t) = \eta_0 \cdot \exp\left(-a \cdot \frac{t}{T}\right) \tag{3}$$

where η_0 is the initial learning rate, a is an exponential decaying constant which is set to 3 in this study, and T is a time constant set to the maximum number of iterations. $r_{jc}(t)$ is the neighborhood kernel function that indicates the distance of a neighborhood neuron j with coordinate (x_j^G, y_j^G) to the winning neuron c at position (x_c^G, y_c^G). This neighborhood function is a nonincreasing function that can be taken as a Gaussian function

$$r_{jc}(t) = \exp\left(-\frac{[(x_j^G - x_c^G)^2 + (y_j^G - y_c^G)^2]}{2(\Omega(t))^2}\right) \tag{4}$$

where $\Omega(t)$ is the width of the neighborhood function that decreases monotonically with iteration t in the form of

$$\Omega(t) = \eta_0 \cdot \exp\left(-\frac{t}{T} \cdot \log(\Omega_0)\right) \tag{5}$$

where Ω_0 is the initial width. The detailed SOM training and mapping algorithm can be found in [14].

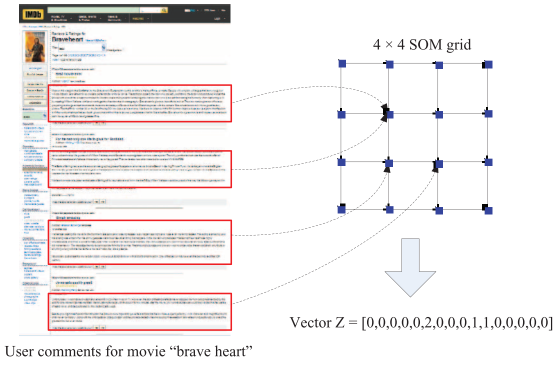

IMDb
Braveheart
4 × 4 SOM grid
Vector Z = [0,0,0,0,0,2,0,0,0,1,1,0,0,0,0,0]
User comments for movie "brave heart"

Fig. 1. Example of the mapping process for user comments.

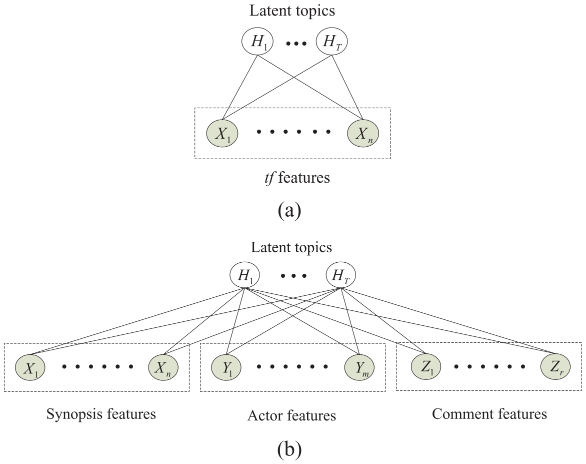


Fig. 2. Topologies of different harmonium models. (a) Basic harmonium. (b) TWH.

The positions of user comments on the grid for each movie are used to construct a vector $\boldsymbol{Z}$ where each element represents the number of occurrences that user comments have been mapped onto a certain position. Here, the new constructed vector is a flat vector with size m^2 transformed from the mapping grid. Fig. 1 illustrates an example of the mapping process. We assume that the movie *Brave Heart* includes four user comments. After SOM training, it is straightforward to map these four comments onto a grid (here, suppose the grid size is 4×4). According to the mapping positions of the comments, we can construct a flat vector [0,0,0,0,0,2,0,0,0,1,1,0,0,0,0,0]. This vector is then treated as the comment features in the TWH modeling process.

IV. TWH

The original harmonium model refers to a family of bipartite undirected graphical models. Fig. 2(a) describes the bipartite topology of a harmonium that contains two layers of nodes. Nodes $X = \{X_i\}$ at the bottom layer represent the observed data and nodes $H = \{H_k\}$ at the top layer denote the latent topics (or hidden units) of the data. For document data, X can represent *tf* feature (i.e., term counts) of each document, and H represents the resultant discriminator by projecting higher dimensional *tf* feature into low-dimensional semantic space. One of the advantages of harmonium model is that the nodes within the same layer are conditionally independent given the nodes in the other layer, which facilitates the generation of harmonium distribution based on two between-layer conditional distributions $p(x|h)$ $(p(x|h) = \prod_i p(x_i|h))$ and $p(h|x)$ $(p(h|x) = \prod_j p(h_j|x))$ [18], [19].

