Combining Rating and Review Data by Initializing Latent Factor Models with Topic Models for Top-N Recommendation

FRANCISCO J. PEÑA, Insight Centre for Data Analytics, University College Dublin, Ireland
DIARMUID O'REILLY-MORGAN, Insight Centre for Data Analytics, University College Dublin, Ireland
ELIAS Z. TRAGOS, Insight Centre for Data Analytics, University College Dublin, Ireland
NEIL HURLEY, Insight Centre for Data Analytics, University College Dublin, Ireland
ERIKA DURIAKOVA, Insight Centre for Data Analytics, University College Dublin, Ireland
BARRY SMYTH, Computer Science, University College Dublin, Ireland
AONGHUS LAWLOR, University College Dublin, Ireland

User and Item vector comes from Topic Modelling

Nowadays we commonly have multiple sources of data associated with items. Users may provide numerical ratings, or implicit interactions, but may also provide textual reviews. Although many algorithms have been proposed to jointly learn a model over both interactions and textual data, there is room to improve the many factorization models that are proven to work well on interactions data, but are not designed to exploit textual information. Our focus in this work is to propose a simple, yet easily applicable and effective, method to incorporate review data into such factorization models. In particular, we propose to build the user and item embeddings within the topic space of a topic model learned from the review data. This has several advantages: we observe that initializing the user and item embeddings in topic space leads to faster convergence of the factorization algorithm to a model that out-performs models initialized randomly, or with other state-of-the-art initialization strategies. Moreover, constraining user and item factors to topic space allows for the learning of an interpretable model that users can visualise.

ACM Reference Format:
Francisco J. Peña, Diarmuid O'Reilly-Morgan, Elias Z. Tragos, Neil Hurley, Erika Duriakova, Barry Smyth, and Aonghus Lawlor. 2020. Combining Rating and Review Data by Initializing Latent Factor Models with Topic Models for Top-N Recommendation. 1, 1 (September 2020), 9 pages. https://doi.org/10.1145/3383313.3412207

1 INTRODUCTION

Recommender systems (RS) have presented themselves as powerful tools to help users make the right choice. The first RSs were trained on explicit rating data provided by users for items, and learn a model to predict the ratings of unrated items. Later, research focused on the Top-N recommendation problem, learning to predict the set of N items that are most likely to satisfy a user's need. Matrix factorization (MF) models have proven highly effective on this task.

Authors' addresses: Francisco J. Peña, francisco.pena@insight-centre.org, Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland; Diarmuid O'Reilly-Morgan, diarmuid.oreillymorgan@insight-centre.org, Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland; Elias Z. Tragos, elias.tragos@insight-centre.org, Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland; Neil Hurley, neil.hurley@insight-centre.org, Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland; Erika Duriakova, erika.duriakova@ucd.ie, Insight Centre for Data Analytics, University College Dublin, Dublin, Ireland; Barry Smyth, barry.smyth@ucd.ie, Computer Science, University College Dublin, Dublin, Ireland; Aonghus Lawlor, aonghus.lawlor@insight-centre.org, University College Dublin, Dublin, Ireland.

Manuscript submitted to ACM

Such models [18, 19, 21, 22, 31] learn latent space embeddings of users and items either from explicit ratings, implicitly gathered interaction data or (recently) text data (i.e. user reviews).

Incorporating review data in RS models has proved to improve recommendation performance [7, 28]. In the area of text analytics, there exist a number of well-studied topic modelling algorithms that can find structure in textual information, and use that structure to cluster the data into meaningful topics. A number of state-of-the-art works have already addressed this question. Several approaches such as Collaborative Topic Regression (CTR) [32], Hidden Factors as Topics (HFT) [26], Ratings Meet Reviews (RMR) [25] and JMARS [10] strive to produce topic models and rating predictions by optimizing a hybrid loss function reducing the rating error and maximising the corpus likelihood. The topic models learned in these works are probabilistic generative models. Our work is different from all of the above, as we focus instead on initialization using topic models. Moreover, we forego joint learning, in favour of the flexibility of developing a methodology that can be applied to any existing latent factor RS model that learns user and item embeddings. Our approach is as follows; firstly we learn a topic model over the review data, extracting a topic space in which documents and words are embedded. Then, initializing the user and item factors of the MF problem within this topic space, we optimize these initial embeddings by minimising the loss function over the interaction data. Our proposed model out-performs a number of state-of-the-art initialization strategies, yielding more accurate RS models as evaluated on a number of empirical datasets. The fact that user topics describe user preferences and item topics describe item qualities helps the algorithm to achieve a high prediction accuracy in a small number of iterations. The use of a data source different to the rating matrix during the initialization process helps the algorithm to avoid local minima and ultimately reach higher prediction accuracy at convergence.

