ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Regularize and Explicit Collaborative Filtering With Textual Attention

Charles-Emmanuel Dias, Vincent Guigue and Patrick Gallinari

Sorbonne Universités, UPMC Univ Paris 06,
UMR 7606, LIP6, F-75005, Paris, France

Abstract. Recommendation can be seen as tantamount to blind sentiment analysis, i.e. a sentiment prediction without text data. In that sense, we aim at encoding priors on users and items while reading their reviews, using a deep architecture with personalized attention modeling. Following this idea, we build an hybrid hierarchical sentiment classifier which is then used as a recommender system in inference.

1 Introduction

Since the Netflix [1] competition, many collaborative filtering –especially matrix factorization– models where developed to exploit implicit [2] or explicit ratings [3] to build user and item profiles. Classically, $(user, item, ratings)$ triplets are the main source of information to build such profiles. Yet, taking advanced features into account such as review text [4] or time [5], leads to significant gains in accuracy. However, even if accurate, suggestions made by collaborative filtering can sometime be hard to understand or even spurious [6]. Such systems are therefore frequently considered as *blackboxes* [7]. This poor scrutability leads to a poor trust of users in recommendation systems which lowers their effectiveness. Thus, yielding understandable recommendations is as important as yielding accurate ones [8]. Consequently, we focus on two closely related scientific issues: building for each user and item relevant latent profiles; and sentiment extraction using an architecture derived from sentiment classification state of the art [9, 10].

Our goal is to have an effective and interpretable collaborative filtering system which uses review texts to improve and explicit suggestions. Learning a language model to regularize a matrix factorization framework by predicting words, sentences or reviews in addition to rating has been investigated in [11, 12]. In this paper, we follow a different route. We consider recommendation as blind sentiment analysis, namely an opinion prediction without text data, where the goal is to predict how a user will react to an item *a priori*. Our architecture encodes each users' and items' bias at a word and sentence level by using a hierarchical attention mechanism. Those bias are tied to the latent profiles for recommendation which aim at predicting review text instead of rating. We show that, fully grounding recommendation in text space, yields an effective and interpretable collaborative filtering algorithm.

The paper is organized as follows. First, in Section 2, we present our method which is evaluated in Section 3. Finally, Section 4 contains concluding remarks.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

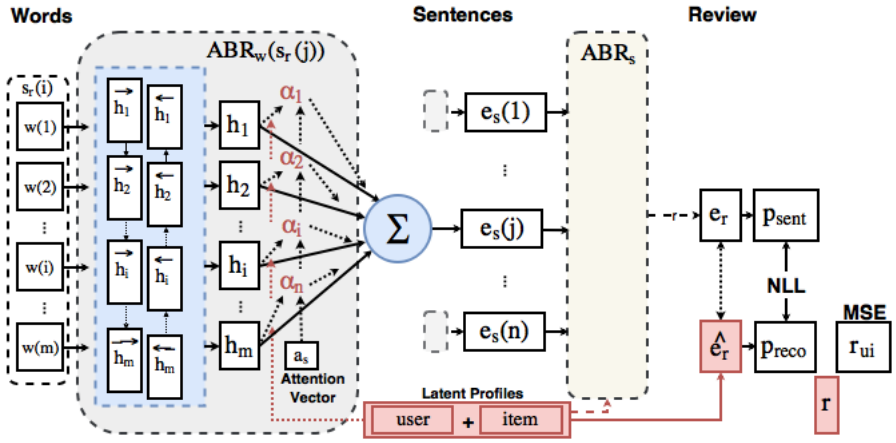

Fig. 1: The Hierarchical Network for Attentive Recommendation on a single input r. ABR_s embeds the review's n sentences into one vector e_r. Here, ABR_w encodes the j^{th} sentence

2 Attention to Regularize and Explicit Collaborative Filtering

Our goal is to predict how a user will react to an item, which is a standard collaborative filtering task. Ratings tend to be heavily biased as some people give always good/bad ratings and some items are under/overrated. This bias is usually modeled by the global rating average μ which is offset by an item bias b_i and a user bias b_u; this provides a strong baseline [3].

$$b_{ui} = \mu + b_i + b_u \qquad b_i = \mu_i - \mu \qquad b_u = \mu_u - \mu \tag{1}$$

Here, to refine this mean behavior, we build an attentional sentiment classifier coupled to a recommender system. To build latent profiles, we aim at extracting relevant informations from review text using an attention mechanism. The sentiment analysis sub-objective enables us to extract user/item priors from review text. We first describe how to build such *two-headed* network before explaining how to train it.