A. Exponential Family Harmonium (EFH)

The EFH model [20], a special class of harmonium models in the exponential family, can be treated as an undirected probability model that combines latent topics in the log-probability domain. The conditional distributions at the two layers and the joint distribution (harmonium random field) are in the form of [20]

$$p(x|h) = \prod_i p(x_i|h) \propto \prod_i \exp\left\{\left(\theta_i + \sum_j W_{ij} g(h_j)\right) f(x_i)\right\} \tag{6}$$

$$p(h|x) = \prod_j p(h_j|x) \propto \prod_j \exp\left\{\left(\eta_j + \sum_i W_{ij} f(x_i)\right) g(h_j)\right\} \tag{7}$$

$$p(x,h) \propto \exp\left\{\sum_i \theta_i f(x_i) + \sum_j \eta_j g(h_j) + \sum_{ij} W_{ij} f(x_i) g(h_j)\right\} \tag{8}$$

where $\{f(x_i)\}$ and $\{g(h_j)\}$ are the sufficient statistics of node $\{x_i\}$ and $\{h_j\}$. $\{\theta_i\}$, $\{\eta_j\}$, and $\{W_{ij}\}$ are the parameters, which can be determined by the learning algorithm. In the above distributions, the global partition function is not explicitly shown, making the harmonium learning more difficult. From the distributions, it is noted that the term $\{W_{ij}\}$ couples the data nodes x to the latent topics h. By learning and inference, latent topics h will be harmonized with the observed data x such that h captures the semantics in x [19].

B. RAP Model

To generate a component-wise nonlinear projection from the input space to the output latent space, Gehler *et al.* [18] extended the EFH model to the RAP model, a more general topology of the EFH. The RAP model couples latent topics to term counts using a conditional Poisson distribution involving a single weight matrix. RAP uses conditional Poisson distribution for the *tf* feature and conditional binomial distribution for the latent topics as follows [18]:

$$p(x|h) = \prod_i \left(\text{Poisson}_{x_i}\left(\alpha_i + \sum_k W_{ik} h_k\right)\right) \tag{9}$$

$$p(h|x) = \prod_k \left(\text{Binomial}_{h_k}\left(\sigma\left(\tau_k + \sum_i W_{ik} x_i\right), M_k\right)\right) \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid function, α_i is the log mean rate of the conditional Poisson distribution for term i, $\tau_k = \log\left(p_k/(1-p_k)\right)$ (p_k is the probability of success), and M_k is the total number of samples for the conditional binomial distribution for topic k. The joint distribution over (x,h) can be expressed as

$$\begin{aligned} p(x,h) \propto \exp\Bigg\{ & \sum_i (\alpha_i x_i - \log(\Gamma(x_i))) \\ & + \sum_k (\tau_k h_k - \log(\Gamma(h_k)) - \log(\Gamma(M_k - h_k))) \\ & + \sum_{ik} W_{ik} x_i h_k \Bigg\} \end{aligned} \tag{11}$$

where $\Gamma(\cdot)$ is the Gamma function. The marginal probability of nodes x is given by

$$\begin{aligned} p(x) \propto \exp\Bigg\{ & \sum_i (\alpha_i x_i - \log(\Gamma(x_i))) \\ & + \sum_k \left(M_k \log\left(1 + \exp\left(\sum_i W_{ik} x_i + \tau_k\right)\right)\right)\Bigg\}. \end{aligned} \tag{12}$$

RAP can model the behavior that the values of the variables at the opposite layer shift the canonical parameters of the variables at the corresponding layer. The variation of $\{\tau_k\}$ decides the impact on the Poisson rate $\{\alpha_i\}$ with the rate adapting property [18], [19].

