Exploiting Knowledge Graphs for Auto-Encoding User Ratings in Recommender Systems*

Vito Bellini, Tommaso Di Noia, Eugenio Di Sciascio, and Angelo Schiavone

Polytechnic University of Bari - SisInf Lab, Bari, Italy
firstname.lastname@poliba.it

Abstract. In the last decade, driven also by the availability of an unprecedented computational power and storage capabilities in cloud environments, deep learning emerged as one of the most promising approaches in the generation and training of models that can be applied to a wide variety of application fields. In this paper, we instigate how to exploit the semantic information encoded in a knowledge graph to build connections between units in a Neural Network, thus leading to a semantics-aware autoencoder, SEM-AUTO, able to extract and weight semantic features that can eventually be used to build a recommender system. We tested how our approach behaves in the presence of cold users on the MovieLens 1M dataset and compare results with BPRSLIM.

1 Introduction

Recommender systems (RS) are essential tools that help users to find those items that are relevant to them. Collaborative filtering (CF) approaches are able to provide very accurate recommendations, especially if many data about users-items interactions are available, but they fail when users rate a few items (cold users) or items have a few ratings (cold items).

Among the various deep learning approaches, autoencoders are a particular configuration of neural networks which have recently attracted attention in the RS community. In [9], the authors utilize Denoising Auto-Encoders to learn user-item preferences by reconstructing the input data, i.e. user-item interactions, from its corrupted version, while [8] integrates side information in a CF approach based on Stacked Denoising Autoencoders to alleviate the cold start problem. In [7] the authors compare item-based autoencoding and user-based autoencoding, outperforming all the baselines in terms of RMSE.

A novel idea to tackle the recommendation problem was introduced in [5] where the authors exploit for the first time information extracted from Linked Open Data to improve the recommendation process; afterwards, several works have been proposed, such as [2, 3].

* An extended version of this work has been published in [1].
IIR 2018, May 28-30, 2018, Rome, Italy.

In this paper we present how to to build a user profile by leveraging both autoencoders and semantic information available a knowledge graph, whose structure is further exploited to draw the topology of the underlying neural network. Each neuron in the hidden layer represents a class or a category associated to an item; the resulting neural network is trained to autoencode the ratings of each user. Therefore, every hidden neuron weight is interpreted as the relevance the corresponding feature has for that user. Eventually, the vectors of feature weights are used to compute a top-N recommendation list.

2 Semantics-aware Autoencoders for Recommendation

Artificial Neural Networks (ANNs) are computational models originally proposed to catch underlying relationships in a set of data by using positive and negative examples fed into the network (supervised learning). Roughly, in an autoencoder network one tries to "predict" x from x. The idea is to first compress (encode) the input vector to fit in a smaller feature space, and then try to reconstruct (decode) it back. This means that the model learns in the hidden layers, a representation of the input and therefore a latent representation of the knowledge behind the input data.

Semantics-aware autoencoders Autoencoders, just like other methods for latent representation, are unable to provide an explanation for the latent factors they provide. To address this issue, we propose to give a meaning to connection with the hidden layer and to its neurons by exploiting semantic information explicitly available in knowledge graphs. The main idea of the SEM-AUTO approach is to map connections between units from layer `i` to layer `i+1`, reflecting the nodes available in a knowledge-based graph (KG). In particular, we mapped the autoencoder network topology with the categorical information related to items rated by users. As the nodes in the hidden layer correspond to categories in the knowledge graph, once the model has been trained, the sum of the weights of edges entering a node represents somehow its worthiness in the definition of a rating. If we consider the nodes associated (connected) to a specific item, their weight may be considered as an initial form of explanation for a given rating. Please note, that such autoencoders do not need bias nodes because these latter are not representative of any semantic data in the graph.

Once the network converges we have a latent representation of features associated to a user profile together with their weight. However, very interestingly, this time the features also have an explicit meaning as they are in a one to one mapping with elements (nodes) in a knowledge graph. Our autoencoder is therefore capable of learning for each user the semantics behind her ratings and eventually weight them. Given the trained autoencoder, a user profile is then built by considering the features associated to items she rated in the past.

Recommendation. As we said before, the weight associated to a feature f_n is the summation of the weights w_{jn} computed in the semantic autoencoder for each edge entering the node representing the feature itself. As we train an auto encoder for each user, we have weights changing depending on the original user profile $P(u) = \{\langle i, r \rangle\}$ with i being an item rated by the user and r its associated rating. More formally, we have

$$w(f_n, u) = \sum_{j=0}^{j=inndeg(f_n)} w_{jn}$$

where $inndeg(f_n)$ is the number of edges entering the node representing the feature f_n. Due to the high sparseness of the feature-item matrix, we exploited collaborative information available in the original dataset to further enhance user profiles. We projected them in a Vector Space Model where each feature is a dimension of the vector space and computed the cosine similarity between users and, for each user we computed the set $K(u)$ containing the k users most similar to u. Hence, we can infer the value of missing features for user u with its average from her neighborhood. After this post-processing step, ratings for unknown items $\tilde{i}$ to u can be computed by just sum item's features with the weights from user profile u.