Hierarchical Network for Attentive Recommendation

Our model dubbed Hierarchical Network for Attentive Recommendation is depicted in figure 1. It is composed of two attentive bi-directional recurrent modules **(ABR)** to hierarchically encode reviews. Each attentive module is linked to a learnable user and item profiles which are used to build a surrogate review embedding in the recommendation setting. Both review and surrogate embeddings are classified using the same softmax classifier.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Attentive bi-directional recurrent module (ABR). It is the main building block of this model. It takes a sequence and a *(user,item)* embeddings pair as input and returns an attended embedding. Formally, given a sequence $seq = \{s_1, ..., s_i, ..., s_n\}$ composed of n elements. To obtain its embedding e_s, it is first fed through a bi-directional recurrent neural network $RF = \{\overrightarrow{RF}, \overleftarrow{RF}\}$ which reads it in both directions to get a sense of the local contexts within the sequence. Then, the output at each time step is concatenated to obtain a sequence of hidden representation h_i (eq. 2). Here, we use LSTM's [13] as recurrent cells.

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}], \qquad \overrightarrow{h_i} = \overrightarrow{RF}(s_i), \qquad \overleftarrow{h_i} = \overleftarrow{RF}(s_i), \tag{2}$$

Each element h_i is then projected into an attention space U to compute its affinity α_i with a learnt attention vector a and normalized with a softmax function. This attention vector a can be viewed as a question embedding, automatically learning discriminative features with respect to the task. The representation of the sequence e_s is the sum of the contextual embeddings of the sequence h_i, weighted by their affinity α_i computed in attention space with respect to the attention vector (eq. 3).

$$e_s = \sum_{i=1}^{n} \alpha_i h_i, \qquad \alpha_i = \frac{\exp(a^\intercal t_i)}{\sum_i \exp(a^\intercal t_i)} \tag{3}$$

In order to personalize attention, this projection in attention space is parametrized using the *(user,item)* embeddings (eq. 4).

$$t_i = tanh(W^{txt} h_i + W^{user} user + W^{item} item + b_t) \tag{4}$$

Customizing attention per *(user,item)* pair makes the model capable of learning a parametrized bias that we want to exploit for recommendation.

General architecture of the Network. It is simply made of two stacked ABR followed by a softmax classifier. The first ABR encodes each sentences from its sequence of words (ABR_w) while the second one encodes each previously encoded sentences representations into one final review embedding (ABR_s).

$$ABR_w : (s_r(t), p) \mapsto e_s(t) \qquad ABR_s : (\{e_s(t)\}, p) \mapsto e_r \tag{5}$$

Formally, given one input, a $(user, item) = p$ embeddings pair and its review r, composed of n sentences of variable word length t_n. We obtain its attended representation e_r by feeding each of its sentences embeddings $\{e_s(t)\}$ into ABR_s. Embeddings which were obtained by feeding sequentially each review's sentences $s_r(t)$ into ABR_w.

$$r = \{s_r(1), ..., s_r(n)\} = \{\{w(1), ..., w(t_1)\}^1, ..., \{w(1), ..., w(t_n)\}^n\} \tag{6}$$

$$e_r = ABR_s(\{ABR_w(s_r(1), ui), ..., ABR_w(s_r(n), ui)\}, ui) \tag{7}$$

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Dataset (#reviews)	Mean (μ)	w/offset	w/HFT	w/HNAR (ours)
Instant Video (37.126)	1.266	0.946	0.933	**0.923**
Digital Music (64,706)	1.165	0.857	0.844	**0.839**
Video Games (231,780)	1.441	1.122	**1.097**	**1.097**
CSJ (278,677)	1.218	1.134	1.107	**1.092**
Movie (1,697,533)	1.438	1.055	1.020	**1.012**

Table 1: Mean prediction error on the rating prediction task evaluated in MSE. Baselines are global rating average μ, full rating bias (eq.1) and HFT model [11]. Reported values are the mean prediction error over 5 splits.

In parallel, a surrogate review embedding $\hat{e_r}$ is computed using only the $(user, item)$ embeddings.

$$\hat{e_r} = user + item \tag{8}$$

Finally, those embeddings e_r and $\hat{e_r}$ are fed through the same classification layer.

$$p_{sent} = Softmax(W^{pred} e_r + b_p) \qquad p_{reco} = Softmax(W^{pred} \hat{e_r} + b_p) \tag{9}$$

The final predicted rating, is a learnt ratio between the mean ratings (eq.1) and the sum of the softmax output weighted by all the possible ratings r (1-5).

$$r_{ui} = (1 - \alpha) < \mathbf{r}.p_{reco} > + \alpha b_{ui} \qquad \alpha = \mathbb{R} \in [0, 1] \tag{10}$$

The whole network is trained end-to-end using gradient descent to both minimize cross-entropy error on sentiment analysis and mean-squared error on rating prediction. Regularization is done using a standard $\mathcal{L}_2$ norm penalization.

Explanation generation. Explanations are drawn from the model using the attention weights. For each suggestion made to user u on item i we aggregate every sentences from every reviews existing on item i which we map to our attention space. Following this we obtain a sorted list of each words, sentences and reviews existing which can be viewed as mimicking the user's attention over existing reviews to find potentially interesting features.

