Combining Deep Learning and Topic Modeling for Review Understanding in Context-Aware Recommendation

Mingmin Jin[1] **Xin Luo**[1] **Huiling Zhu**[1,2] **Hankz Hankui Zhuo**[1,2*]
[1]School of Data and Computer Science, Sun Yat-Sen University
[2]Guangdong Key Laboratory of Big Data Analysis and Processing
Guangzhou, China
{jinmm[1],luox35[1],zhuhling6[2],zhuohank[2]}@{mail2[1],mail[2]}.sysu.edu.cn

Abstract

With the rise of e-commerce, people are accustomed to writing their reviews after receiving the goods. These comments are so important that a bad review can have a direct impact on others buying. Besides, the abundant information within user reviews is very useful for extracting user preferences and item properties. In this paper, we investigate the approach to effectively utilize review information for recommender systems. The proposed model is named LSTM-Topic matrix factorization (LTMF) which integrates both LSTM and Topic Modeling for review understanding. In the experiments on popular review dataset Amazon , our LTMF model outperforms previous proposed HFT model and ConvMF model in rating prediction. Furthermore, LTMF shows the better ability on making topic clustering than traditional topic model based method, which implies integrating the information from deep learning and topic modeling is a meaningful approach to make a better understanding of reviews.

1 Introduction

Recommender systems (RSs) are widely used in the field of electronic commerce to provide personalized recommendation services for customers. Most popular RSs are based on *Collaborative Filtering* (CF), which makes use of users' explicit ratings or implicit behaviour for recommendations (Koren, 2008). But CF models suffer from data sparsity, which is also called "*cold-start*" problem. Models perform poorly when there is few available data. To alleviate this problem, utilizing user reviews can be a good approach because user reviews can directly reflect users' preferences and items' properties and exactly correspond to the user latent factors and item latent factors in CF models.

*corresponding author

To understand user reviews, previous approaches are mainly based on *topic modeling*, a suite of algorithms that aim to discover the thematic information among documents (Blei, 2012). The simplest and commonly used topic model is *latent dirichlet allocation*(LDA). Recently, as deep learning shows great performance in computer vision (Krizhevsky et al., 2017) and NLP (Kim, 2014), some approaches combining deep learning with CF are proposed to capture latent context features from reviews.

However, we find there are some limitations in existing models. First, the LDA algorithm used in previous models like Hidden Factors as Topics (HFT) (McAuley and Leskovec, 2013) ignores contextual information. If a user writes "***I prefer apple than banana when choosing fruits***" in a review, we can clearly know the user's preference and recommend items including apple. But LDA ignores the structural information and considers the two words as the same since they both appear once in the sentence.

Compared with topic modeling, deep learning methods such as *Convolutional Neural Network* (CNN) and *Recurrent Neural Network* (RNN) are able to retain more context information. CNN uses sliding windows to capture local context and word order. RNN considers a sentence as a word sequence, and the former word information will be reserved and passed back, which gives RNN the ability to retain the whole sentence information.

But these still exist some problems. For CNN, the sizes of sliding windows are often small, which causes CNN model fails to link words in the sentence begin and end. Given the review ***"I prefer apple than google when choosing jobs"***, CNN can not notice the two words *'apple'* and *'jobs'* simultaneously if the windows size is small, so it will meet the ambiguity problem that the word *'apple'* means fruit or company. For RNN, al-

Proceedings of NAACL-HLT 2018, pages 1605–1614
New Orleans, Louisiana, June 1 - 6, 2018.

though it performs better than CNN on persisting former information, the information will still decreases with the length of sentences increasing. So when a review is long, the effect of RNN is limited.

Faced with these problems, we propose to integrate deep learning and topic modeling to extract more global context information and get a deeper understanding of user reviews. Deep learning methods can reserve context information, while topic modeling can provide word co-occurrence relation to make a supplement for information loss.

We use *Long Short-Term Memory* (LSTM) network for the deep learning part, because it is a special type of RNN which has better performance on gradient vanishing and long term dependence problems than vanilla RNN structure. We use LDA for the topic modeling part. Then the two parts are integrated into a matrix factorization framework. The final model is named **LSTM-Topic matrix factorization** (**LTMF**).

Furthermore, as the topic modeling part and deep learning part are connected in our model, the topic clustering results will be influenced by the deep learning information. In experiments, LTMF shows a better topic clustering ability than traditional LDA based HFT model. This gives us some inspiration on using the integrating methods into other tasks like sentiment classification.

In the remainder of the paper, we first review previous work related to our work. Then we address preliminaries and present our models in detail. After that we evaluate our approach by comparing our approach with state-of-the-art algorithms. Finally we conclude the paper with future work.

2 Related Work

There has been some earlier approaches to extract review information for RSs. Wang and Blei (2011) proposed *collaborative topic regression* (CTR) that combined topic modeling and collaborative filtering in a probabilistic model. McAuley and Leskovec (2013) developed a statistical model HFT using a transfer function to combine rating and review tightly. Ling et al. (2014) and Bao et al. (2014) proposed models similar to CTR and HFT with some structural differences.

