The Impact of Cluster Centroid and Text Review Embeddings on Recommendation Methods

Peter Dolog
Department of Computer Science, Aalborg University
Aalborg–East, Denmark
dolog@cs.aau.dk

Ylli Sadikaj
Faculty of Computer Science and UniVie Doctoral School Computer Science, University of Vienna
Vienna, Austria
ylli.sadikaj@univie.ac.at

Yllka Velaj
Faculty of Computer Science and ds:Univie
University of Vienna
Vienna, Austria
yllka.velaj@univie.ac.at

Andreas Stephan
Faculty of Computer Science and UniVie Doctoral School Computer Science, University of Vienna
Vienna, Austria
andreas.stephan@univie.ac.at

Benjamin Roth
Faculty of Computer Science and Faculty of Philological and Cultural Studies, University of Vienna
Vienna, Austria
benjamin.roth@univie.ac.at

Claudia Plant
Faculty of Computer Science and ds:Univie
University of Vienna
Vienna, Austria
claudia.plant@univie.ac.at

ABSTRACT

Recommendation systems often neglect global patterns that can be provided by clusters of similar items or even additional information such as text. Therefore, we study the impact of integrating clustering embeddings, review embeddings, and their combinations with embeddings obtained by a recommender system. Our work assesses the performance of this approach across various state-of-the-art recommender system algorithms. Our study highlights the improvement of recommendation performance through clustering, particularly evident when combined with review embeddings, and the enhanced performance of neural methods when incorporating review embeddings.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Clustering**; • **Computing methodologies** → *Information extraction.*

KEYWORDS

Clustering, Text Embedding, Recommender Systems

ACM Reference Format:
Peter Dolog, Ylli Sadikaj, Yllka Velaj, Andreas Stephan, Benjamin Roth, and Claudia Plant. 2024. The Impact of Cluster Centroid and Text Review Embeddings on Recommendation Methods. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3589335.3651570

1 INTRODUCTION

The principal idea of any recommender system method is to find items from the recommendation domain that are the most relevant to user interests or preferences. Most recommendation methods map item profiles, users' interests, and preference profiles in a common vector space where a similarity measure can be applied to find associations between them. These methods aim to embed items that are more relevant to the user's interests and preferences closer to a user in such vector space.

Several approaches are proposed in the literature, such as k-nearest-neighbor methods [11], dimensionality reduction methods [10], and different neural network architectures [6] to reduce the dimensionality of items and users embedding vectors. Recently, graph-based methods have been considered, especially on knowledge graphs [20], which connect entities representing knowledge about items and user rating behavior. These approaches exploit the structured information provided by knowledge graphs, which could be more beneficial than using unstructured data.

As for user preference data, the rating history of users serves as valuable data for mining information about users' interests. Recently, there has been a growing interest in using review text data as a valuable source for understanding user preferences. This data source has the potential to complement the information derived from rating histories. We hypothesize that the insights gained from analyzing review text may be even more robust or enhance the patterns observed in rating behavior.

The idea of grouping related data points in a vector space can also be found in the clustering domain. Clustering algorithms are also often based on mapping objects to a vector space where a similarity measure can be used to group together relevant data points. The difference is that clustering algorithms are mostly unsupervised, [13], while recommender system methods require labeled data.

We investigate the impact of clustering, review embeddings, and their combinations on recommender systems. To this end, we propose **a methodology for combining clustering and review embeddings with embeddings obtained from a recommender system**. We study the performance of this methodology on different representative state-of-the-art recommender system algorithms. In addition, we provide the following contributions and findings: a) A graph clustering algorithm, i.e., SpectralMix [13] is, for the first time, adapted for recommendations with two additional variants of objective functions: DOT product and multiplication between

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05.
https://doi.org/10.1145/3589335.3651570

DOT product and Euclidean Distance; b) Embeddings from clustering improve the recommendation performance on embeddings obtained from non-neural methods and neural methods when considered together with review embeddings; c) Integrating review embeddings into embeddings obtained by all methods improves the recommendation performance; d) Integrating embeddings from clustering decreases the performance of the collaborative filtering method LightGCN [6].