C. TWH Model

For video and image applications, Xing *et al.* [21] proposed a dual wing harmonium (DWH) model for the fusion of features from text features and image features. In their DWH model, authors directly treated the term counts via a Bernoulli distribution and the whole image color histogram via a multivariate Gaussian distribution. These two types of features were then projected into the latent space with low dimension. This new fusion strategy performs well for image annotation and video classification. Motivated by [21], Zhang *et al.* extended the RAP model to a new DWH model for document data [19]. The architecture of DWH for document data consists of two wings at the bottom layer. One wing represents the observed *tf* feature and the other denotes the sampled term connection frequency (tcf) feature. They treated the *tf* via a Poisson distribution and the *tcf* via a Bernoulli distribution. Encouraging results have been achieved by the DWH model in document retrieval [19]. In this paper, we attempt to extend the RAP model further to a TWH model for movie data. Fig. 2(b) shows the architecture of TWH for movie data that consist of three wings at the bottom layer. Specifically, one wing represents the observed *tf* feature $\{X_i\}$ extracted from the movie synopsis. The second wing represents the actor feature $\{Y_i\}$ sampled from the actor list. The third wing represents the comment feature $\{Z_i\}$ obtained from the SOM grid. Thus, TWH integrates the multiple features, associated with synopsis, actor list and user comments, as low-level features into latent topics as high-level features to represent movie semantics. These three types of features interact with each other through the weighted matrices.

In our TWH, we use conditional Poisson distribution for the *tf* feature obtained from synopsis as follows:

$$p(x_i|h) = \text{Poisson}\left(x_i|\alpha_i + \sum_k W_{ik}h_k\right). \tag{13}$$

The actor list indicates the presence of each actor. So, we use conditional Bernoulli distribution for the binary state of this actor feature as follows:

$$p(y_j|h) = \text{Bernoulli}\left(y_j|\sigma\left(\beta_j + \sum_k U_{jk}h_k\right)\right) \tag{14}$$

where $\{U_{jk}\}$ represents the weighted matrix coupling the actor feature to latent topics, β_j is the learning parameter for the jth component in the actor vector. For the comment feature, we have used an SOM to produce clusters of user comments on the SOM grid. Due to the variety of semantics delivered by user comments, we can use another Poisson distribution to approximately represent the distribution of comments over the SOM grid. The conditional Poisson distribution is in the form of

$$p(z_l|h) = \text{Poisson}\left(z_l|\gamma_l + \sum_k V_{lk}h_k\right) \tag{15}$$

where $\{V_{lk}\}$ represents the weighted matrix coupling the comment feature to latent topics, and γ_l is the learning parameter for the lth component in the comment vector. Finally, the latent topics $\{H_k\}$ follow the conditional binomial distribution depending on a weighted combination of synopsis feature x, binary actor feature Y, and comment feature z in the following way:

$$\begin{aligned} &p(h_k|x, Y, z)\\ &= \text{Binomial}\left(h_k|\sigma\left(\tau_k + \sum_i W_{ik}x_i + \sum_j U_{jk}y_j + \sum_l V_{lk}z_l\right), M_k\right). \end{aligned} \tag{16}$$

We then define the following joint distribution to be consistent with the above conditional distributions:

$$\begin{aligned} p(x, Y, z, h) \propto \exp\Bigg\{ &\sum_i (\alpha_i x_i - \log(\Gamma(x_i)))\\ &+ \sum_l (\gamma_l z_l - \log(\Gamma(z_l)))\\ &+ \sum_j \beta_j y_j + \sum_k (\tau_k h_k - \log(\Gamma(h_k))\\ &- \log(\Gamma(M_k - h_k))) + \sum_{ik} W_{ik}x_i h_k\\ &+ \sum_{jk} U_{jk}y_j h_k + \sum_{lk} V_{lk}z_l h_k \Bigg\}. \end{aligned} \tag{17}$$

The marginal distribution over (x, Y, z) can be expressed as follows by marginalizing out the latent topics h in (17):

$$\begin{aligned} p(x, Y, z) \propto \exp\Bigg\{ &\sum_i (\alpha_i x_i - \log(\Gamma(x_i))) + \sum_j \beta_j y_j\\ &+ \sum_l (\gamma_l z_l - \log(\Gamma(z_l))) + \sum_k \Bigg(M_k \log \Bigg(1 + \exp\\ &\times \left(\sum_i W_{ik}x_i + \sum_j U_{jk}y_j + \sum_l V_{lk}z_l + \tau_k\right)\Bigg)\Bigg)\Bigg\}. \end{aligned} \tag{18}$$

The detailed derivation of (18) can be found in the Appendix. Likewise, in (17) and (18), the global partition function is not explicitly shown.

From the above probability distributions, we can see that the TWH model in this paper is an extension of the RAP model. It inherits rate adapting property that is determined by synopsis, actor, and comment features simultaneously. Thus, the learned latent topics will capture more information from movies to perform movie recommending task.