Recently, several researchers begin to utilize deep learning in RSs. Wang et al. (2015) presented a Bayesian model *collaborative deep learning* (CDL) leveraging SDAE neural networks as a text feature learning component. Bansal et al. (2016) trained a *gated recurrent units* (GRUs) network to encode text sequences into latent vectors. Zhao et al. (2016) trained a deep CNN to discover the abstract representation of movie posters and still frames, and incorporated it into a neighborhood CF model. Kim et al. (2016) utilized CNN to retain contextual information in review, and developed a document context-aware recommendation model (ConvMF). The ConvMF model is a recently proposed model and is shown to outperform PMF and CDL, and we choose it as a baseline in our experiments. Zheng et al. (2017) proposed the *Deep Cooperative Neural Networks* (DeepCoNN) model which constructed two concurrent CNN to simultaneously model user and item reviews and then combined the features into Factorization Machine. Attention in neural networks has been popular in nearly years, Seo et al. (2017) proposed a model using CNN with dual attention for rating prediction. There are some similarity between the D-attn model with our LTMF model for we both want to extract more global information, where they use attention CNN model and we utilize the information from both topic modeling and deep learning. The D-attn model fail to work if there is not enough reviews, while our LTMF model use review information as a supplementary of rating. So it can still work effectively even there are few reviews.

Besides, Diao et al. (2014) proposed a method jointly modeling aspects, sentiments and ratings for movie recommendation. Hu et al. (2015) proposed MR3 model to combine ratings, social relations and reviews together for rating prediction. These hybrid models boost the performance than individual components, which also give us some inspiration on proposing the LTMF framework.

3 Preliminary

3.1 Notations

We use explicit ratings as the training and test data. Suppose there are M users

$$\mathcal{U} = \{u_1, u_2, ..., u_i, ..., u_M\}$$

and N items

$$\mathcal{V} = \{v_1, v_2, ..., v_j, ..., v_N\},$$

where each user and item is represented by a K-dimension latent vector, $u_i \in \mathbb{R}^K$ and $v_j \in \mathbb{R}^K$. The rating sparse matrix is denoted as $R \in \mathbb{R}^{M \times N}$, where r_{ij} is the rating of user u_i on item v_j. D is the review (document) corpus where d_{ij} is the review of user u_i on item v_j.

3.2 PMF: a standard matrix factorization model

Probabilistic Matrix Factorization (PMF) (Mnih and Salakhutdinov, 2008) is an effective recommendation model that uses matrix factorization (MF) technique to find the latent features of users and items from a probabilistic perspective. In PMF, the predicted rating $\hat{R}_{ij}$ is expressed as the inner product of user latent vector u_i and item latent vector v_j : $\hat{R}_{ij} = u_i^T v_j$. To get latent vectors, PMF minimises the following loss function:

$$\mathcal{L} = \sum_i^M \sum_j^N I_{ij}(R_{ij} - u_i^T v_j)^2 + \lambda_u \sum_i^M \| u_i \|_F^2 + \lambda_v \sum_j^N \| v_j \|_F^2, \quad (1)$$

where R_{ij} is the observed rating. The first part of Eq.(1) is the sum-of-squared-error between predicted and observed ratings and the second part is quadratic regularization terms to avoid overfitting. λ_u and λ_v are corresponding regularization parameters. I_{ij} is the indicator function which equals 1 if i-th user rated j-th item, and equals 0 otherwise.

3.3 HFT: understand reviews through topic modeling

Hidden Factors as Topics (HFT) (McAuley and Leskovec, 2013) provides an effective approach to integrates topic modeling into traditional CF models. It utilizes LDA, the simplest topic model which assumes there are k topics $\mathcal{T} = \{t_1, t_2, ..., t_k\}$ in document corpus D. Each document $d \in D$ has a topic distribution θ_d over $\mathcal{T}$ and each topic has a word distribution ϕ over a fixed vocabulary. To connect the document-topic distribution θ and item factors v, HFT proposes a transformation function:

$$\theta_{j,k} = \frac{\exp(\kappa v_{j,k})}{\sum_k \exp(\kappa v_{j,k})}, \quad (2)$$

where $v_{j,k}$ is the k-th latent factor in item vector v_j and $\theta_{j,k}$ is the k-th topic probability in item document-topic distribution θ_j, κ is the parameter controlling the "peakiness" of the transformation. Besides, HFT introduces an additional variable ψ to ensure the word distribution ϕ_k is a stochastic vector which satisfies $\sum_w \phi_{k,w} = 1$, the relation is denoted as follows:

$$\phi_{k,w} = \frac{\exp(\psi_{k,w})}{\sum_w \exp(\psi_{k,w})} \quad s.t. \quad \sum_w \phi_{k,w} = 1. \quad (3)$$

The final loss function is :

$$\mathcal{L} = \sum_i^M \sum_j^N I_{ij}(R_{ij} - \hat{R}_{ij})^2 - \lambda_t \sum_d^N \sum_{n \in N_d} \log \theta_{d,z_{d,n}} \phi_{z_{d,n},w_{d,n}}, \quad (4)$$

where $\hat{R}_{ij}$ is predicted ratings, θ and ϕ are the topic and word distribution respectively, $w_{d,n}$ is the n-th word in document d and $z_{d,n}$ is the word's corresponding topic, λ_t is a regularization parameters.