2 RELATED WORK

Previous work on clustering and review text analysis in the context of recommender systems focuses mostly on specific methods and usually provides either a view on clustering and recommendation or review text analysis and recommendation.

[16] proposes to recommend items a user has not seen yet and purchased by users from the same cluster where a user is assigned to. Several clustering algorithms are compared there, but the impact on the state-of-the-art recommendation methods is not studied. Clustering of social tagging data showed scalability and performance benefits in tensor-based recommender [7] or in the context-dependent variant of hierarchical clustering-based recommender [14]. Recently, a multi-clustering approach was applied in a fashion recommender system based on neighborhood approaches [1].

Hybrid recommenders that consider review text have also been studied. [3] surveys the approaches for review text-based recommenders. [4] studies selected text embedding methods in connection to recommenders by measuring mean squared error. Text representation learning methods have been applied in multi-modal recommender systems [18] but only on categorical, title, and text descriptions of items. In both works, the focus is on adding information from the text to recommender systems. In our work, we study in detail the impact of both clustering and recommendation. We use sentence embeddings instead of word embeddings to investigate the impact of the text reviews. Due to space limitations, we show our study on simple text review embedding aggregations for users and items to show the impact on recommendation accuracy.

Overall, in this paper, we provide a unique analysis of the impact of clustering, review text embeddings, and their combination on different selected state-of-the-art methods for recommendation systems. We consider representative methods from the neighborhood-based class, collaborative filtering class with dimensionality reduction, graph convolutions, graph-based methods considering contrastive learning, and spectral analysis considering homogeneous and heterogeneous graphs.

3 METHODOLOGY

Methods Selection

We have selected the following representative methods, including statistical, deep learning and spectral clustering based approaches:

KGCL [20]: A Knowledge Graph (KG) based method with a contrastive learning component bringing a de-noising component which overcomes noise and sparsity in knowledge graphs.

LightGCN [6]: A collaborative filtering method where user and item embeddings are computed in convolutions over a collaborative rating graph. It belongs to a group of neural graph convolutional techniques with a minimal set of components in its neural network architecture. In various experiments, it performed better than more complex neural collaborative filtering algorithms.

BPRMF [10]: A representative matrix factorization method that produces user and item vectors with significantly reduced dimensionality. It learns the most representative features of items and user profiles from implicit rating data.

UserKNN [11] (KNN): A neighborhood-based collaborative filtering method exploiting information from co-rated items or users who co-rated items. N most similar user or item rating vectors are considered to predict an item's recommendation/rating score. It is a simple yet well-performing method in many cases. Since KNN is not an embedding-based method, we are not integrating it with review or cluster embedding.

Spectral Clustering Based Recommendation [17]: Produces embedding as an eigenvector decomposition of the laplacian matrix of the knowledge graph enhanced with user-item interaction (rating relation added). This is similar to how other matrix factorization methods work on rating data. The difference is that here, we consider rating interaction, knowledge graph relations, and eigenvector decomposition of a matrix constructed from the collaborative knowledge graph. In simple spectral clustering, we do not consider relation and node types. For recommendation purposes, we select embeddings of user and item nodes to perform DOT product or Euclidean Distance between them.

SpectralMix Based Recommendation [13]: It is an unsupervised embedding approach that enhances spectral clustering by utilizing information from multiple types of edges and nodes. For recommendation purposes, we select user and item embeddings to perform prediction with DOT product or Euclidean Distance. Euclidean distance is originally used as an objective function for learning in SpectralMix. We add the DOT product and the multiplication of the DOT product with Euclidean distance objective functions.

