JOINT TRAINING OF RATINGS AND REVIEWS WITH RECURRENT RECOMMENDER NETWORKS

Chao-Yuan Wu
University of Texas at Austin
cywu@cs.utexas.edu

Amr Ahmed & Alex Beutel*
Google Research
{amra,alexbeutel}@google.com

Alexander J. Smola
Carnegie Mellon University
alex@smola.org

ABSTRACT

Accurate modeling of ratings and text reviews is at the core of successful recommender systems. In this paper, we provide a neural network model that combines ratings, reviews, and temporal patterns to learn highly accurate recommendations. We co-train for prediction on both numerical ratings and natural language reviews, as well as using a recurrent architecture to capture the dynamic components of users' and items' states. We demonstrate that incorporating text reviews and temporal dynamic gives state-of-the-art results over the IMDb dataset.

1 INTRODUCTION

Designing highly accurate recommender systems has been the focus of research in many communities and at the center of many products for the past decade. The core goal is to predict which items a given user will like or dislike, typically based on a database of previous ratings and reviews (Salakhutdinov & Mnih, 2008; Beutel et al., 2015; McAuley & Leskovec, 2013; Diao et al., 2014).

This previous research has been remarkably successful, but has two significant limitations that we discuss and address in this paper. First, prediction accuracy has rarely been measured by the ability of a model to predict *future* ratings. Rather, recommendation accuracy has been derived from a random split of the ratings data, which undermines our understanding of the models' usefulness in practice. More recently, Recurrent Recommender Networks (RRN) use a recurrent neural network to capture changes in both user preferences and item perceptions, and *extrapolate* future ratings in an autoregressive way (Wu et al., 2017). However, temporal patterns in reviews are largely unexplored.

Second, models of reviews in recommender system fall significantly behind the state-of-the-art in natural language processing. The bag-of-words model used in previous research improves over not using text, but is limited in the degree to which it can understand the review. In fact, the drawback of an underfitting model is especially salient in the case of reviews, because they are much more diverse and unstructured than regular documents.

Here, we combine these powerful neural-based language models with Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) recurrent neural networks (RNN) to learn both accurate recommendations and accurate reviews. Our main contributions are as follows:

- **Joint generative model:** We propose a novel joint model of ratings and reviews via interacting recurrent networks (particularly LSTM).
- **Nonlinear nonparametric review model:** By learning a function of user and movie state dynamics, we can capture the evolution of reviews (as well as ratings) over time.
- **Experiments** show that by jointly modeling ratings and reviews along with temporal patterns, our model achieves state-of-the-art results on IMDb dataset in terms of forward prediction, i.e. in the realistic scenario where we use only ratings strictly prior to prediction time to predict future ratings.

*A majority of this work was done while the author was at Carnegie Mellon University.