4 The LTMF Model

We propose the LSTM-Topic matrix factorization (LTMF) model, which integrates LSTM and topic modeling for recommendation. The model utilizes both rating and review information. For the rating part, we use probabilistic matrix factorization to extract rating latent vectors. For the review part, we use LDA (following the way of HFT) to extract topic latent vectors and adopt an LSTM architecture to generate document latent vectors . Then we combine the three vectors into a unified model. The overview of LTMF model is shown in Figure 1.

4.1 Parameter Relation

The left of Figure 1 is the parameters relations in LTMF model, which can be divided into three parts: $\Theta = \{\mathcal{U}, \mathcal{V}\}$ is the parameters associated with rating MF, $\Phi = \{\theta, \phi\}$ is the parameters associated with topic model, $\Omega = \{W, l\}$ is the parameters associated with LSTM. The shaded nodes are data (R:rating, D: reviews) where the others are parameters. Single connection lines represent there are constraint relationship between the two nodes. Double connections (e.g. $\mathcal{V}$ and θ) mean the relationship is bidirectional so they can affect each other's results.

4.2 LSTM Architecture

The right of Figure 1 is the LSTM architecture used in our models. For the j-th item, we concatenate all of its reviews as one document se-

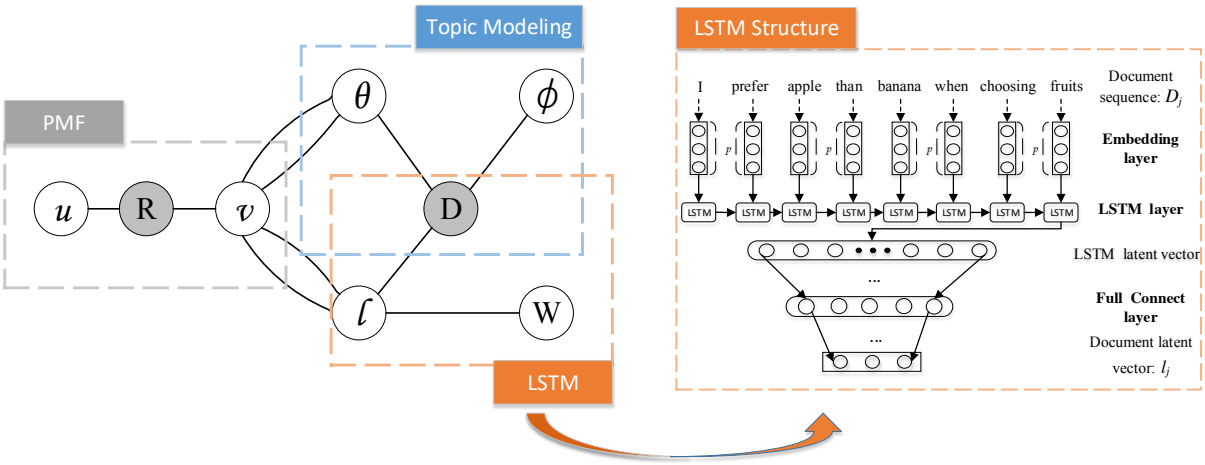

Figure 1: The overview of LTMF model. Left:Parameters relations of LTMF. Right:the detailed LSTM architecture.

quence D_j. Every word in the document sequence $D_j = (w_1, w_2, ..., w_{nj})$ will firstly be embedded into a p dimension vector. Next, word vectors are sent into LSTM network according to the word order in D_j and produces a latent vector. Finally, the latent vector is sent to a full connect layer whose output is the document latent vector l_j. The above process can be written as:

$$l_j = LSTM(D_j, W), \tag{5}$$

where D_j is the input document sequence, W represents weights and bias variables in LSTM network.

4.3 Probabilistic Prior

Gaussian distribution is the basic prior hypothesis in our model. We place zero-mean spherical Gaussian priors on user latent features u, LSTM weights W and observed ratings R. For item vector v, we place the Gaussian prior on its difference with LSTM output l_j :

$$p(v|l_j, \sigma_v^2) = \prod_{j=1}^{N} \mathcal{N}(v_j - l_j|0, \sigma_v^2 I)$$
$$= \prod_{j=1}^{N} \mathcal{N}(v_j|l_j, \sigma_v^2 I).$$

The function is important for connecting ratings and reviews. Although document vector l_j is closed to item feature vector v_j for they both reflect item's properties. There still exists some discrepancies. For example, when writing reviews, users usually write more about appearance and only briefly mention price. So in review based document vector l_j, the weight of "appearance" will be larger than rating based latent vector v_j. To preserve the discrepancy between v_j and l_j , we import the Gaussian noise vector σ_v as the offset.