Review Based Recommendation: A heuristic method designed to investigate the impact of a single review-based embedding component in the recommendation. We construct an item-related review embedding as a mean aggregate of review text embeddings that are linked to that item. Similarly, to obtain a user-related embedding, we aggregate (mean) item-related review embeddings from all users related to the items the particular user has rated. Thus, we obtain a single embedding for each user and each item, respectively, as an input to calculate the prediction, as both DOT product and Euclidean distance between them. We utilize the SentenceTransformer model [9] for review text embeddings.

Learning

SpectralMix is designed for multi-relational graphs, and we construct different relations between users and items. Also, we consider two versions of ratings to obtain embeddings. The first version learns only from movie ratings, and the second version learns from all ratings, including ratings on entities in the knowledge graph. In the other recommendation methods, we utilize their learning methods without modification, as mentioned by their authors.

Prediction

Each method, except KNN, produces embeddings of users and items. We test for two prediction methods: 1) **DOT product** of user and item embedding vectors as it is the standard method used for prediction in all state-of-the-art recommendation methods, and 2) **Euclidean distance** of user and item embedding vectors because it is used for the objective function in the SpectralMix. In the case of the KNN, we apply the standard weighted rating average of neighbors for prediction.

Integration of Cluster Centroid and Review Embeddings

Cluster Centroid Embeddings. Our aim is to investigate how helpful the cluster centroid embeddings are. We apply the K-Means algorithm to the embeddings obtained from each recommendation method, producing cluster centroids of similar users and items. We discuss in section 4 regarding the number of clusters parameter for the K-Means algorithm. We include two versions of cluster centroid embeddings integration: 1) By concatenating user and item embeddings with the cluster centroid embeddings for a cluster where user and item belong to, respectively; 2) By computing DOT product between user and item embeddings and a cluster centroid embeddings of a cluster where user and item belong to, respectively. These enhanced user and item embeddings are used for prediction in the same way as described above (DOT product or Euclidean distance).

Text Review Embeddings. We construct text (sentence) embeddings of the reviews using the widely used SentenceTransformer model [9][1]. Each item has up to N reviews. We compute an aggregated (mean) embedding of reviews for each item to integrate it into an item embedding. We concatenate such review embeddings to the corresponding item embeddings. Similarly, we aggregate (mean) review embeddings of items that a user has rated. After the aggregation, we concatenate such aggregated review embeddings with user embeddings. Ideally, we would want to have reviews written by users from our repository, as in [4]. Unfortunately, this data is not available, and thus, we apply the above-mentioned heuristic.

Integration. We also combine embeddings of cluster centroids, reviews, and recommendations. We consider two options: 1) We concatenate review embeddings with user and item embeddings enhanced with cluster centroid embeddings (cluster centroid embeddings are obtained before review embeddings are concatenated with user and item embeddings); 2) We cluster user and item embeddings enhanced with review embedding (cluster centroid embeddings are obtained after the review embedding is concatenated with user and item embedding).

4 EXPERIMENTAL RESULTS

Datasets. We evaluate our solutions on the Mindreader dataset [2][2]. The Mindreader dataset features a knowledge graph in the movie domain and contains user ratings for movies and entities in the knowledge graph. It also contains explicit ratings for expressing an undecided preference and a negative rating. The dataset contains 218, 794 ratings from 2, 316 users, over 12, 206 entities, and an associated knowledge graph of 18, 133 movie-related entities. The knowledge graph is used for KGCL, Spectral Clustering, and SpectralMix. For KGCL, only positive ratings of movies are used since it cannot deal with other ratings. For SpectralMix, a full set of ratings is used to obtain embeddings, but only positive ratings for the prediction are considered, and 8 different relations between movies are constructed, such that two movies are connected if they share the same director, producer, genre, etc. For LightGCN, only the user-movie bipartite graph is used. For KNN and BPRMF, only the rating matrix derived from positive ratings of users for movies is used. We consider items and users that appear at least once in the testing fold for a particular fold of evaluation. In addition to the Mindreader dataset, we use a movie review dataset [8]. This dataset contains 50, 000 English movie reviews along with their associated sentiment labels "positive" and "negative". The link between movies and reviews is obtained through the same movie ID in both datasets.