3 Experiments

In this section we first evaluate the quantitative performance of our model on the rating prediction task. Then, we present a peek inside the model and show how a recommendation can be explained by extracting sentences or words relevant to the user.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

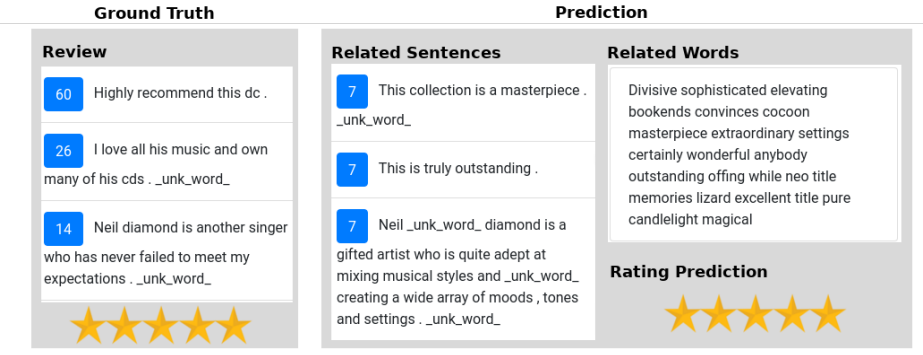

Fig. 2: Example of extracted sentences and words to highlight recommendation. Words are unordered. Sentences are ordered by attention score (in blue).

Protocol and Hyperparameters. For our experiments, we use several amazon reviews datasets of increasing sizes which are reduced such that each of the remaining users and items have 5 reviews each. We split each review text per sentences and tokenized words using *spacy*[1] NLP library. Words with less than 5 occurrences are discarded and replaced by an *unk* token. We used the pre-trained GloVe [14] vectors of 200 dimensions trained on Wikipedia and Gigaword[2] as word vectors and every hidden dimensions were set to size 200. Finally, we used Adam as optimizer.

Results. Quantitative Results are presented in Table 1. Baselines are computed using HFT [11] original implementation [3]. As Table 1 shows, appending an attentive network to the recommendation objective improves its performance a bit more than using a topic model. Moreover, attention can be used to get an insight on what is interesting about one item as it models each users rating behavior. Figure 2 shows an example of sentences and words extracted from existing reviews to explicit recommendation. In this particular example, it highlights that this item is about Neil Diamond and that his music is considered as "a classic".

4 Discussion & perspectives

In this paper, we propose an original vision of the recommendation task where the rating prediction in inference corresponds to a kind of blind sentiment analysis –without text–. We implemented a two-headed deep architecture that exploits attention parameters to build efficient user & item profiles: on the one hand,

[1] `https://spacy.io/`
[2] `https://nlp.stanford.edu/projects/glove/`
[3] `http://cseweb.ucsd.edu/~jmcauley/code/code_RecSys13.tar.gz`

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

attention improves sentiment classification and enables us to extract relevant explanations; on the other hand profiles are used to predict accurate ratings.

We consider several perspectives around this work, in particular, we would like to build a slight variant of the current system where the predicted rating is computed according to the sentiment found in the sentences selected by the attention model on the targeted item.

Acknowledgment: This work was partially founded by the FUI ITER-RH & BInD grants.

References

[1] James Bennett and Stan Lanning. The Netflix Prize. *KDD Cup and Workshop*, pages 3–6, 2007.

[2] Yehuda Koren and Robert Bell. Advances in Collaborative Filtering. *Recommender Systems Handbook*, pages 145–186, 2011.

[3] Y. Koren, R. Bell, and C. Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42:42–49, 2009.

[4] Mickaël Poussevin, Vincent Guigue, and Patrick Gallinari. Extended Recommendation Framework: Generating the Text of a User Review as a Personalized Summary. 2014.

[5] Elie Guardia-Sebaoun, Vincent Guigue, and Patrick Gallinari. Latent Trajectory Modeling : A Light and Efficient Way to Introduce Time in Recommender Systems. *the 2015 ACM conference on Recommender systems, RecSys 2015*, 2015.

[6] Doris Xin, N Goodwin Ave, Nicolas Mayoraz, Hubert Pham, and John R Anderson. Folding : Why Good Models Sometimes Make Spurious Recommendations. pages 201–209, 2017.

[7] Shay Ben-elazar and Noam Koenigstein. A Hybrid Explanations Framework for Collaborative Filtering Recommender Systems. In *RecSys 2014*, 2014.

[8] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. *Proceedings - International Conference on Data Engineering*, pages 801–810, 2007.

[9] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical Attention Networks for Document Classification. In *Proceedings of NAACL-HLT 2016*, 2016.

[10] Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. Neural Sentiment Classification with User and Product Attention. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[11] J McAuley and J Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13*, pages 165–172, 2013.

[12] Amjad Almahairi, Kyle Kastner, Kyunghyun Cho, and Aaron Courville. Learning Distributed Representations from Reviews for Collaborative Filtering. *Proceedings of the 9th ACM Conference on Recommender Systems - RecSys '15*, pages 147–154, 2015.

[13] S Hochreiter. Long Short-Term Memory. *Neural Computation*, 1997.

[14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.