4.4 Objective Function

Finally, we maximize the log-posterior of the three parts and get the objective function as follows:

$$\begin{aligned}\mathcal{L} = &\sum_{i}^{M}\sum_{j}^{N} I_{ij}(R_{ij} - u_i^T v_j)^2 \\ &- \lambda_t \sum_{d}^{N}\sum_{n\in N_d} \log \theta_{z_{d,n}} \phi_{z_{d,n},w_{d,n}} \\ &+ \lambda_u \sum_{i}^{M} \| u_i \|_F^2 + \lambda_v \sum_{j}^{N} \| v_j - l_j \|_F^2 \\ &+ \lambda_W \sum_{k}^{N_k} \| W_k \|_F^2,\end{aligned} \tag{6}$$

where N_k is the number of weighs in LSTM network, $\lambda_u, \lambda_v, \lambda_W$ are regularization parameters. z is the topic assignment for each word, λ_t is the regularization parameters to control the proportion of topic part.

The objective function of LTMF can be considered as an extended PMF model where the information from topic modeling and LSTM is included as regular terms. In the next section, we will explain how LTMF leverages the information from topic modeling and LSTM, and why LTMF can combine the information of the two parts.

4.5 The Effectiveness of LTMF

As shown in Figure 1, item vectors $\mathcal{V}$ connect with both topic part and LSTM part, which means the information from the two part will both affect the

result of item vectors. If we take partial derivative of Eq.(6) with respect to v_j, the constraint relationship can be clearer:

$$\frac{\partial \mathcal{L}}{\partial v_j} = \sum_{i=1}^{M} 2I_{ij}(R_{ij} - u_i^T v_j)u_i + 2\lambda_v(v_j - l_j) - \lambda_t \kappa \sum_{k=1}^{K}(n_{j,k} - N_j \frac{\exp(\kappa v_{j,k})}{\sum \exp(\kappa v_{j,k})}), \quad (7)$$

In Eq.(7), the optimization direction of v_j is subject to two regular terms. The former one is controlled by LSTM vector l_j. The latter one is controlled by topic parameters (κ, $n_{j,k}$, N_j). Hence, we can leverages the information from both LSTM and topic modeling for recommendation.

Besides, note the double connections between item vector $\mathcal{V}$ and topic distribution θ in Figure 1. They mean the information from topic modeling can affect the result of $\mathcal{V}$, while the change in $\mathcal{V}$ can also be passed to topic part and affect the review understanding result of topic modeling by Eq.2. For $\mathcal{V}$ and LSTM vector l, the analysis is the same. Indeed, item vectors $\mathcal{V}$ plays the role of transporter to connect LSTM part and topic modeling part. This is why LTMF can combine the information of topic modeling and LSTM to make a deeper understanding of user reviews.

Furthermore, LTMF provides an effective framework to integrate topic model with deep learning networks for recommendation. In experiments, we replace the LSTM part with CNN to make a comparison model. Experiments show both models boost the rating prediction accuracy.

4.6 Optimization

Our objective is to search:

$$\arg\min_{\Theta,\Phi,z,\kappa,\Omega} \mathcal{L}(\Theta, \Phi, z, \kappa, \Omega). \quad (8)$$

Recall that Θ is the parameters associated with ratings MF, Φ is the parameters associated with topic modeling, z is the topic assignment for each word, κ is the peakiness parameter to control the transformation between item vector v and topic distribution θ, Ω is the parameters associated with LSTM.

For v_j is coupled with the parameters of topic modeling and LSTM vector, we cannot optimize these parameters independently. We adopt a procedure that alternates between two steps. In each step we fix some parameters and optimize the others. The optimization process is shown below:

1. solve the objective by fixing z^t and Ω^t:
$$\arg\min_{\Theta,\Phi,\kappa} \mathcal{L}(\Theta, \Phi, z^t, \kappa, \Omega^t)$$
to update Θ^{t+1}, Φ^{t+1}, κ^{t+1}.

2. (a) update Ω^{t+1} with fixing ${v_j}^{t+1}$ and document sequence D_j.

   (b) sample $z_{d,j}^{t+1}$ with probability
$$p(z_{d,j}^{t+1} = k) = \phi_{k,w_{d,j}}^{t+1}.$$

In the step 1, we fix z and Ω to update remaining terms Θ, Φ, κ by L-BFGS algorithm. In the step 2, we fix Θ, Φ and κ to update LSTM parameters Ω and topic model parameters z. Since LSTM part and topic part are independent when item vectors $\mathcal{V}$ are certain, we can update the two term respectively. In step 2(a), we update Ω by back propagation algorithm. With fixing the other parameters, the objective function of W can be seen as a weighted squared error function ($\| v_j - l_j \|_F^2$) with L_2 regularized terms ($\| W \|_F^2$), which means we can use D_j as the input and v_j is the label to run the back propagation process. In step 2(b), we iterates through all documents and each word within to update $z_{d,j}$ via Gibbs Sampling. The reason why we do not divide the process into three steps is that the step 2(a) and 2(b) are independent with step 1 finished, which means we can parallelize the two steps.