Number of Clusters. We used the Silhouette score [12] and Elbow analysis [15] to find an optimal number of clusters on the Mindreader dataset. The analysis suggested that the optimal number of clusters should be 3, 4, and 5. Additionally, we include $K = 2$ clusters to see how the performance changes, and we keep the same number of clusters for all considered recommendation methods.

Metrics. We perform Top K recommendations evaluation on 5 folds and utilize LensKIT [5] for our evaluations. We adopt standard Top K recommendation metrics provided by LensKIT. **Normalized Discounted Cumulative Gain (NDCG)@K** computes a mean utility score of ranked lists of recommendations of the size K. It values more if the relevant items are in the upper part of the list. **Hit@K** computes the fraction of the lists of the size K with at least one relevant item for a user appearing in a list generated for the user. **Recall@K** computes a mean recall of ranked lists of size K. **Reciprocal Rank@K** computes the reciprocal rank of the first relevant item in the list of recommendation with size K. We evaluate our methods on ranked lists of 5, 10, 50, and 100 items.

Results. Figure 1 shows experimental results for selected methods for varying sizes of recommendation lists ($k = 5, 10, 50, and 100$). We have studied all mentioned combinations of methods. Due to space limitations, we selected only the 10 best-performing ones. In the names of methods, we use the following abbreviations: 1) **DP – DOT Product** ; 2) **ED – Euclidean Distance**; 3) **DP KC – K is a number and C means clusters, meaning the number of clusters**; 4) **CC – Concatenating cluster centroids**; 5) **DC – DOT product with cluster centroids**; 6) **CR – Concatenating Reviews**; 7) **OF-Method – Objective function in SpectralMix with DP, ED, or DP x ED**.

Adding the information provided by review embeddings improved all the methods, and the improvement is rather large. We can also note that the 10 best-performing results include the heuristic method based only on review embedding. That alone says that reviews are a valuable source of information and often even better than some established methods, such as KNN or some combinations of SpectralMix with clustering and review embedding. This is natural since the reviews contain rich information about user preferences, and when connected with ratings, it can further amplify the user preference insights.

The impact of clustering depends on the method. In the LightGCN, the clustering applied on embedding from collaborative rating makes the performance worse. KGCL performance is improved by integrating the text review embeddings, but the integration of the clustering makes it worse. It seems that clustering on embedding

[1]Model available in the Transformers library[19] at https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.
[2]Available at: https://mindreader.tech/dataset/