Overall our main contributions are the following:

(1) We use the results of topic modelling to initialize the latent factors of three well-known RS agorithms.
(2) We show that our model provides better performance against a number of state-of-the-art methods that initialize the latent factors with other strategies.
(3) We show how, within this methodology, we can obtain an interpretable model that users can visualize. This comes at the cost of sacrificing prediction accuracy, but our results show that the performance of our interpretable model is, in most cases, better than that of a randomly initialized model.

2 BACKGROUND

2.1 Initialization In Latent Factor Models

Latent factor models have proven to be very successful for both predicting user ratings and proposing Top-N recommendations. One of the first models designed specifically to produce Top-N recommendations is Weighted Regularized Matrix Facrtorization (WRMF) [18], which converts explicit feedback ratings into implicit feedback by binarizing the ratings into a preference ρ and then assigns a confidence value c to that preference. Preferences are predicted by a dot multiplication between the user latent vector $\mathbf{p}_u$ and the item latent vector $\mathbf{q}_i$. Similarly to WRMF, Bayesian Personalized Ranking (BPR) [31] predicts preferences by multiplying the user and item latent factor vectors, but focuses on making a pairwise ordering between items in which seen items should always be ranked above unseen items for every user. Rank-SGD [19] also uses a pairwise loss function, but includes the actual scores in the loss function. These approaches have proven to work well for Top-N recommendations, however they do not consider user reviews.

MF models are traditionally initialized with random values [14, 23]. However, their performance can be improved if more sophisticated initialization strategies are used. The two common goals of initialization are: (i) achieve faster convergence and (ii) reach better performance. In this paper, we focus more on the latter objective, but we can also achieve better performance in fewer iterations than other algorithms. One well-known initialization approach is NNDSVD [4] that is used for Non-negative Matrix Factorization (NMF). NNDSVD uses two Singular Value Decomposition (SVD) processes that are deterministic in order to find initialization values for the latent factor matrices. Hidasi and Tikk present SimFactor, an initialization method for Alternating Least Squares (ALS) that works with implicit feedback datasets [13, 14] and is based on the similarity between users and items. More recently, Nasiri and Minaei presented an initialization method that completes the missing entries from the sparse rating matrix using user and item averages, followed by factorizing the rating matrix with SVD [27], we call this method Average SVD. Our initialization strategy is different from the above that use the rating matrix both for initialization and model training, because for initialization we use topic models extracted from reviews, and we exploit the rating matrix only at the algorithm training step. The only exception being [14] that uses tags and contextual information in order to build similarity matrices.

2.2 Topic Modeling

Topic modeling is an information retrieval technique that aims to find a latent semantic structure between terms based on their co-occurrence within documents without relying on any form of labeled data [15]. In topic models, terms are grouped together into topics that typically represent a concept or a theme and topics are grouped into documents. Topic modeling algorithms use a document-term frequency matrix in order to create topic models. Well known algorithms include Probabilistic Latent Semantic Analysis (pLSA) [15] and Latent Dirichlet Allocation (LDA) [3] which are probabilistic generative models. NMF can also be used to decompose the document-term matrix and produce topic models as it is done in [24] and an ensemble of NMF models is presented in [2].

Topic based recommenders have been a popular approach to mix textual reviews with ratings. There are several models that jointly learn a topic model and a rating matrix [10, 25, 26, 32]. These derive from LDA [3] and jointly learn the topic model and the latent factors matrices for rating prediction using a probabilistic generative model. McAuley and Leskovec present the HFT [26] model that learns alternating between minimizing the prediction error in a step and then maximizing the log likelihood of the corpus in the next step. HFT uses a transformation function to relate the latent factors with the topics. RMR is presented in [25] and uses a mixture of Gaussians to model the ratings assuming that the mixture proportion has the same distribution as the topic distribution. In this way, the need for a transformation function is also avoided. Diao et al. present JMARS [10] an unsupervised model that mines aspects from movie reviews using topic modeling and integrates the mined aspects into the recommendation engine. In [8, 9] aspect-aware model that correlates the user and item embeddings on a set of aspects obtained from the reviews is presented. Hou et al. introduces a model called AMF [16] that is built on top of the ALFM model [30], but differently to ALFM, AMF pre-trains the aspects matrix by using LDA, and once the topic model has been created it uses it as part of the model to learn the latent factor matrices that serve to predict ratings.