Finally, we repeat these two steps until convergence. In practice, we run the step 1 with 5 gradient iterations using LBFGS, then we iterate the LSTM part 5 times. At the same time, we update the topic model part once. The whole process is called a cycle, and it usually takes 30 cycles to reach a local optimum.

In addition to the gradient of v_j, the gradients of other parameters used in step 1 are listed as follows:

$$\frac{\partial \mathcal{L}}{\partial u_i} = \sum_{j=1}^{N} 2I_{ij}(R_{ij} - u_i^T v_j)v_j + 2\lambda_u u_i. \quad (9)$$

$$\frac{\partial \mathcal{L}}{\partial \psi} = -\lambda_t \sum_{w=1}^{N_w} \sum_{k=1}^{K} \left(n_{k,w} - N_k \frac{\exp(\psi_{k,w})}{z_w} \right) \quad (10)$$

$$\frac{\partial \mathcal{L}}{\partial \kappa} = -\lambda_t \sum_{j=1}^{N} \sum_{k=1}^{K} v_{j,k} \left(n_{jk} - N_j \frac{\exp(\kappa v_{j,k})}{z_j} \right). \quad (11)$$

where ψ is used to determine word distribution ϕ by Eq.(3); $n_{k,w}$ is the number of times that word

Dataset	users	items	ratings	av. words per item	density
Amazon Instant Video (AIV)	29753	15147	135167	86.69	0.0030%
Apps for Android (AFA)	240931	51599	1322839	65.80	0.0106%
Baby (BB)	71812	42515	379969	81.83	0.0124%
Musical Instruments (MI)	29005	48751	150526	86.44	0.0106%
Office Product (OP)	59844	60628	286521	82.41	0.0079%
Pet Supplies (PS)	93270	70063	477859	78.86	0.0073%
Grocery Gourmet Food (GGF)	86389	108448	508676	76.97	0.0054%
Video Games (VG)	84257	39422	476546	92.55	0.0143%
Patio Lawn and Garden (PLG)	54167	57826	242944	82.22	0.0078%
Digital Music (DM)	56812	156496	351769	38.79	0.0040%

Table 1: Statistics of datasets.

w occurs in topic k; N_w is the word vocabulary size of the document corpus; N_k is the number of words in topic k; $n_{j,k}$ is the number of times when topic k occurs in the document of item j; N_j is the total number of words in document j; z_w and z_j are the corresponding normalizers:

$$z_w = \sum_{k=1}^{K} \exp(\psi_{k,w}), \quad z_j = \sum_{k=1}^{K} \exp(\kappa v_{j,k}).$$

5 Experiment

5.1 Datasets

We use the real-world Amazon dataset[1] (collected by McAuley et al. (2015)) for our experiments. For the original dataset is too large, we choose 10 sub datasets in experiments. To increase data density, we remove users which have less than 3 ratings. For raw review texts, we adopt the same preprocessing methods as ConvMF[2]: set the maximum length of a item document to 300; remove common stop words and document specific words which have document frequency higher than 0.5; choose top 8000 distinct words as the vocabulary; remove all non-vocabulary words to construct input document sequences. After preprocessing, the statistics of datasets are listed in Table 1, where the abbreviations of datasets are shown in parentheses.

5.2 Evaluation Procedure

5.2.1 Baseline

The baselines used in our experiments are listed as follows:

- PMF: Probabilistic Matrix Factorization (PMF) (Mnih and Salakhutdinov, 2008) is a standard matrix factorization model for RSs. It only uses rating information.

- HFT: This is a state-of-art method that combines reviews with ratings (McAuley and Leskovec, 2013). It utilizes LDA to capture unstructured textual information in reviews.

- ConvMF: Convolutional Matrix Factorization (ConvMF) (Kim et al., 2016) is a recently proposed recommendation model. It utilizes CNN to capture contextual information of item reviews.

- LMF: LSTM Matrix Factorization (LMF) is a submodel of LTMF without the topic part. We can compare it with ConvMF to show the effectiveness of LSTM than CNN on review understanding.

- CTMF: We modify the LTMF model by replacing the LSTM part with CNN (following the structure of ConvMF) and construct the comparison model CNN-Topic Matrix Factorization (CTMF). CTMF can be used to evaluate the effectiveness of combining deep learning and topic modeling.

In experiments, we randomly split one dataset into training set, test set, validation set under proportions of 80%, 10%, 10%, where each user and item appears at least once in the training set. We use Mean Square Error (MSE) as metric to evaluate various models.

