






Novel and Diverse Recommendations by Leveraging Linear Models with User and Item Embeddings

Alfonso Landin^(✉), Javier Parapar^{}, and Álvaro Barreiro^{}

Information Retrieval Lab, Centro de Investigación TIC (CITIC),
Universidade da Coruña, A Coruña, Spain
{alfonso.landin,javierparapar,barreiro}@udc.es

Abstract. Nowadays, item recommendation is an increasing concern for many companies. Users tend to be more reactive than proactive for solving information needs. Recommendation accuracy became the most studied aspect of the quality of the suggestions. However, novel and diverse suggestions also contribute to user satisfaction. Unfortunately, it is common to harm those two aspects when optimizing recommendation accuracy. In this paper, we present EER, a linear model for the top-N recommendation task, which takes advantage of user and item embeddings for improving novelty and diversity without harming accuracy.

Keywords: Collaborative filtering · Novelty · Diversity · User and item embeddings

1 Introduction

In recent years, the way users access services has shifted from a proactive approach, where the user actively looks for the information, to one where the users take a more passive role, and content is suggested to them. Within this transformation, Recommender Systems have played a pivotal role, enabling an increase in user engagement and revenue.

Recommender Systems are usually classified into three families [1]. The first approach, content-based systems, use item metadata to produce recommendations [7]. The second family, collaborative filtering, is composed of systems that exploit the past interactions of the users with the items to compute the recommendations [10, 17]. These interactions can take several forms, such as ratings, clicks, purchases. Finally, hybrid approaches combine both to generate suggestions. Collaborative Filtering (CF) systems can be divided into memory-based systems, that use the information about these interactions directly to compute the recommendations, and model-based systems, that build models from this information that are later used to make the recommendations.

In this paper, we will present a CF model to address the top-N recommendation task [4]. The objective of a top-N recommender is to produce a ranked list

of items for each user. These systems can be evaluated using traditional IR metrics over the rankings [2, 4]. In that evaluation approach, accuracy is usually the most important metric and has been the focus of previous research and competitions [3]. Nevertheless, other properties are also important, such as diversity and novelty [8, 13]. Diversity is the ability of the system to make recommendations that include items equitably from the whole catalog, which is usually desired by vendors [5, 22]. On the other hand, novelty is the capacity of the system to produce unexpected recommendations. This characteristic is a proxy for serendipity, associated with higher user engagement and satisfaction [6]. All these properties, accuracy, diversity and novelty, are linked to the extent that raising accuracy usually lowers the best achievable results in the other properties [11].

In this paper, we propose a method to augment an existing recommendation linear model to make more diverse and novel recommendations, while maintaining similar accuracy results. We do so by making use of user and item embeddings that are able to capture non-linear relations thanks to the way they are obtained [21]. Experiments conducted on three datasets show that our proposal outperforms the original model in both novelty and diversity while maintaining similar levels of accuracy. With reproducibility in mind, we also make the software used for the experiments publicly available¹.

2 Background

In this section, we introduce FISM, the existing recommendation method we augment in our proposal. After that, we introduce `prefs2vec`, the user and item embedding model used to make this enhancement.

2.1 FISM

FISM is a state-of-the-art model-based recommender system proposed by Kabbur et al [9]. This method learns a low rank factorization of an item-item similarity matrix, which is later used to compute the scores to make the predictions. This method is an evolution of a previous method, SLIM [16], that learns this matrix without factorizing it. Factorizing the similarity matrix allows FISM to overcome SLIM’s limitation of not being able to learn a similarity other than zero for items that have never been rated both by at least one user. As a side effect of this factorization, it lowers the space complexity from $\mathcal{O}(|\mathcal{I}|^2)$ to $\mathcal{O}(|\mathcal{I}| \times k)$, $k \ll |\mathcal{I}|$. It also drops the non-negativity constraint and the constraint that the diagonal of the similarity matrix has to contain zeroes. As a consequence of these changes, the optimization problem can be solved using regular gradient descent algorithms, instead of the coordinated gradient descent used by SLIM, leading to faster training times.