In recommender systems in general, there seems to be a gap: models that are used for Top-N recommendation do not incorporate information from reviews [18, 19, 31] and models that include reviews are mostly designed for rating prediction [5, 6, 10, 25, 26, 32, 34], Joint Representation Learning (JRL) [33] being one of the few exceptions. We want to address this gap by improving the existing Top-N recommenders and adding reviews information into those models.

3 APPROACH

Our approach involves the following steps: (i) learn a topic model from review data; and (ii) initialise the user and item factors of a latent space recommender model in the topic space learned at step (i). Then,(iii), using the rating data, we optimise the user and item factors, by running an SGD to minimise the top-N loss function. We detail each step below.

Learning the topic model. Our datasets include a set of reviews for each item, written by users who accessed the item. These reviews constitute the "documents" over which the topic model is run. We firstly mine features from the reviews. Similarly to [11, 28, 29], we create a bag-of-words using a vocabulary consisting only of the nouns in the reviews, in order to obtain a TF-IDF matrix $\mathbf{T}$ of size $|W| \times |R|$, where $|W|$ is the size of the vocabulary and $|R|$ is the number of reviews. For example, in a dataset of hotel reviews, the nouns associated with a hotel might be words such as *swimming pool*, *bedroom*, *cleanliness* that capture a different aspect of a hotel. Then, using a topic modelling algorithm (i.e. LDA [3], NMF [24] or topic ensembling [2]) we obtain a $|W| \times k$ matrix, $\mathbf{H}$, representing an embedding of terms into a k-dimensional topic space, where k is the dimensional representation of each term in the vocabulary. To map users into topic space, we group all of the reviews written by a user into a single document and thus generate the TF-IDF matrix $\mathbf{T}_U$ which is $|U| \times |W|$, where $|U|$ is the number of users. Then, the user documents are "folded" into the topic model by applying a projection to $\mathbf{T}_U$ [2], $\mathbf{A} = \mathbf{T}_U \cdot \mathbf{H}$. In a similar manner, all reviews associated with an item can be used to fold the item into topic space, using the $|I| \times |W|$ TF-IDF matrix $\mathbf{T}_I$, $\mathbf{B} = \mathbf{T}_I \cdot \mathbf{H}$, where $|I|$ is the number of items. Each row of $\mathbf{A}$ now corresponds to a user, with columns corresponding to the k topics. An entry $\mathbf{A}_{u,t}$ indicates the strength of association of user u to the topic t. The same applies to the item-topic matrix $\mathbf{B}$.

Topic Initialized Latent Factor Model. Given a latent space dimension f, the goal of an MF recommendation model is to find a vector $\mathbf{p}_u \in \mathbb{R}^f$ for each user u and a vector $\mathbf{q}_i \in \mathbb{R}^f$ for each item i such that a prediction $\hat{y}(u, i)$, for a given (u, i) pair can be obtained from the inner product $\mathbf{p}_u^\top \mathbf{q}_i$. Gathering user vectors and item vectors into the $|U| \times f$ matrix $\mathbf{P}$ and the $|I| \times f$ matrix $\mathbf{Q}$, respectively, we can associate topic space with the latent space of the ratings factorization problem by setting $f = k$ and initializing $\mathbf{P} = \mathbf{A}$ or $\mathbf{Q} = \mathbf{B}$. From this initialization in topic space, the MF model further optimizes the factors, using the rating data. Our method can be applied to any model that employs latent factors. In fact, we will evaluate it on three such models, WRMF [18], BPR [31] and Rank-SGD [19].

4 EVALUATION

Datasets. We have selected four datasets from different domains to conduct our experiments. For all datasets we executed a preprocessing step in which we removed reviews that were repeated, that were not in English language and that had missing or erroneous IDs. Subsequently, we performed part-of-speech tagging and lemmatization for each one of the reviews. We perform lemmatization in order to group together words that are syntactic variants of the same base word. In the final step of the preprocessing task, we built bag of words for the reviews. Given that we can not create a user-topics (or item-topics) matrix without any reviews, we remove from the test set users and items that are not present in the train set. Table 1 provides a summary of the datasets after the data preprocessing task has been executed.