V. Learning and Inference

The parameters of the proposed TWH model $\{\alpha_i\}$, $\{\beta_j\}$, $\{\gamma_l\}$, $\{\tau_k\}$, $\{W_{ik}\}$, $\{U_{jk}\}$, and $\{V_{lk}\}$ can be learned by maximizing the likelihood of the movie data according to (18). Due to the complexity of the model, it is extremely difficult to obtain closed-form solution to the optimization problem. We have to perform stochastic gradient ascent on the log-likelihood of data in iteration. The learning rules can be derived from log-likelihood of (18) in the following way:

$$\delta\alpha_i = \langle x_i\rangle_{\tilde{p}} - \langle x_i\rangle_p \tag{19}$$

$$\delta\beta_j = \langle y_j\rangle_{\tilde{p}} - \langle y_j\rangle_p \tag{20}$$

$$\delta\gamma_l = \langle z_l\rangle_{\tilde{p}} - \langle z_l\rangle_p \tag{21}$$

$$\delta\tau_k = M_k(\langle\sigma(\bar{h}_k + \tau_k)\rangle_{\tilde{p}} - \langle\sigma(\bar{h}_k + \tau_k)\rangle_p) \tag{22}$$

$$\delta W_{ik} = M_k(\langle x_i\sigma(\bar{h}_k + \tau_k)\rangle_{\tilde{p}} - \langle x_i\sigma(\bar{h}_k + \tau_k)\rangle_p) \tag{23}$$

$$\delta U_{jk} = M_k(\langle y_j\sigma(\bar{h}_k + \tau_k)\rangle_{\tilde{p}} - \langle y_j\sigma(\bar{h}_k + \tau_k)\rangle_p) \tag{24}$$

$$\delta V_{lk} = M_k(\langle z_l\sigma(\bar{h}_k + \tau_k)\rangle_{\tilde{p}} - \langle z_l\sigma(\bar{h}_k + \tau_k)\rangle_p) \tag{25}$$

where $\bar{h}_k = \sum_i W_{ik}x_i + \sum_j U_{jk}y_j + \sum_l V_{lk}z_l$, $\langle\cdot\rangle_{\tilde{p}}$ represents the expectation under empirical distribution (i.e., data average), and $\langle\cdot\rangle_p$ represents the expectation under model distribution of the harmonium at the current values of the parameters. However, due to the presence of global partition function in the log-likelihood of (18), it is hard to directly estimate the model expectation $\langle\cdot\rangle_p$. There are many approximate inference methods to estimate this expectation such as CD learning [22], mean field (MF) approximation [23], and Langevin method [24]. CD learning algorithm is proposed to approximate exact gradient ascent search. MF is an alternative method that approximates the model distribution through a factorized form as a product of

Algorithm 1. CD learning procedure for the TWH model

```
Initialize the parameters: $\{\alpha_i\}$, $\{\beta_j\}$, $\{\gamma_l\}$, $\{\tau_k\}$, $\{W_{ik}\}$, $\{U_{jk}\}$, and $\{V_{lk}\}$.
repeat
    Sample the latent topics given the input data using Eq.(16);
    Resample the corresponding synopsis data-case given the sampled values of the latent topics using Eq.(13);
    Resample the corresponding actor data-case given the sampled values of the latent topics using Eq.(14);
    Resample the corresponding comment data-case given the sampled values of the latent topics using Eq.(15);
    Compute the data averages and sample averages in Eqs.(19)-(25);
    Update the parameters using the gradient ascent rules in Eqs.(19)-(25).
until Convergence.
Return: $\{\alpha_i\}$, $\{\beta_j\}$, $\{\gamma_l\}$, $\{\tau_k\}$, $\{W_{ik}\}$, $\{U_{jk}\}$, and $\{V_{lk}\}$.
```

marginal distributions over clusters of variables [21], [23]. With inheriting all the proposal moves of Langevin Monte Carlo method, the Langevin approach uses noisy steepest ascent to avoid local optima as well as taking advantage of the gradient information [24]. In this section, we only introduce the detailed CD learning algorithm for TWH training. We have also compared the performance of different algorithms for learning and inference in the experiment.

In each step of gradient ascent, CD starts from a separate Gibbs sampler defined by (13)–(16) at a data case, runs it for only a few steps and then uses these samples to approximate the model expectation $\langle\cdot\rangle_p$ together with computing the gradient through (19)–(25). It has been proved that the parameters through this learning process will converge to the maximum likelihood estimation [22]. The whole learning procedures are illustrated in Algorithm 1.