¹ <https://gitlab.irilab.org/irilab/eer>.

2.2 User and Item Embeddings

Embedding models allow transforming high-dimensional and sparse vector representations, such as classical one-hot and bag-of-words, into a space with much lower dimensionality. In particular, previous word embedding models, that produce fixed-length dense representations, have proven to be more effective in several NPL tasks [14, 15, 19].

Recently, `prefs2vec` [21], a new embedding model for obtaining dense user and item representations, an adaptation of the CBOW model [14], has shown that these embeddings can be useful for the top-N recommendation task. When used with a memory-based recommender, they are more efficient than the classical representation [21]. The results show that not only they can improve the accuracy of the results, but also their novelty and diversity. The versatility of this embedding model, in particular of the underlying neural model and the way it is trained, is also shown in [12]. Here the prediction capabilities of the neural model are used directly in a probabilistic recommender.

3 Proposal

In this section, we present our method to enhance diversity and novelty in recommendation, explaining how the model is trained and used to produce recommendations. Firstly, we introduce how the product of user and item embeddings (based on `prefs2vec`) can be used to make recommendations, which is later used as part of the proposal.

3.1 User and Item Embeddings Product

As representations of users and items in a space with much lower dimensionality, `prefs2vec` embeddings can be viewed as latent vectors. However, there is no sense in multiplying both item and user vectors as they have different basis even when they have the same dimensions. This is a consequence of learning the item and user representations independently, how `prefs2vec` initializes the parameters of the model and how the training is performed.

However, it is possible to make this product if we can compute a change of basis matrix $\mathbf{T} \in \mathbb{R}^{d \times d}$ to transform the user embeddings into the item embeddings space. This way we can calculate an estimated ratings matrix $\hat{\mathbf{R}}$ using the simple matrix multiplication:

$$\hat{\mathbf{R}} = \mathbf{E}\mathbf{T}\mathbf{F}^T \tag{1}$$

where $\mathbf{E} \in \mathbb{R}^{|\mathcal{U}| \times d}$ is the matrix of user embeddings, and $\mathbf{F} \in \mathbb{R}^{|\mathcal{I}| \times d}$ is the matrix of item embeddings, one embedding in each row. The transformation matrix \mathbf{T} is learned by solving the optimization problem with ℓ_2 regularization:

$$\underset{\mathbf{T}}{\text{minimize}} \frac{1}{2} \|\mathbf{R} - \hat{\mathbf{R}}\|_F^2 + \frac{\beta_e}{2} \|\mathbf{T}\|_F^2 \tag{2}$$

where \mathbf{R} is the ratings matrix and β_e is the regularization hyperparameter. This problem can be solved using gradient descent algorithms.

Once the transformation matrix has been trained, recommendations can be produced by computing the estimated rating matrix $\hat{\mathbf{R}}$ as described in Eq. 1. Recommendations are made to each user by sorting the corresponding row and picking the top-N items not already rated by the user. We dubbed this recommender ELP, short for Embedding Linear Product, and we present its performance in Table 3 in the experiments section.

3.2 Embedding Enhanced Recommender

We have seen that linear methods, like FISM, can obtain good accuracy figures. On the other side, as results in Table 3 show, ELP is able to provide good figures in novelty and diversity, thanks to the embedding model capturing non-linear relations between users and items.