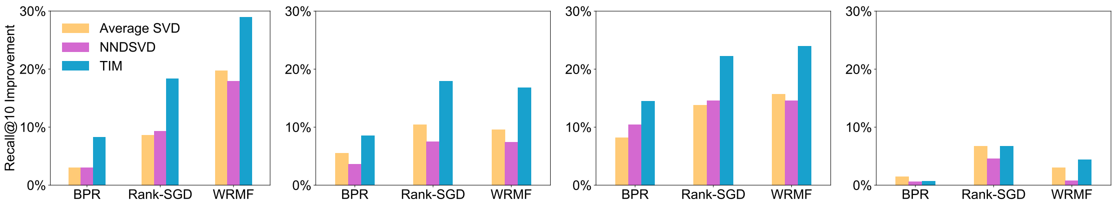
Evaluation Metrics. To evaluate the prediction performance of our algorithm in terms of ranking prediction we use (i) Precision@10, (ii) Recall@10, (iii) HitRatio@10 and (iv) NDCG@10 as the evaluation metrics [8, 20].

Baselines. We compare our methodology against random, NNDSVD [4] and Average SVD [27] initializations. We experiment initializing the following algorithms: (a) BPR[31], (b) Rank-SGD[19], and (c) WRMF[18].

Dataset	**Records**	**Users**	**Items**	**Sparsity**
Amazon Toys & Games	154,290	17,898	11,635	0.9993
Amazon Pet Supplies	147,385	18,645	8,395	0.9991
Amazon Health & Personal Care	323,553	36,432	17,996	0.9995
Yelp RecSys	141,393	7,634	5,315	0.9965

Table 1. Description of the datasets.

Evaluation Methodology. To evaluate our model we split the data three-ways in chronological order using a 80-10-10 ratio for training, validation and testing, respectively. The oldest 80% of the records are used to train the algorithms, the hyperparameter values are selected based on the performance on the validation set. The results in this section show the performance of the algorithms on the test set, selecting the best run across all iterations. The topic model is trained first on the train data and then it is used as an input to our model. To train the model we use 10 negative samples for each positive one as described in [12, 17]. To evaluate the model, we follow the evaluation protocol used in [12, 17], where for each positive item associated with a target user in the test set, we randomly sample 50 negative items that have no interaction records with the user. We report Recall, Hit Ratio (HT), NDCG and Precision at rank N as the evaluation metrics for measuring the model's accuracy recording the best run across all of the epochs for each algorithm.



(a) Amazon Toys & Games (b) Amazon Pet Supplies (c) Amazon Health (d) Yelp RecSys

Fig. 1. Recall@10 percentage improvement of different initialization models compared to random across multiple datasets

We initialized each of the algorithms BPR, Rank-SGD and WRMF with topic models created using NMF. We chose this combination because all methods had slightly better performance with NMF than with LDA and topic ensembling.

Impact Of The Initalization Strategy. In this section we compare Topic Initialized latent factors Model (TIM) against NNDSVD [4], one of most widely used methods for NMF initialization, and the more recent Average SVD [27] method. We used `scikit-learn`'s version of NNDSVD and our own implementation of Average SVD. Figure 1 shows the Recall@10 performance improvement of BPR, Rank-SGD and WRMF using the initialization methods compared to random initialization across the four datasets. It is evident that, TIM consistently improves the performance for all algorithms achieving the best Recall@10 in all cases except from the BPR at Yelp, which is very close to AverageSVD. These results are encouraging because they suggest that our algorithm is model agnostic.

Convergence Analysis. We compare the four initialization strategies used in combination with BPR, Rank-SGD and WRMF [32]. Table 2 displays the Recall@10 after 1, 10 and 100 iterations of each algorithm. We observe that NNDSVD outperforms TIM on Yelp, although not by much. This tells us that we should take care at the time of choosing the initialization strategy as some are more suited for certain metrics. TIM performs the best with WRMF after 100 iterations across all of the datasets, the same happens with BPR and Rank-SGD after one iteration. We also notice that on there