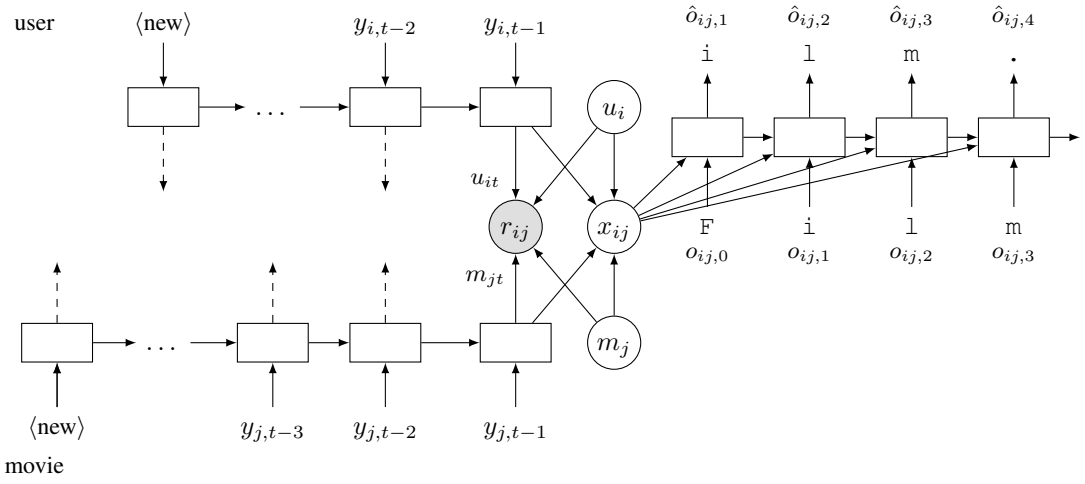

Figure 1: Joint Review-Rating Recurrent Recommender Networks: We use recurrent networks to capture the temporal evolution of user and movies states. These states are directly used to predict ratings and within an LSTM to model review text.

2 MODEL

Figure 1 shows a depiction of our model: Joint Review-Rating Recurrent Recommender Network. Here we use two LSTM RNNs that take user/movie history as input to capture the temporal dynamics in both user and movie states. Given stationary and dynamic states of user i and movie j, we define generator functions that emit both rating $r_{ij}|t$ and reviews $o_{ij}|t$ at time step t. Formally,

$$\begin{aligned} r_{ij}|t = f(u_i, m_j, u_{it}, m_{jt}) \quad &\text{and} \quad o_{ij}|t = \psi(u_i, m_j, u_{it}, m_{jt}) \\ u_{i,t+1} = g(u_{it}, \{r_{ij}|t\}) \quad &\text{and} \quad m_{j,t+1} = h(m_{jt}, \{r_{ij}|t\}), \end{aligned}$$

where u_i and m_j denote stationary states, and u_{it} and m_{it} denote the dynamic state at t. Note that here essentially we learn the *functions* that find the states instead of learning the states directly.

Dynamic User and Movie State The key idea is to use user/movie rating history as inputs to update the states. In this way we can model e.g. the change of user (movie) state caused by having watched and liked/disliked a movie (being liked/disliked by certain users). At each step of the user-state RNN, the network takes $y_t := W_{\text{embed}}\,[x_t, 1_{\text{newbie}}, \tau_t, \tau_{t-1}]$, where x_t is the rating vector, 1_{newbie} is the indicator for new users, and τ_t is wall-clock time. The jth element of x_t is the rating the user gives for movie j at time t, and 0 otherwise. The state update is given by standard $u_t := \text{LSTM}(u_{t-1}, y_t)$. In the above we omit user index for clarity. The movie-state RNN is defined in the same way.

Rating Emissions We supplement the time-varying profile vectors u_{it} and m_{jt} with stationary ones u_i and m_j respectively. These *stationary* components encode time-invariant properties such as long-term preference of a user or the genre of a movie.

The review rating is thus modeled as a function of both dynamic and stationary states, i.e.

$$r_{ij} = f(u_{it}, m_{jt}, u_i, m_j) := \langle \tilde{u}_{it}, \tilde{m}_{jt} \rangle + \langle u_i, m_j \rangle \tag{1}$$

where $\tilde{u}_{it}$ and $\tilde{m}_{jt}$ are affine functions of u_{it} and m_{jt} respectively. That is, we have

$$\tilde{u}_{it} = W_{\text{user}} u_{it} + b_{\text{user}} \quad \text{and} \quad \tilde{m}_{jt} = W_{\text{movie}} m_{jt} + b_{\text{movie}}$$

Review Text Model Review text is modeled by a character-level LSTM network. This network shares the same user/movie latent states with the rating model. We fuse the stationary and dynamic states of both user of movie by the bottleneck layer $x_{\text{joint},ij}$ given below:

$$x_{\text{joint},ij} := \phi(W_{\text{joint}}\,[u_{it}, m_{jt}, u_i, m_j] + b_{\text{joint}}) \quad \text{and} \quad \tilde{x}_{ij,k} := \left[x_{o_{ij,k}}, x_{\text{joint},ij}\right]$$