[1] http://jmcauley.ucsd.edu/data/amazon/
[2] http://dm.postech.ac.kr/ cartopy/ConvMF/

Dataset	(a) PMF	(b) HFT	(c) ConvMF	(d) CTMF	(e) LMF	(f) LTMF
AIV	1.436 (0.02)	1.368 (0.02)	1.388 (0.02)	1.350 (0.03)	1.321 (0.02)	**1.309** (0.02)
AFA	1.673 (0.01)	1.649 (0.01)	1.651 (0.01)	1.648 (0.01)	1.635 (0.01)	**1.629** (0.01)
BB	1.643 (0.01)	1.577 (0.01)	1.556 (0.02)	1.531 (0.01)	1.513 (0.02)	**1.499** (0.01)
MI	1.555 (0.03)	1.423 (0.02)	1.399 (0.02)	1.367 (0.02)	1.317 (0.02)	**1.302** (0.02)
OP	1.622 (0.02)	1.547 (0.02)	1.501 (0.02)	1.466 (0.02)	1.432 (0.02)	**1.420** (0.02)
PS	1.796 (0.01)	1.736 (0.01)	1.698 (0.02)	1.680 (0.02)	1.646 (0.02)	**1.626** (0.01)
GGF	1.585 (0.01)	1.539 (0.01)	1.478 (0.01)	1.446 (0.02)	1.393 (0.01)	**1.386** (0.01)
VG	1.510 (0.02)	1.468 (0.01)	1.463 (0.01)	1.448 (0.01)	1.423 (0.01)	**1.409** (0.01)
PLG	1.854 (0.02)	1.779 (0.02)	1.710 (0.02)	1.678 (0.02)	1.628 (0.02)	**1.608** (0.02)
DM	1.197 (0.01)	1.171 (0.01)	1.032 (0.01)	0.990 (0.01)	0.968 (0.01)	**0.965** (0.01)

Table 2: MSE results of various models (K=5). The best results are highlighted in bold. The standard deviations of MSE results are shown in parenthesis.

5.2.2 Implementation Details

For all models, we set the dimension of user and item latent vectors $K = 5$, and initialize the vectors randomly between 0 and 1. Topic number and the dimension of document latent vector l are also set to 5. For methods using deep learning, we initialized word latent vectors randomly with the embedding dimension $p = 200$. The optimization algorithm used in back propagation is *rmsprop* and the activation function used in fully connected layer is *tanh* . In LSTM network, we set the output dimension to 128 and dropout rate 0.2. For CTMF, we adopt the same setting as ConvMF where the sliding window sizes is $\{3, 4, 5\}$ and the shared weights per window size is 100.

Hyper parameters are set as follows. For PMF, $\lambda_u = \lambda_v = 0.1$. For HFT, we select $\lambda_t \in \{1, 5\}$ which gives better result in each experiment. For LMF and ConvMF, we set $\lambda_u = 0.1$ and $\lambda_v = 5$. For LTMF and CTMF, we select $\lambda_t \in \{0.05, 0.1, 0.5\}$ which gives the lowest validation set error.

5.3 Quantitative analysis of rating prediction

We evaluate these models and report the lowest test set error on each dataset. The MSE results are shown in Table 2 where the best result of each dataset is highlighted in bold and the standard deviations of corresponding MSE are recorded in parenthesis.

We can see that the **LTMF model consistently outperform these baselines on all datasets** . This clearly confirms the effectiveness of our proposed method. To make a more intuitive comparison, the improvement histograms of these models are shown in Figure 2.

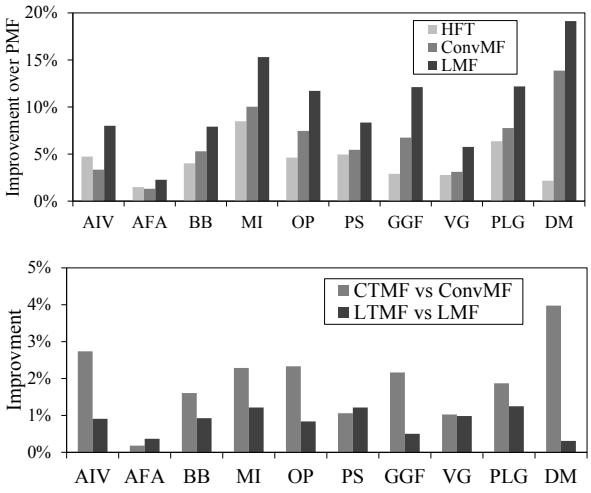

Figure 2: Above: Improvements of HFT, ConvMF and LMF, compared with PMF on different datasets. Below: Improvements of CTMF and LTMF, compared with ConvMF and LMF respectively.