After learning and inference, all the movie data can be projected to low-dimensional latent representations. The TWH model is now ready to perform movie recommendation. Given a movie, we can simply compare it with other movies in a data set and return the most similar ones to construct the recommended list. The similarity between two movies can be calculated by the cosine distance between their latent representations learned by the TWH model.

VI. Experiment

A. Data Set and Experimental Setup

At present, there is no publicly available benchmark data set for evaluating the performance of content-based movie recommendation. In order to provide a real-life and demanding testing platform, we have established a large-scale data set compiled from the Internet movie database[1] (IMDb), the world's most popular and authoritative source for movie, TV, and celebrity content. Initially, we downloaded 259 825 movies from the IMDb. We then filtered out many unpopular movies, because they usually have neither synopsis description nor user comments. 5921 movies were left as the final experiment data set. The vocabulary size for synopsis is 14 448. For each movie, we only considered the top six actors in the actor ranking list, as the leading actors are more important in a movie. The total number of actors presented in these movies is 17 730, which is equal to the vocabulary size for actor features. To date, these movies include 1 256 762 user comments in total. The SOM grid size for mapping these user comments was set to 100 × 100. The initial learning rate for SOM was set to 0.3. The initial radius of the neighborhood function was set to half-length of the square grid. The number of total training iterations was set to five times of the total number of user comments. In this study, these parameter settings were observed to be a good choice. The corpus was split into a training set and a test set that is used for query. We randomly selected 592 test movies. The remaining 5329 movies were used for TWH training. All the experiments were performed on a PC with Intel(R) Xeon(R) CPU X3430 at 2.40 GHz and 8.00 GB memory. The feature extraction programs were written in Java programming language. All the recommending programs were tested on MATLAB 7.12.0 (R2011a).

B. Comparative Study on Recommending Performance

For comparison, we compared the proposed TWH model with the state-of-the-art algorithms including DWH [19], RAP [18], LDA [17], PLSI [16], LSI [15], and VSM [13]. Parameters involved in TWH were set as follows: the learning rate and the momentum term to speed up the convergence in TWH were set to 0.01 and 0.95, respectively; and the TWH based on 1000 learning iterations using gradient ascent on mini-batches of 100 random training samples per iteration. DWH and RAP used the same parameter settings for training. It is worth pointing out that if we view *tf* and *tcf* features in document data as synopsis and actor features in movie data, respectively, the DWH model can be straightforwardly applied to movie modeling. RAP, LDA, PLSI, LSI, and VSM used the synopsis features for recommending.

In IMDb, popular movies have around 12 relevant movies being recommended according to other users' preferences. We used this labeled information for performance evaluation. For each test movie, we built a random pool with size 6 (one positive + five negative) to measure the performance of different recommending models. To quantify the recommending results, we used four commonly used metrics [25]: normalized discounted cumulative gain at rank k (NDCG@k), mean reciprocal rank (MRR), success at rank k (S@k), and precision at rank k (P@k). The larger values these metrics deliver, the better the algorithm performs.

The numerical results with respects to NDCG@1, NDCG@6, and MRR are summarized in Table I. Figs. 3–5 visually illustrate the comparative results using different models. Here, the number of latent topics produced by TWH, DWH, RAP, LDA, PLSI, and LSI is 150. From Table I, it is clear that TWH performs the best in comparison to other methods. Specifically, TWH achieves at least around 4% improvement of NDCG@1 and 3% improvement of MRR over other approaches, respectively. RAP and DWH deliver similar results. It suggests that combining actor features into movie

[1] [Online]. Available: www.imdb.com

TABLE I
RECOMMENDING RESULTS OF COMPARED MODELS

Models	NDCG@1	NDCG@6	MRR
TWH	43.59	72.04	62.94
DWH	32.31	65.58	54.45
RAP	32.31	65.83	54.77
LDA	38.29	69.32	59.33
PLSI	40.00	70.04	60.29
LSI	35.56	68.53	58.22
VSM	36.24	68.51	58.23

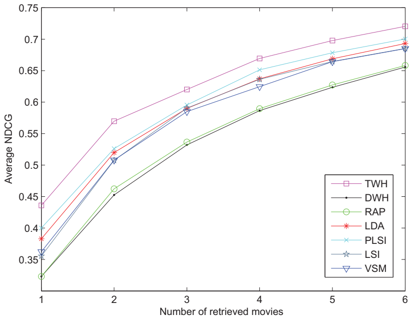


Fig. 3. Average NDCG against number of recommended movies.