Algorithm	**Initialization**	**Amazon Toys**			**Amazon Pets**			**Amazon Health**			**Yelp RecSys**		
		1	**10**	**100**	**1**	**10**	**100**	**1**	**10**	**100**	**1**	**10**	**100**
BPR	Average SVD	0.093	0.121	0.247	0.187	0.264	0.346	0.136	0.173	0.295	0.144	0.157	0.225
	NNDSVD	0.189	0.196	0.249	0.298	0.315	0.342	0.229	**0.243**	0.304	0.177	**0.204**	**0.227**
	Random	0.164	0.176	0.210	0.239	0.256	0.282	0.212	0.224	0.263	0.141	0.162	0.205
	TIM	**0.200**	**0.231**	**0.262**	**0.299**	**0.319**	**0.361**	**0.233**	0.204	**0.316**	**0.166**	0.162	0.224
Rank-SGD	Average SVD	0.103	0.160	0.232	0.214	0.296	0.327	0.149	0.229	0.280	0.150	0.184	**0.216**
	NNDSVD	0.191	0.224	0.232	0.299	0.320	0.311	0.231	**0.279**	0.282	**0.180**	**0.208**	0.213
	Random	0.113	0.133	0.195	0.153	0.210	0.265	0.146	0.193	0.244	0.089	0.139	0.190
	TIM	**0.211**	**0.231**	**0.247**	**0.321**	**0.344**	**0.339**	**0.238**	0.273	**0.300**	0.162	0.191	0.212
WRMF	Average SVD	0.099	0.144	0.258	0.198	0.270	0.347	0.148	0.201	0.311	0.149	0.163	0.227
	NNDSVD	**0.190**	**0.219**	0.254	0.298	0.303	0.340	0.235	**0.257**	0.302	**0.180**	**0.199**	0.222
	Random	0.142	0.189	0.207	0.158	0.266	0.302	0.148	0.229	0.257	0.151	0.162	0.213
	TIM	0.183	0.201	**0.281**	**0.310**	**0.309**	**0.362**	**0.242**	0.232	**0.330**	0.166	0.167	**0.228**

Table 2. Recall@10 after 1, 10 and 100 iterations when seeded by different initializations strategies. The highest recall is in bold.

are a few cases in which NNDSVD has the best performance after 1 and 10 iterations, which is expected since it is well known that NNDSVD has a fast convergence [1, 4, 23]. However, after 100 iterations, TIM is able to catch up.

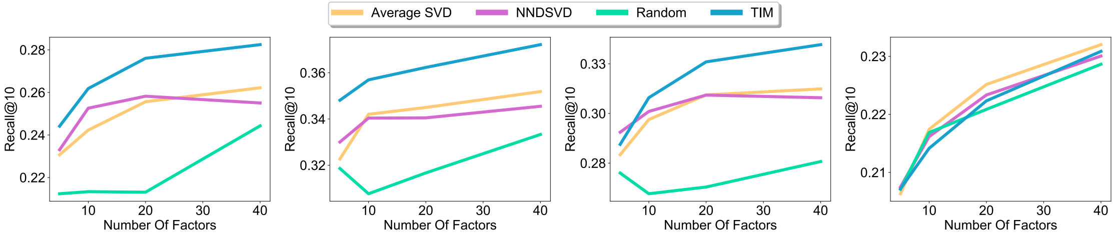
To analyse the performance at convergence time we executed each algorithm for 500 iterations. Table 3 reports the best result over the 500 iterations. Notice that TIM has the best performance across all of the datasets with the exception of BPR on Yelp RecSys. In summary TIM is able to achieve high accuracy both after a few and after many epochs. If one is looking for a quick solution then both TIM and NNDSVD perform well. However, for best performance which needs larger number of epochs, TIM demonstrated the ability to find the best solution.

Alg.	**Initializ.**	**Amazon Toys**				**Amazon Pets**				**Amazon Health**				**Yelp RecSys**			
		Recall	**HT**	**NDCG**	**Prec**	**Recall**	**HT**	**NDCG**	**Prec**	**Recall**	**HT**	**NDCG**	**Prec**	**Recall**	**HT**	**NDCG**	**Prec**
BPR	Avg SVD	0.252	0.516	0.164	0.072	0.352	0.650	0.208	0.095	0.298	0.596	0.185	0.081	**0.232**	**0.772**	**0.236**	**0.132**
	NNDSVD	0.252	0.513	0.165	0.072	0.345	0.642	0.207	0.093	0.305	0.601	0.187	0.082	0.230	0.767	0.231	0.131
	Random	0.244	0.512	0.155	0.070	0.333	0.627	0.194	0.090	0.276	0.569	0.171	0.074	0.229	0.767	0.229	0.130
	TIM	**0.264**	**0.535**	**0.169**	**0.076**	**0.362**	**0.660**	**0.215**	**0.097**	**0.316**	**0.614**	**0.190**	**0.085**	0.230	0.769	0.233	0.131
Rank-SGD	Avg SVD	0.240	0.492	0.156	0.068	0.333	0.629	0.201	0.090	0.291	0.586	0.179	0.079	0.220	0.756	0.227	0.125
	NNDSVD	0.241	0.501	0.158	0.069	0.324	0.622	0.199	0.087	0.294	0.588	0.183	0.079	0.215	0.752	0.222	0.123
	Random	0.221	0.475	0.143	0.063	0.302	0.593	0.182	0.081	0.256	0.540	0.162	0.069	0.206	0.735	0.216	0.117
	TIM	**0.261**	**0.532**	**0.166**	**0.075**	**0.356**	**0.652**	**0.212**	**0.096**	**0.313**	**0.609**	**0.188**	**0.085**	**0.220**	**0.757**	**0.228**	**0.125**
WRMF	Avg SVD	0.262	0.536	0.164	0.075	0.349	0.653	0.208	0.094	0.312	0.620	0.189	0.084	0.228	0.768	0.229	0.130
	NNDSVD	0.258	0.527	0.164	0.074	0.342	0.645	0.208	0.092	0.309	0.615	0.189	0.084	0.223	0.761	0.228	0.127
	Random	0.219	0.467	0.143	0.063	0.319	0.607	0.191	0.086	0.270	0.560	0.169	0.073	0.221	0.757	0.227	0.126
	TIM	**0.282**	**0.566**	**0.173**	**0.081**	**0.372**	**0.674**	**0.217**	**0.100**	**0.335**	**0.644**	**0.197**	**0.090**	**0.231**	**0.770**	**0.234**	**0.132**