The figure above are the improvements of HFT, ConvMF and LMF compared with PMF on different datasets, where PMF only uses rating information and the other three use both rating and review information with different approaches. We observe that all three methods make significant improvements over PMF, which indicates review information is helpful to model user and item features as well as improve recommendation results. Compared with HFT, LMF makes over 3% improvement on 9 out of the 10 datasets. ConvMF performs better than HFT while LMF still obtains over 3% improvement than ConvMF on 7 datasets. The differences between HFT, ConvMF and LMF can be attributed to their individual methods for re-

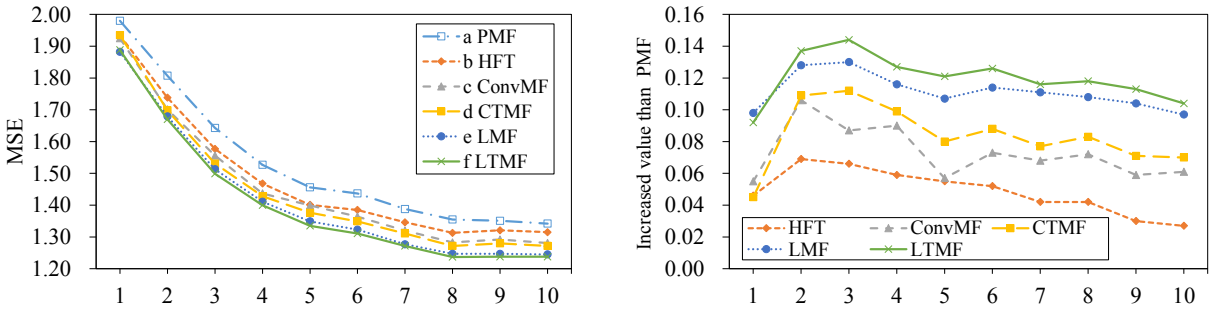

Figure 3: Results for recommendation within limited ratings and reviews. Left: the MSE values of all models. Right: the incerase compared with PMF.

views understanding. As mentioned in Section 1, Topic Modeling based HFT only considers the co-existence of words in texts and ignores structural context information. CNN based ConvMF lacks the ability to capture global context information due to the size limitation of sliding windows. This is exactly what LSTM possesses and why LSTM based LMF model outperforms ConvMF.

The figure below is the comparison of two integrated models (LTMF and CTMF) that import topic information with two original models that only use deep learning (LMF and ConvMF). We can see that both integrated models outperform the original models, which confirms our conjecture that **recommendation results can be improved by combining structural and unstructured information**. For CTMF model, it makes over 2% improvement on 5 out of 10 datasets compared with ConvMF. As to LTMF model, it achieves nearly 1% improvements that LMF on 7 out of 10 datasets.

The reason why LTMF gains less promotion can be explained from two sides. Numerically, for the comparison model LMF is already a strong baseline proposed by ourselves, it's more difficult to make a significant improvement. Theoretically, since LSTM can persist enough global information when the input sentence is relatively short, the supplements of topic information in LTMF are not so remarkable. As an illustration, we can compare the results on datasets "DM" and "VG". For the dataset "DM", as shown in Table 1, it has the fewest words per item (38.79) and the improvement of LTMF is minimum. But for the dataset "VG", it has the most words per item (92.55). The global context information obtained by LSTM will still decrease with such long sentences, and the topic information can make an effective supplement. So the improvement of LTMF on "VG" is greater and comparable with CTMF.

5.4 Recommendation with different data sparsity

Rating data and review data are always sparse in RSs. To compare these models on making recommendation in different data sparsity, especially for new users who only have limited ratings, we choose the dataset "Baby" and refilter it to make sure every user has at least N ratings (N varies from 1 to 10). A greater N means the user has rated more items, so the data sparsity problem is weaker. We test all models on the 10 subsets of "Baby" with the same dataset split ratio and text preprocessing. The final results are shown in Figure 3, where the left one is the MSE values of all models, and the right one is the increase of the other models compared with PMF.

We can observe that all models gain better recommendation accuracy with the increment of user rating number N. In other words, user and item latent features can be better extracted with more useful information. When N is small, especially when $N = \{2, 3\}$, the models which utilize both review and rating information achieve biggest improvements over PMF. It suggests that **review information can provide effective supplement when rating data is scarce**. With the increase of N, the improvements of all review used models become smaller. This is because models can extract more features from gradually dense ratings data, and the effectiveness of review data begins to decrease. Same as the previous experiment, our LTMF model achieve the best results in the comparison with other models.

5.5 Qualitative Analysis

In HFT, the result of topic words only depends on the information from Topic Modeling. But in our

Office Product (OP)				
topic1	topic2	topic3	topic4	topic5
envelope	markers	pins	wallet	planner
erasers	**compatible**	scale	notebooks	keyboard
needs	lead	***huge***	window	tab
numbers	mail	credit	notebook	remove
letters	***nice***	**document**	cardboard	stickers
christmas	camera	***attach***	plug	clips

Table 3: Top topic words discovered by HFT

Office Product (OP)				
topic1	topic2	topic3	topic4	topic5
bands	bags	scale	wallet	folder
drum	camera	**document**	clock	folders
remote	cabinet	magnets	coins	binder
chalk	**compatible**	monitors	notebooks	stickers
presentation	tray	pins	shredder	remove
buttons	party	fax	bookmark	head

Table 4: Top topic words discovered by LTMF

proposed LTMF framework, the information extracted by LSTM and Topic Modeling will both affect the final word clustering results. So, we can compare the topic words discovered by HFT and LTMF to evaluate whether combing LSTM and Topic Modeling is able to make a better understanding of user reviews.