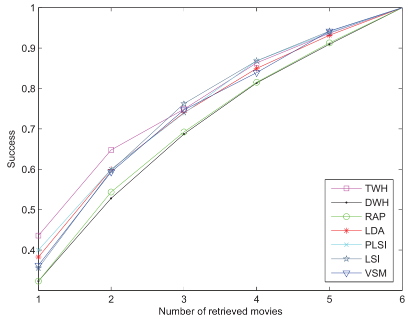


Fig. 4. Success against number of recommended movies.

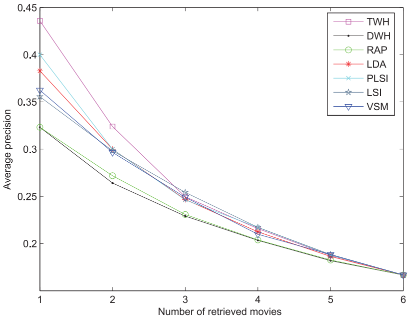


Fig. 5. Average precision against number of recommended movies.

representation does not bring a net performance gain. However, TWH produces significant performance improvement over DWH and RAP, which indicates that integrating user comment features into movie representation is beneficial to recommendation. From Fig. 3, TWH consistently performs better than other methods with different numbers of recommended movies. Fig. 4 shows that TWH delivers better results when the number of recommend movies is smaller than 3, but TWH, LDA, PLSI, LSI, and VSM demonstrate similar performance when the number of recommended movies increases. Similar results can be found in Fig. 5.

The number of latent topics may also affect the recommending results of our model. To show this effect, we provided the empirical results of dimensionality reduction related methods with different feature dimensions varying from 50 to 250 at an increment of 50. The results with respect to NDCG@1 are shown in Fig. 6. It is observed that the TWH model delivers relatively stable results with different numbers of latent topics. Likewise, TWH consistently performs better than other methods.

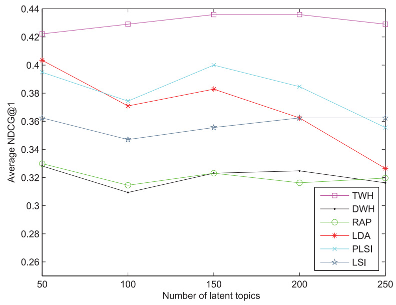


Fig. 6 Average NDCG@1 against number of latent topics.

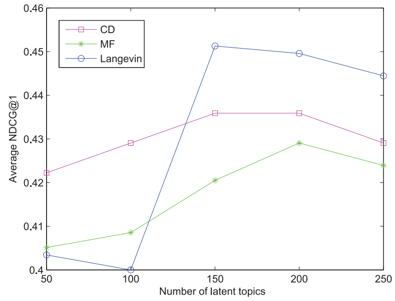


Fig. 7. Comparative results of different learning methods for TWH training.

C. Comparative Study on TWH Learning Algorithms

This section studies the effect of different learning approaches for TWH inference based on recommending results. Fig. 7 shows the NDCG@1 values of the TWH model implemented using different approximate inference methods with the number of latent topics from 50 to 250 at an increment of 50. From Fig. 7, the Langevin method performs the best when the number of latent topics is larger than around 130, but there is a sharp drop when the number of latent topics decreases to 100. In contrast, CD learning method shows relatively stable performance against the change of the number of latent topics. Thus, CD learning should be advocated in real applications.

VII. CONCLUSION

A new TWH model for movie data is proposed for the application of movie recommendation. This TWH model integrates multiple features into low-dimensional semantic space with latent topics for movie representation. In particular, user comments are newly incorporated into movie representation. By taking advantage of the power of SOM clustering, a new method is introduced to transform comment features for the TWH modeling. TWH extends the basic RAP model to three wings by using different conditional probability distributions.

It does not only include the properties of RAP but also contains capability to capture the semantic information hidden in actor and comment features. The experimental results corroborate that the proposed approach is effective for movie recommendation. Our future work will include integrating more features into movie representation and develop new semantic fusion strategies. We also plan to work on the inference algorithms to enhance the learning efficiency of harmonium models.