Table 3. Top-N performance @10 over various datasets. The best performance is highlighted in bold.

Analysis Of The Influence Of The Number of Latent Factors On The Performance. In the following charts we can see how the number of latent factors has influence on the performance. In both datasets, the higher the number of topics the higher the recall, with 40 latent factors reaching the best performance. We remind the reader that the number of topics is also equal to the number of latent factors.

Interpretability. There are situations in which it is preferable to sacrifice prediction over interpretability, i.e. when explaining to a user why to stay in a certain hotel. NMF models are preferred over other models like SVD because its non-negativity allows to map the factor vectors to conceptual properties of the data [4, 23]. In the previous experiments,



(a) Amazon Toys & Games

(b) Amazon Pet Supplies

(c) Amazon Health

(d) Yelp RecSys

Fig. 2. Influence of the number of latent factors on Recall@10 on multiple datasets

we seeded both user and item latent factors with topic models, allowing the RS algorithm to optimize both user/item latent factor matrices. Here, we fix one of the latent factor matrices and optimize only the other one. In this way, the model training minimizes the ranking prediction error constrained to the topics on the other matrix. The practice of initializing one matrix and learning the remaining one is common among ALS algorithms [23]. Since the focus of this paper is on Stochastic Gradient Descent (SGD) we leave it to future work to explore ALS variants.

The advantage of relating topics directly to latent factors is that one can provide the users with visual explanations such as the one presented in Figure 3 in which we have plotted the weight of each latent factor (that relates directly to a topic) for an example user and a recommended item. To plot Figure 3 we labeled each axis with the word with the highest weight for a topic, we then rescaled each latent feature vector so that its maximum weight had a value of 1.0. This graph was created using the Yelp RecSys dataset and a topic model with five topics. Here, we indicate why we are suggesting a certain restaurant to a user: we can see that the user and the restaurant profiles are a close match.

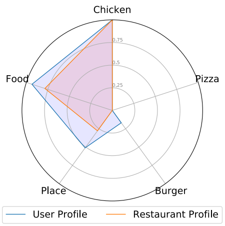


Fig. 3. User-restaurant profiles.

Alg.	Initializ.	Amazon Toys			Amazon Pets			Amazon Health			Yelp RecSys		
		Recall	HT	NDCG	Recall	HT	NDCG	Recall	HT	NDCG	Recall	HT	NDCG
BPR	TIM-I	0.17	0.38	0.12	0.24	0.51	0.16	0.20	0.43	0.13	0.11	0.48	0.13
	TIM-U	**0.22**	**0.47**	**0.14**	**0.32**	**0.62**	**0.19**	**0.24**	**0.52**	**0.16**	**0.17**	**0.63**	**0.17**
Rank-SGD	TIM-I	0.17	0.38	0.12	0.24	0.50	0.16	0.20	0.43	0.13	0.11	0.47	0.13
	TIM-U	**0.22**	**0.47**	**0.14**	**0.32**	**0.62**	**0.19**	**0.24**	**0.53**	**0.16**	**0.17**	**0.63**	**0.17**
WRMF	TIM-I	0.17	0.38	0.12	0.24	0.51	0.16	0.20	0.43	0.13	0.11	0.49	0.13
	TIM-U	**0.22**	**0.48**	**0.14**	**0.32**	**0.62**	**0.19**	**0.25**	**0.53**	**0.16**	**0.17**	**0.63**	**0.17**

Table 4. Performance comparison between TIM-U and TIM-I initializations.