We choose the dataset "Office Product" (OP) and show the top topic words of HFT and LTMF in Table 3 and Table 4. As we can see, there are many words existed in both tables (e.g. "wallet", "notebooks", "document"). These words are closely related to the category of dataset "Office Product", which implies both models can get a good interpretation of user reviews.

However, when we carefully compare the two tables, there exists some differences. In Table 3, there are some adjectives and verbs which have little help for topic clustering (e.g. "nice", "huge", "attach"), but they still get large weights and appear in the front of topic words list. Obviously, HFT misinterprets these words for they usually appear together with the real topic words. In Table 4, we are not able to find them in top words list, because extra information from LSTM makes a timely supplement. Besides, similar situations also occur on words "document" and "compatible". The word "document" is an apparent topic word, so LTMF gives it a larger weight in topic words list. For the word "compatible", as an adjectives, it can provide less topic information than nouns, so LTMF decreases its weight and put "camera" in the second place. From the above analysis we can see LTMF shows the better topic clustering ability than HFT.

6 Conclusion and Future Work

In this paper, we investigate the approach to effectively utilize review information for RSs. We propose the LTMF model which integrates both LSTM and Topic modeling in context aware recommendation. In the experiments, our LTMF model outperforms HFT and ConvMF in rating prediction especially when the data is sparse. Furthermore, LTMF shows better ability on making topic clustering than traditional topic model based method HFT, which implies integrating the information from deep learning and topic modeling is a meaningful approach to make a better understanding of reviews. In the future, we plan to evaluate more complex networks for recommendation tasks under the framework proposed by LTMF. Besides, we are interested to apply the method of combing topic model and deep learning into some traditional NLP tasks.

Acknowledgments

We thank the National Key Research and Development Program of China (2016YFB0201900), National Natural Science Foundation of China (U1611262), Guangdong Natural Science Funds for Distinguished Young Scholar (2017A030306028), Pearl River Science and Technology New Star of Guangzhou, and Guangdong Province Key Laboratory of Big Data Analysis and Processing for the support of this research.

References

Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the gru: Multi-task learning for deep text recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, New York, NY, USA, RecSys '16, pages 107–114. https://doi.org/10.1145/2959100.2959180.

Yang Bao, Hui Fang, and Jie Zhang. 2014. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, AAAI'14, pages 2–8. http://dl.acm.org/citation.cfm?id=2893873.2893874.

David M. Blei. 2012. Probabilistic topic models. *Commun. ACM* 55(4):77–84. https://doi.org/10.1145/2133806.2133826.

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '14, pages 193–202. https://doi.org/10.1145/2623330.2623758.

Guang-Neng Hu, Xin-Yu Dai, Yunya Song, Shu-Jian Huang, and Jia-Jun Chen. 2015. A synthetic approach for recommendation: Combining ratings, social relations, and reviews. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, IJCAI'15, pages 1756–1762. http://dl.acm.org/citation.cfm?id=2832415.2832493.

Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, New York, NY, USA, RecSys '16, pages 233–240. https://doi.org/10.1145/2959100.2959165.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pages 1746–1751. https://doi.org/10.3115/v1/D14-1181.

Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '08, pages 426–434. https://doi.org/10.1145/1401890.1401944.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60(6):84–90. https://doi.org/10.1145/3065386.

Guang Ling, Michael R. Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM Conference on Recommender Systems*. ACM, New York, NY, USA, RecSys '14, pages 105–112. https://doi.org/10.1145/2645710.2645728.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, New York, NY, USA, RecSys '13, pages 165–172. https://doi.org/10.1145/2507157.2507163.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, SIGIR '15, pages 43–52. https://doi.org/10.1145/2766462.2767755.

Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, Curran Associates, Inc., pages 1257–1264.

Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, New York, NY, USA, RecSys '17, pages 297–305. https://doi.org/10.1145/3109859.3109890.

Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '11, pages 448–456. https://doi.org/10.1145/2020408.2020480.

Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '15, pages 1235–1244. https://doi.org/10.1145/2783258.2783273.

Lili Zhao, Zhongqi Lu, Sinno Jialin Plan, and Qiang Yang. 2016. Matrix factorization+ for movie recommendation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, IJCAI'16, pages 3945–3951. http://dl.acm.org/citation.cfm?id=3061053.3061171.

Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, New York, NY, USA, WSDM '17, pages 425–434. https://doi.org/10.1145/3018661.3018665.