Appendix

Here, we show the derivation of the marginal distribution over (x, Y, z) in the TWH model. We defined the joint distribution over (x, Y, z, h) as shown in (17) (see Section IV-C). On the other hand, the latent topics $\{H_k\}$ follow the conditional binomial distribution depending on a weighted combination of synopsis feature x, binary actor feature Y, and comment feature z as follows:

$$\begin{aligned}
&p(h_k|x, Y, z)\\
&= \text{Binomial}\left(h_k|\sigma\left(\tau_k+\sum_i W_{ik}x_i+\sum_j U_{jk}y_j+\sum_l V_{lk}z_l\right), M_k\right)\\
&= \binom{M_k}{h_k}\cdot\left(\sigma\left(\tau_k+\sum_i W_{ik}x_i+\sum_j U_{jk}y_j+\sum_l V_{lk}z_l\right)\right)^{h_k}\\
&\cdot\left(\sigma\left(\tau_k+\sum_i W_{ik}x_i+\sum_j U_{jk}y_j+\sum_l V_{lk}z_l\right)\right)^{(M_k-h_k)}
\end{aligned} \tag{26}$$

where $\binom{M_k}{h_k} = \frac{\Gamma(M_k)}{\Gamma(h_k)\Gamma(M_k-h_k)} = \frac{M_k!}{h_k!(M_k-h_k)!}$. According to the definition of conditional probability distribution, we are ready to derive the marginal distribution over (x, Y, z) as shown in (27), at the bottom of the page which is exactly consistent with (18).

References

[1] J. de la Rosa *et al.*, "A negotiation-style recommender based on computational ecology in open negotiation environments," *IEEE Trans. Ind. Electron.*, vol. 58, no. 6, pp. 2073–2085, Jun. 2011.

[2] Y. Li *et al.*, "An efficient recommendation method for improving business process modeling," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 502–513, Feb. 2014.

[3] J. C. Ferreira, V. Monteiro, and J. L. Afonso, "Vehicle-to-anything application (V2Anything App) for electric vehicles," *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1927–1937, Aug. 2014.

[4] X. Luo *et al.*, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1273–1284, May 2014.

[5] M. Balabanovic and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Commun. Assoc. Comput. Mach.*, vol. 40, no. 3, pp. 66–72, 1997.

[6] X. N. Lam *et al.*, "Addressing cold-start problem in recommendation systems," in *Proc. 2nd Int. Conf. Ubiq. Inform. Manage. Commun.*, 2008, pp. 208–211.

[7] H. Mak, I. Koprinska, and J. Poon, "INTIMATE: A web-based movie recommender using text categorization," in *Proc. IEEE/WIC Int. Conf. Web Intell.*, 2003, pp. 602–605.

[8] R. Mukherjee, G. Jonsdottir, S. Sen, and P. Sarathi, "MOVIES2GO: An online voting based movie recommender system," in *Proc. 5th Int. Conf. Auton. Agents*, 2001, pp. 114–115.

[9] M. Szomszor *et al.*, "Folksonomies, the semantic web, and movie recommendation," in *Proc. 4th Eur. Semantic Web Conf. (ESWC) Workshop Bridging Gap Between Semantic Web Web 2.0*, 2007, pp. 71–85.

[10] J. Lees-Miller, F. Anderson, B. Hoehn, and R. Greiner, "Does wikipedia information help netflix predictions?" in *Proc. 7th Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2008, pp. 337–343.

[11] G. Lekakos and P. Caravelas, "A hybrid approach for movie recommendation," *Multimedia Tools Appl.*, vol. 36, nos. 1–2, pp. 55–70, 2008.

[12] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*. New York, NY, USA: Springer, 2011, pp. 73–105.

[13] G. Salton and M. McGill, Eds., *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1983.

[14] T. Kohonen, *Self-Organizing Maps*. Berlin, Germany: Springer-Verlag, 1997.

[15] S. Deerwester and S. Dumais, "Indexing by latent semantic analysis," *J. Am. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

[16] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, nos. 1–2, pp. 177–196, 2001.

[17] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[18] P. V. Gehler, A. D. Holub, and M. Welling, "The rate adapting Poisson model for information retrieval and object recognition," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 337–344.