3 Experiments

In this section, we present the experimental evaluations on MovieLens 1M dataset, focused on cold-users with a number of ratings equal to 2 or 5.

	#ratings	**k**	**f1@10**	**precision@10**	**recall@10**	**nDCG@10**	**ERR-IA@10**
BPRSLIM	2	–	0.021741632	0.032649007	0.016297099	0.023353576	0.018825394
SEM-AUTO		10	**0.023096283**	**0.033046358**	**0.017751427**	**0.028283378**	**0.02600731**
BPRSLIM	5	–	0.039078954	0.050066225	0.032046252	0.045158629	0.042121369
SEM-AUTO		100	0.038598535	**0.054039735**	0.030020531	**0.048623943**	**0.047124717**

Table 1: Experimental Results. **#ratings** represents the number of ratings in cold users. **k** is the number of similar users belonging to $K(u)$.

In our experiments, we compared our approach with the implementation of BPRSLIM [6] available in MyMediaLite [4] as baseline. BPRSLIM is a CF state-of-the-art sparse linear method that leverages the objective function as Bayesian personalized ranking. In Table 1 we report only those configurations for which our semantic-autoencoder gets the best results compared to BPRSLIM. We can see that for a number of ratings equal to 2 and 5, we outperform BPRSLIM in terms of precision and nDCG. Our approach gets much better results also in terms of recall and ERR-IA for very cold users, i.e., with only 2 ratings in the profile. As the number of ratings grows, the collaborative component becomes more relevant and BPRSLIM beats our SEM-AUTO approach. It is interesting to note that, depending on the number of ratings in the user profile,the performance in term of accuracy decreases as the number of neighbors increases. As for diversity (ERR-IA@10), in very cold user situations, SEM-AUTO shows to diversify recommendation results better than BPRSLIM.

http://mymedialite.net

4 Conclusion and Future Work

In this paper, we have presented a novel method to design semantics-aware autoencoders (SEM-AUTO) driven by information encoded in knowledge graphs. We compute a latent representation of items and attach an explicit semantics to selected features. This allows our system to exploit the power of deep learning techniques and to have a meaningful and understandable representation of the trained model. We used our approach to autoencode user ratings in a recommendation scenario via the DBpedia knowledge graph and proposed a simple algorithm to compute recommendations based on the semantic features we extract with our autoencoder. Experimental results show that even with a simple approach that just sums the weights associated to features we are able to beat state of the art recommendation algorithms for cold user scenarios.

References

1. Bellini, V., Anelli, V.W., Di Noia, T., Di Sciascio, E.: Auto-encoding user ratings via knowledge graphs in recommendation scenarios. In: Proceedings of the 2Nd Workshop on Deep Learning for Recommender Systems. pp. 60–66. DLRS 2017, ACM, New York, NY, USA (2017)
2. Di Noia, T., Ostuni, V.C., Tomeo, P., Di Sciascio, E.: Sprank: Semantic path-based ranking for top-n recommendations using linked open data. ACM Trans. Intell. Syst. Technol. **8**(1) (Sep 2016)
3. Fernández-Tobías, I., Tomeo, P., Cantador, I., Di Noia, T., Di Sciascio, E.: Accuracy and diversity in cross-domain recommendations for cold-start users with positive-only feedback. In: Proceedings of the 10th ACM Conference on Recommender Systems. RecSys '16, ACM (2016)
4. Gantner, Z., Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: MyMediaLite: A free recommender system library. In: 5th ACM International Conference on Recommender Systems (RecSys 2011) (2011)
5. Heitmann, B., Hayes, C.: C.: Using linked data to build open, collaborative recommender systems. In: In: AAAI Spring Symposium: Linked Data Meets Artificial Intelligence. (2010 (2010)
6. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. pp. 452–461. UAI '09, AUAI Press (2009)
7. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: Autorec: Autoencoders meet collaborative filtering. In: Proceedings of the 24th International Conference on World Wide Web. pp. 111–112. WWW '15 Companion, ACM, New York, NY, USA (2015)
8. Strub, F., Gaudel, R., Mary, J.: Hybrid recommender system based on autoencoders. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. pp. 11–16. DLRS 2016, ACM, New York, NY, USA (2016)
9. Wu, Y., DuBois, C., Zheng, A.X., Ester, M.: Collaborative denoising auto-encoders for top-n recommender systems. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. pp. 153–162. WSDM '16, ACM, New York, NY, USA (2016)