We propose to capture both properties by joining the models together in the EER model (Embedding Enhanced Recommender). We choose the RMSE variant of FISM as it matches the loss used in ELP. We also use a trainable scalar parameter α to joint the models, as the scores obtained from each recommender need not be on the same scale. This results in the following equation to calculate the estimated ratings matrix:

$$\hat{\mathbf{R}} = \mathbf{R}\mathbf{P}\mathbf{Q} + \alpha\mathbf{E}\mathbf{T}\mathbf{F}^\top \quad (3)$$

where $\mathbf{P} \in \mathbb{R}^{|\mathcal{I}| \times k}$ and $\mathbf{Q} \in \mathbb{R}^{k \times |\mathcal{I}|}$ are the low rank factorization of the item-item similarity matrix. The parameters of the model, \mathbf{P} , \mathbf{Q} , \mathbf{T} and α , are learned by solving the joint ℓ_2 regularized optimization problem resulting from the previous joint equation, using standard gradient descent algorithms:

$$\underset{\mathbf{P}, \mathbf{Q}, \mathbf{T}, \alpha}{\text{minimize}} \frac{1}{2} \|\mathbf{R} - \hat{\mathbf{R}}\|_F^2 + \frac{\beta}{2} (\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) + \frac{\beta_e}{2} \|\mathbf{T}\|_F^2 \quad (4)$$

Similar to the case of ELP, once the parameters are learned, we make the recommendations by calculating the estimated ratings matrix using Eq. 3, sorting each row and picking the top-N items not yet rated by the user corresponding to that row.

4 Experiments and Results

In this section, we introduce the datasets used to perform our experiments, the evaluation protocol followed and the metrics used. After that, we present the results of our experiments.

4.1 Datasets

To evaluate our proposal, we conducted a series of experiments on several datasets, from different domains: the MovieLens 20M dataset², a movie dataset,

² <https://grouplens.org/datasets/movielens/20m/>.

the book dataset LibraryThing, and the BeerAdvocate dataset³, consisting of beer reviews. Table 1 shows statistics of each collection. In order to perform the experiments, the datasets were divided randomly into train and test sets. The training dataset consisted of 80% of the ratings of each user, with the remaining 20% forming the test dataset.

Table 1. Statistics of the collections.

Dataset	Users	Items	Ratings	Density
MovieLens 20M	138,493	26,744	20,000,263	0.540%
LibraryThing	7,279	37,232	749,401	0.277%
BeerAdvocate	33,388	66,055	1,571,808	0.071%

4.2 Evaluation Protocol

We follow the TestItems evaluation methodology [2] to evaluate the performance. To assess the accuracy of the rankings, we use Normalized Discounted Cumulative Gain (nDCG), using the *standard formulation* as described in [23], with the ratings in the test set as graded relevance judgments. We considered only items with a rating of 4 or more, on a 5 point scale, to be relevant for evaluation purposes. We also measured the diversity of the recommendations using the complement of the Gini index [5]. Finally, we use the mean self-information (MSI) [24] to assess the novelty of the recommendations. All the metrics are evaluated at cut-off 100 because it has shown to be more robust with respect to the sparsity and popularity biases than shallower cut-offs [20]. We perform a Wilcoxon test [18] to assess the statistical significance of the improvements regarding nDCG@100 and MSI@100, with $p < 0.01$. We cannot apply it to the Gini index because we are using a paired test and Gini is a global metric. Results in Table 3 are annotated with their statistical significance.

4.3 Results and Discussion

We performed a grid search over the hyperparameters of the original model and our proposal tuning them to maximize nDCG@100. Although we aim to increase diversity and novelty, we want the recommendations to be effective, which is why the tuning is done over accuracy. For the parameters of the `prefs2vec` model, we took those that performed better in [21]. For reproducibility’s sake, values for the best hyperparameters for each collection can be consulted in Table 2.

Table 3 shows the values of nDCG@100, Gini@100 and MSI@100 for FISM, EER and ELP. The results show that EER outperforms the baseline (FISM) on both novelty and diversity. It also surpasses it on accuracy on the MovieLens 20M and LibraryThing datasets. In the case of diversity, we can see important

³ <https://snap.stanford.edu/data/web-BeerAdvocate.html>.

Table 2. Best values of the hyperparameters for nDCG@100 for FISM and our proposals EER and ELP.