[19] H. Zhang, T. W. S. Chow, and M. K. M. Rahman, "A new dual wing harmonium model for document retrieval," *Pattern Recog.*, vol. 42, no. 11, pp. 2950–2960, 2009.

$$\begin{aligned}
&p(x, Y, z)\\
&= \frac{p\,(x, Y, z, h)}{p\,(h|x, Y, z)}\\
&= \frac{p\,(x, Y, z, h)}{\prod_k p\,(h_k|x, Y, z)}\\
&\propto \frac{\exp\left\{\sum_i(\alpha_i x_i - \log(\Gamma(x_i))) + \sum_l(\gamma_l z_l - \log(\Gamma(z_l))) + \sum_j \beta_j y_j + \sum_k(\tau_k h_k - \log(\Gamma(h_k)) - \log(\Gamma(M_k - h_k))) + \sum_{ik} W_{ik}x_i h_k + \sum_{jk} U_{jk}y_j h_k + \sum_{lk} V_{lk}z_l h_k\right\}}{\binom{M_k}{h_k}\left(\sigma\left(\tau_k + \sum_i W_{ik}x_i + \sum_j U_{jk}y_j + \sum_l V_{lk}z_l\right)\right)^{h_k}\left(\sigma\left(\tau_k + \sum_i W_{ik}x_i + \sum_j U_{jk}y_j + \sum_l V_{lk}z_l\right)\right)^{(M_k-h_k)}}\\
&= \exp\left\{-\sum_k \log(\Gamma(M_k))\right\}\cdot\exp\left\{\sum_i(\alpha_i x_i - \log(\Gamma(x_i))) + \sum_l(\gamma_l z_l - \log(\Gamma(z_l))) + \sum_j \beta_j y_j + \sum_k\left(M_k \log\left(1 + \exp\left(\sum_i W_{ik}x_i + \sum_j U_{jk}y_j + \sum_{lk} V_{lk}z_l + \tau_k\right)\right)\right)\right\}\\
&\propto \exp\left\{\sum_i(\alpha_i x_i - \log(\Gamma(x_i))) + \sum_l(\gamma_l z_l - \log(\Gamma(z_l))) + \sum_j \beta_j y_j + \sum_k\left(M_k \log\left(1 + \exp\left(\sum_i W_{ik}x_i + \sum_j U_{jk}y_j + \sum_{lk} V_{lk}z_l + \tau_k\right)\right)\right)\right\}
\end{aligned} \tag{27}$$

[20] M. Welling, M. Rosen-Zvi, and G. Hinton, “Exponential family harmoniums with an application to information retrieval,” in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2004, vol. 17, pp. 1481–1488.

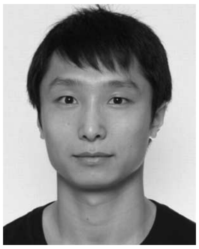
[21] E. Xing, R. Yan, and A. Hauptmann, “Mining associated text and images with dual-wing harmoniums,” in *Proc. Conf. Uncertainty Artif. Intell.*, 2005, pp. 633–641.

[22] M. Welling and G. E. Hinton, “A new learning algorithm for mean field Boltzmann machines,” in *Proc. Int. Conf. Artif. Neural Netw. (ICANN’02)*. Berlin, Germany: Springer-Verlag, 2002, pp. 351–357.

[23] E. Xing, M. Jordan, and S. Russell, “A generalized mean field algorithm for variational inference in exponential families,” in *Proc. Uncertainty Artif. Intell. (UAI’03)*, 2003, pp. 583–591.

[24] I. Murray and Z. Ghahramani, “Bayesian learning in undirected graphical models: Approximate MCMC algorithms,” in *Proc. 20th Annu. Conf. Uncertainty Artif. Intell.*, 2004, pp. 392–399.

[25] B. Sigurbjörnsson and R. Van Zwol, “Flickr tag recommendation based on collective knowledge,” in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 327–336.



Haijun Zhang (M’13) received the B.Eng. degree in civil engineering, and the Master’s degree in control theory and engineering from Northeastern University, Shenyang, China, in 2004 and 2007, respectively, and the Ph.D. degree in electronic engineering from the City University of Hong Kong, Hong Kong, in 2010.

He was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada, from 2010 to 2011. Since 2012, he has been with the Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China, where he is currently an Associate Professor of Computer Science. His research interests include multimedia data mining, machine learning, pattern recognition, evolutionary computing, and communication networks.

Yuzhu Ji received the B.S. degree in computer science from PLA Information Engineering University, Zhengzhou, China, in 2012, and the M.S. degree in computer engineering from the Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China, in 2015, where he is currently pursuing the Ph.D. degree in computer science.

His research interests include data mining, computer vision, image processing, and deep learning.

Jingxuan Li received the B.S. degree in software engineering from the Harbin Institute of Technology, Harbin, China, in 2013. She is currently pursuing the Master’s degree in computer engineering from the Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China.

Her research interests include data mining, natural language processing, and deep learning.

Yunming Ye (M’04) received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2003.

He is currently a Professor with the Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China. His research interests include data mining, text mining, and ensemble learning algorithms.