As we can see in Table 4 interpretability comes at a cost. Here we have the TIM-U model, which we initialize with $\mathbf{P} = \mathbf{A}$ and $\mathbf{Q} = \mathbf{B}$. The parameters of $\mathbf{P}$ remain constant while the loss function is optimized by varying only the values of $\mathbf{Q}$ (in TIM-I $\mathbf{Q}$ is left fixed). Because only one matrix is to be learned, TIM-U has a fast convergence, but unlike TIM, TIM-U does not outperform the random initialization across all datasets (compared with the values in Table 3). We also see that there is a drop in the performance of TIM-U and TIM-I compared to TIM, which is expected since in TIM we don't fix the initialised latent factor matrices and we continue to optimise their values. We don't expect TIM-I to perform well since the grouped item reviews are written by different users forming a very heterogeneous document. On the other hand, grouped user reviews are written by the same users and therefore express users preferences in a consistent way (for most users), this results in cleaner user preferences obtained by the topic model. Depending on the dataset and on the situation, one might consider sacrificing prediction power and choosing the TIM-U model to favor interpretability.

5 DISCUSSION AND CONCLUSIONS

We have presented TIM, a model that builds document topic matrices using them as a seed for the latent factor matrices in matrix factorization models for Top-N recommendations. The topics represent user preferences (or item qualities) and are used as ground truth to learn latent factors, allowing for interpretability. Our model can be integrated into several existing latent factor models such as BPR, Rank-SGD and WRMF among many others. We evaluated TIM on four datasets from different domains showing superior performance in terms of ranking prediction.

One particular difference between NNDSVD, Average SVD and TIM is that the former two exploit information from the rating matrix in order to build the initialization matrices, while TIM uses the reviews transformed into topic models. Topic models (frequently used to rank documents) can be very helpful to rank items, which gives TIM an advantage for improved performance. Higher convergence speed is also achieved with topic models because they help gather user preferences and item qualities from the reviews, putting the algorithm in a better initial position for the optimization.

Building a model that jointly learns from both reviews and ratings simultaneously is a problem with a higher complexity and a bigger search space than learning only ratings. In our case, the goal is to improve Top-N recommendations, therefore we discard the additional complexity given by the joint models and focus only on optimizing the ranking prediction function. Nevertheless, if interpretability is the goal, we have provided the Topic-User Initialized latent factors Model (TIM-U) along with an example of a visual interpretation. However, as followup work we also plan to analyse and compare the approach of jointly learning our model with the two step approach adopted in this work.

ACKNOWLEDGMENTS

This work was supported by the Science Foundation Ireland through the Insight Centre for Data Analytics under grant number SFI/12/RC/2289_P2.

REFERENCES

[1] Syed Muhammad Atif, Sameer Qazi, and Nicolas Gillis. 2019. Improved SVD-based initialization for nonnegative matrix factorization using low-rank correction. *Pattern Recognition Letters* 122 (2019), 53 – 59. https://doi.org/10.1016/j.patrec.2019.02.018

[2] Mark Belford, Brian Mac Namee, and Derek Greene. 2018. Stability of topic modeling via matrix factorization. *Expert Systems with Applications* 91 (2018), 159 – 169. https://doi.org/10.1016/j.eswa.2017.08.047

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (March 2003), 993–1022.

[4] C. Boutsidis and E. Gallopoulos. 2008. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition* 41, 4 (2008), 1350 – 1362. https://doi.org/10.1016/j.patcog.2007.09.010

[5] Rose Catherine and William Cohen. 2017. TransNets: Learning to Transform for Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. Association for Computing Machinery, New York, NY, USA, 288–296. https://doi.org/10.1145/3109859.3109878

[6] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-Level Explanations. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1583–1592. https://doi.org/10.1145/3178876.3186070

Read the review!!!

[7] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender Systems Based on User Reviews: The State of the Art. *User Modeling and User-Adapted Interaction* 25, 2 (June 2015), 99–154. https://doi.org/10.1007/s11257-015-9155-5

[8] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose C. Kanjirathinkal, and Mohan Kankanhalli. 2019. MMALFM: Explainable Recommendation by Leveraging Reviews and Images. *ACM Trans. Inf. Syst.* 37, 2, Article 16 (Jan. 2019), 28 pages. https://doi.org/10.1145/3291060

[9] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. 2018. Aspect-Aware Latent Factor Model: Rating Prediction with Ratings and Reviews. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 639–648. https://doi.org/10.1145/3178876.3186145

[10] 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 (KDD '14)*. ACM, New York, NY, USA, 193–202. https://doi.org/10.1145/2623330.2623758

[11] Ruihai Dong, Markus Schaal, Michael P. O'Mahony, Kevin McCarthy, and Barry Smyth. 2013. Opinionated Product Recommendation. In *Case-Based Reasoning Research and Development*, Sarah Jane Delany and Santiago Ontañón (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 44–58.