Model	MovieLens 20M	LibraryThing	BeerAdvocate
FISM	$\beta = 1, k = 1000$	$\beta = 1000, k = 1000$	$\beta = 50, k = 1000$
ELP	$\beta_e = 0.1$	$\beta_e = 10$	$\beta_e = 10$
EER	$\beta = 0.1, \beta_e = 1, k = 1000$	$\beta = 500, \beta_e = 10, k = 1000$	$\beta = 10, \beta_e = 1, k = 1000$

Table 3. Values of nDCG@100, Gini@100 and MSI@100 on MovieLens 20M, LibraryThing and BeerAdvocate datasets. Statistical significant improvements, according to Wilcoxon test with $p < 0.01$, in nDCG@100 and MSI@100 with respect to FISM and our proposals EER and ELP are superscripted with a , b and c respectively.

Model	Metric	MovieLens 20M	LibraryThing	BeerAdvocate
FISM	nDCG@100	0,4641 ^c	0,2878 ^c	0,1502^{bc}
	Gini@100	0,0390	0,0896	0,0363
	MSI@100	230,5480	414,3157	324,4954
EER	nDCG@100	0,4665^{ac}	0,3017^{ac}	0,1452 ^c
	Gini@100	0,0412	0,1072	0,0521
	MSI@100	234,0325 ^a	416,6850 ^a	328,2118 ^a
ELP	nDCG@100	0,3322	0,1850	0,0855
	Gini@100	0,0808	0,2901	0,3221
	MSI@100	307,9538^{ab}	532,9078^{ab}	519,5824^{ab}

improvements. ELP, on the other hand, obtains the best diversity and novelty values, but this comes with a big reduction in accuracy. It is common in the field of recommender systems for methods with lower accuracy to have higher values in diversity and novelty. We believe that the ability of the embeddings to find non-linear relationships contributes to the model novelty and diversity. This property of the model allows it, for example, to discover relationships between popular and not so popular items leading to better diversity. Moreover, the integration in the linear model allows to keep its advantage in terms on accuracy, clearly surpassing the use of embeddings in isolation (ELP).

5 Conclusions and Future Work

In this paper, we presented EER, a method to enhance an existing recommendation algorithm to produce recommendations that are both more diverse and novel, while maintaining similar levels on accuracy. This process is done by combining two models, a linear one that is able to obtain good levels of accuracy, with a model based in an embedding technique that extracts non-linear relationships, allowing it to produce more diverse and novel recommendations.

As future work, we plan to apply the same technique to other recommender systems, examining if it can be applied in general to enhance the recommendations, independently of the base algorithm chosen for the task. We also envision studying the effects that varying the value of α in Eq. 3 has on the recommendations.

Acknowledgements. This work was supported by project RTI2018-093336-B-C22 (MCIU & ERDF), project GPC ED431B 2019/03 (Xunta de Galicia & ERDF) and accreditation ED431G 2019/01 (Xunta de Galicia & ERDF). The first author also acknowledges the support of grant FPU17/03210 (MCIU).