	PMF	Time-SVD++	U-AutoRec	I-AutoRec	RRN (rating)	RRN (rating + text)
IMDb	1.7355	1.7348	1.7332	1.7135	1.7047	**1.7012**
Netflix 6 months	0.9584	0.9589	0.9836	0.9778	**0.9427**	-

Table 1: RRN outperforms competing models in terms of RMSE. In addition, jointly modeling ratings and reviews achieves even better accuracy.

where $o_{ij,k}$ denotes the character at position k for the review given by user i to movie j, and $x_{o_{ij,k}}$ denotes the embedding of the character. ϕ here is some non-linear function. The review text emission model is itself an RNN, specifically a character-level LSTM generative model. For character index $k = 1, 2, \ldots,$

$$h_{ij,k} := \text{LSTM}(h_{ij,k-1}, \tilde{x}_{ij,k}) \quad \text{and} \quad \hat{o}_{ij,k} := \text{softmax}\left(W_{\text{out}} h_{ij,k} + b_{\text{out}}\right)$$

Here a softmax layer at output of LSTM is used to predict the next character.

Training & Prediction Our goal is to predict both accurate ratings and accurate reviews, and thus we minimize $L := \sum_{(i,j)\in\mathcal{D}_{\text{train}}} \left[\left(\hat{r}_{ij}(\theta) - r_{ij}\right)^2 - \lambda \sum_{k=1}^{n_{ij}} \log\left(\Pr(o_{ij,k}|\theta)\right) \right]$, where $\mathcal{D}_{\text{train}}$ is the training set of (i, j) pairs, θ denotes all model parameters, n_{ij} is the number of characters in the review user i gives to movie j, and λ controls the weight between predicting accurate ratings and predicting accurate reviews.

In prediction time, we make rating predictions based on predicted future states. That is, we take the latest ratings as input to update the states, and use the newly predicted states to predict ratings. This differs from traditional approaches where embeddings are estimated instead of inferred.

3 Experiments

We evaluate our model on a k-core of IMDb dataset, first used in Diao et al. (2014), that is the only large-scale movie review dataset available. The training set contains all ratings from July 1998 to December 2012, and the ratings from January to September 2013 are randomly split into a validation set and a test set. We compare our model with PMF (Mnih & Salakhutdinov, 2007), the state-of-the-art temporal model Time-SVD++ (Koren, 2010), and a state-of-the-art neural network-based model, AutoRec (Sedhain et al., 2015).

Rating prediction The results are summarized in Table 1. For completeness, we include the results from Wu et al. (2017) on 6-month Netflix dataset that use ratings only to compare the behavior of different models on different datasets. We see that rating-only RRN outperforms all baseline models in terms of rating prediction consistently in both datasets. More importantly, **joint-modeling ratings and reviews boosts the performance even more**, compared to rating-only RRN. This implies that by sharing statistical strength between ratings and reviews, the rich information in reviews helps us estimate the latent factors better.

Text modeling We also examine the impact of conditioning on user and item states for text modeling. Towards this end, we compare perplexity of characters in testing set with and without using the user/item factors. Perplexity is defined as $\exp\left(-\frac{1}{N_c}\sum_{c\in D_{test}} \log \Pr(c)\right)$, where N_c is the total number of characters in D_{test}, and $\Pr(c)$ is the likelihood of character c. Interestingly, we found that by jointly training with user and item states, the perplexity **improves** from 3.3442 to **3.3362**.

4 Discussion & Conclusion

We present a novel approach that jointly models ratings, reviews, and their temporal dynamics with RRN. We demonstrate that our joint model offers state-of-the-art results on rating prediction in real recommendation settings, i.e. predicting into the future.

REFERENCES

Alex Beutel, Amr Ahmed, and Alexander J Smola. ACCAMS: Additive Co-Clustering to Approximate Matrices Succinctly. In *WWW*, 2015.

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *KDD*. ACM, 2014.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4): 89–97, 2010.

Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*. ACM, 2013.

Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, 2007.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML*, 2008.

Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *WWW Companion*, 2015.

Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. Recurrent recommender networks. In *WSDM*, 2017.