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. https://doi.org/10.1145/3038912.3052569

[13] Balázs Hidasi and Domonkos Tikk. 2012. Enhancing Matrix Factorization through Initialization for Implicit Feedback Databases. In *Proceedings of the 2nd Workshop on Context-Awareness in Retrieval and Recommendation (CaRR '12)*. Association for Computing Machinery, New York, NY, USA, 2–9. https://doi.org/10.1145/2162102.2162104

[14] Balazs Hidasi and Domonkos Tikk. 2013. Initializing Matrix Factorization Methods on Implicit Feedback Databases. 19, 12 (jun 2013), 1834–1853.

[15] Thomas Hofmann. 2001. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Mach. Learn.* 42, 1/2 (Jan. 2001), 177–196. https://doi.org/10.1023/A:1007617005950

[16] Yunfeng Hou, Ning Yang, Yi Wu, and Philip S. Yu. 2019. Explainable recommendation with fusion of aspect information. *World Wide Web* 22, 1 (01 Jan 2019), 221–240. https://doi.org/10.1007/s11280-018-0558-1

[17] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging Meta-Path Based Context for Top- N Recommendation with A Neural Co-Attention Model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. Association for Computing Machinery, New York, NY, USA, 1531–1540. https://doi.org/10.1145/3219819.3219965

[18] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining*. 263–272. https://doi.org/10.1109/ICDM.2008.22

[19] Michael Jahrer and Andreas Töscher. 2011. Collaborative Filtering Ensemble for Ranking. In *Proceedings of the 2011 International Conference on KDD Cup 2011 - Volume 18 (KDDCUP'11)*. JMLR.org, 153–167.

[20] Kalervo Järvelin and Jaana Kekäläinen. 2000. IR Evaluation Methods for Retrieving Highly Relevant Documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '00)*. Association for Computing Machinery, New York, NY, USA, 41–48. https://doi.org/10.1145/345508.345545

[21] 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 (KDD '08)*. Association for Computing Machinery, New York, NY, USA, 426–434. https://doi.org/10.1145/1401890.1401944

[22] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug. 2009), 30–37. https://doi.org/10.1109/MC.2009.263

[23] Amy N Langville, Carl D Meyer, Russell Albright, James Cox, and David Duling. 2014. Algorithms, initializations, and convergence for the nonnegative matrix factorization. *arXiv preprint arXiv:1407.7299* (2014).

[24] Daniel D Lee and H. Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401 (10 1999), 788–791. https://doi.org/10.1038/44565

[25] 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 (RecSys '14)*. ACM, New York, NY, USA, 105–112. https://doi.org/10.1145/2645710.2645728

[26] 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 (RecSys '13)*. ACM, New York, NY, USA, 165–172. https://doi.org/10.1145/2507157.2507163

[27] Mahdi Nasiri and Behrouz Minaei. 2016. Increasing Prediction Accuracy in Collaborative Filtering with Initialized Factor Matrices. *J. Supercomput.* 72, 6 (June 2016), 2157–2169. https://doi.org/10.1007/s11227-016-1717-8

[28] Francisco J. Peña. 2017. Unsupervised Context-Driven Recommendations Based On User Reviews. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 426–430. https://doi.org/10.1145/3109859.3109865

[29] Francisco J. Peña and Derek Bridge. 2017. Recommending from Experience. In *Proceedings of the Thirtieth Florida Artificial Intelligence Research Society Conference*. https://aaai.org/ocs/index.php/FLAIRS/FLAIRS17/paper/view/15439

[30] Lin Qiu, Sheng Gao, Wenlong Cheng, and Jun Guo. 2016. Aspect-based latent factor model by integrating ratings and reviews for recommender system. *Knowledge-Based Systems* 110 (2016), 233 – 243. https://doi.org/10.1016/j.knosys.2016.07.033

[31] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[32] 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 (KDD '11)*. ACM, New York, NY, USA, 448–456. https://doi.org/10.1145/2020408.2020480

[33] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W. Bruce Croft. 2017. Joint Representation Learning for Top-N Recommendation with Heterogeneous Information Sources. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 1449–1458. https://doi.org/10.1145/3132847.3132892

[34] 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 (WSDM '17)*. Association for Computing Machinery, New York, NY, USA, 425–434. https://doi.org/10.1145/3018661.3018665