References

- Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. *Commun. ACM* **40**(3), 66–72 (1997). <https://doi.org/10.1145/245108.245124>
- Bellofón, A., Castells, P., Cantador, I.: Precision-oriented evaluation of recommender systems. In: Proceedings of the 5th ACM Conference on Recommender Systems, RecSys 2011, pp. 333–336. ACM, New York (2011). <https://doi.org/10.1145/2043932.2043996>
- Bennett, J., Lanning, S., et al.: The netflix prize. In: Proceedings of KDD Cup and Workshop, vol. 2007, p. 35. ACM, New York (2007)
- Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the 4th ACM Conference on Recommender Systems, RecSys 2010, pp. 39–46. ACM, New York (2010). <https://doi.org/10.1145/1864708.1864721>
- Fleder, D., Hosanagar, K.: Blockbuster culture’s next rise or fall: the impact of recommender systems on sales diversity. *Manag. Sci.* **55**(5), 697–712 (2009). <https://doi.org/10.1287/mnsc.1080.0974>
- Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: RecSys 2010, pp. 257–260. ACM (2010). <https://doi.org/10.1145/1864708.1864761>
- de Gemmis, M., Lops, P., Musto, C., Narducci, F., Semeraro, G.: Semantics-aware content-based recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) *Recommender Systems Handbook*, pp. 119–159. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6_4
- Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **22**(1), 5–53 (2004). <https://doi.org/10.1145/963770.963772>
- Kabbur, S., Ning, X., Karypis, G.: FISM: factored item similarity models for top-n recommender systems. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, pp. 659–667. ACM, New York (2013). <https://doi.org/10.1145/2487575.2487589>
- Koren, Y., Bell, R.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B. (eds.) *Recommender Systems Handbook*, pp. 77–118. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6_3
- Landin, A., Suárez-García, E., Valcarce, D.: When diversity met accuracy: a story of recommender systems. *Proceedings* **2**(18) (2018). <https://doi.org/10.3390/proceedings2181178>

12. Landin, A., Valcarce, D., Parapar, J., Barreiro, Á.: PRIN: a probabilistic recommender with item priors and neural models. In: Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., Hiemstra, D. (eds.) ECIR 2019. LNCS, vol. 11437, pp. 133–147. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-15712-8_9
13. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: Extended Abstracts on Human Factors in Computing Systems, CHI EA 2006. p. 1097. ACM Press, New York (2006). <https://doi.org/10.1145/1125451.1125659>
14. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. CoRR abs/1301.3, pp. 1–12 (2013)
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS 2013, pp. 3111–3119. Curran Associates Inc., USA (2013)
16. Ning, X., Karypis, G.: Slim: sparse linear methods for top-n recommender systems. In: 2011 IEEE 11th International Conference on Data Mining, pp. 497–506 (2011). <https://doi.org/10.1109/ICDM.2011.134>
17. Ning, X., Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 37–76. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6_2
18. Parapar, J., Losada, D.E., Presedo-Quindimil, M.A., Barreiro, A.: Using score distributions to compare statistical significance tests for information retrieval evaluation. *J. Assoc. Inf. Sci. Technol.* (2019). <https://doi.org/10.1002/asi.24203>
19. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, pp. 1532–1543. ACL, Stroudsburg (2014). <https://doi.org/10.3115/v1/D14-1162>
20. Valcarce, D., Bellogín, A., Parapar, J., Castells, P.: On the robustness and discriminative power of information retrieval metrics for top-n recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, pp. 260–268. ACM, New York (2018). <https://doi.org/10.1145/3240323.3240347>
21. Valcarce, D., Landin, A., Parapar, J., Barreiro, Á.: Collaborative filtering embeddings for memory-based recommender systems. *Eng. Appl. Artif. Intell.* **85**, 347–356 (2019). <https://doi.org/10.1016/j.engappai.2019.06.020>
22. Valcarce, D., Parapar, J., Barreiro, Á.: Item-based relevance modelling of recommendations for getting rid of long tail products. *Knowl.-Based Syst.* **103**, 41–51 (2016). <https://doi.org/10.1016/j.knosys.2016.03.021>
23. Wang, Y., Wang, L., Li, Y., He, D., Chen, W., Liu, T.Y.: A theoretical analysis of NDCG ranking measures. In: Proceedings of the 26th Annual Conference on Learning Theory, COLT 2013. pp. 1–30. JMLR.org (2013)
24. Zhou, T., Kuscsik, Z., Liu, J.G., Medo, M., Wakeling, J.R., Zhang, Y.C.: Solving the apparent diversity-accuracy dilemma of recommender systems. *Proc. Natl. Acad. Sci.* **107**(10), 4511–4515 (2010). <https://doi.org/10.1073/pnas.1000488107>