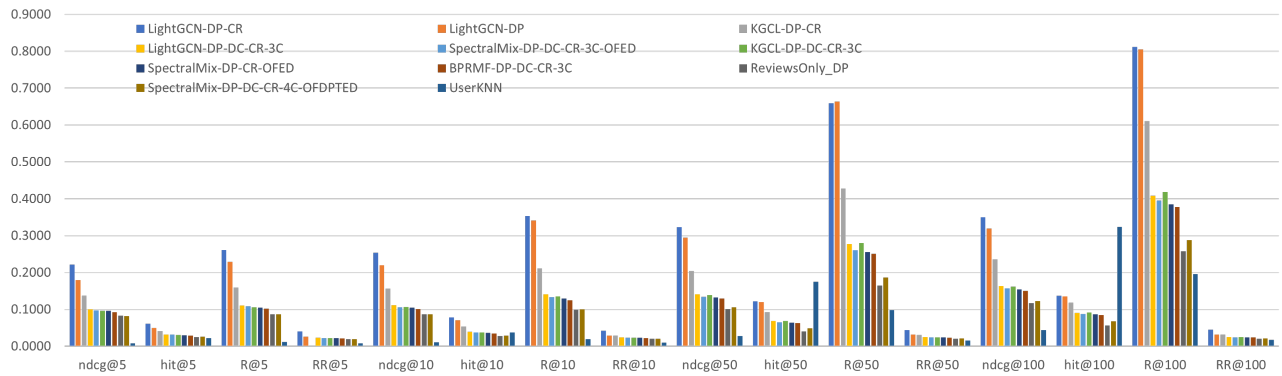

Figure 1: Performance for selected method and their integration with clustering, review embedding, and their combination. Abbreviations: 1) DP – DOT Product; 2) ED – Euclidean Distance; 3) DP KC – K is a number, and C means clusters, meaning the number of clusters; 4) CC – Concatenating cluster centroids; 5) DC – DOT product with cluster centroids; 6) CR – Concatenating Reviews; 7) OF-Method – Objective function in SpectralMix with DP, ED, or DP x ED. Example: LightGCN-DP-DC-CR-3C means LightGCN recommendation method with dot product prediction function (DP) with cluster centroids embeddings integrated by dot product (DC), concatenated review embeddings (CR), with 3 clusters (3C).

from collaborative information can not help to catch hidden information but rather introduce contradictory information into the recommendation process. Methods based on graph cuts or non-neural methods, such as BPRMF, benefit from text review embeddings. Additionally, clustering on integrated information from collaborative, knowledge-based, and review sources provides the best performance. Thus, clustering helps to reduce the noise in those methods.

The collaborative information has by far the largest influence, as shown by the best-performing method, LighGCN. The LightGCN considers a bipartite graph of user ratings on movies. Even the original prediction method without integration with review embeddings achieves the second-best performance, except when overperformed by the KGCL on the Reciprocal Rank metric on larger lists and the KNN on the HIT metric on larger lists. The high dimensional rating vectors and the prediction of the weighted rating average are capable of satisfying more users with at least one relevant recommendation.

5 CONCLUSIONS AND FUTURE WORK

We provided a study on the impact of clustering and review embedding on the performance of selected state-of-the-art recommendation methods. We have shown that clustering and review embeddings both positively impact performance. Moreover, the collaborative signal with the review embeddings of the LightGCN method is superior to the others except for HIT@50 and HIT@100 metrics, where the KNN overperforms more complex methods.

For future work, we will include more datasets to investigate whether the impact is similar across different domains. Different methods that consider the sparsity of the knowledge graph are also interesting to study to improve the state-of-the-art.

ACKNOWLEDGMENTS

This work is supported by a grant from the Carlsberg Foundation by grant agreement nr. CF21-0073.

REFERENCES

[1] P. Bellini, L. A. I. Palesi, P. Nesi, and G. Pantaleo. Multi clustering recommendation system for fashion retail. *Multim. Tools Appl.*, 82(7):9989–10016, 2023.
[2] A. H. Brams, A. L. Jakobsen, T. E. Jendal, M. Lissandrini, P. Dolog, and K. Hose. Mindreader: Recommendation over knowledge graph entities with explicit user ratings. In *ACM CIKM 2020*, page 2975–2982. ACM, 2020.
[3] L. Chen, G. Chen, and F. Wang. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, 25(2):99–154, 2015.
[4] P. A. B. Dezfouli, S. Momtazi, and M. Dehghan. Deep neural review text interaction for recommendation systems. *Applied Soft Computing*, 100, 2021.
[5] M. D. Ekstrand. Lenskit for python: Next-generation software for recommender systems experiments. In *ACM CIKM 2020*, Oct 2020.
[6] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *ACM SIGIR 2020*, page 639–648, 2020.
[7] M. Leginus, P. Dolog, and V. Zemaitis. Improving tensor based recommenders with clustering. In *UMAP 2012*, pages 151–163. Springer, 2012.
[8] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *ACL 2011*, pages 142–150. ACL, jun 2011.
[9] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP 2019*. ACL, 11 2019.
[10] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI 2009*, page 452–461, 2009.
[11] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *ACM CSCW 1994*, 1994.
[12] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20:53–65, 1987.
[13] Y. Sadikaj, Y. Velaj, S. Behzadi, and C. Plant. Spectral clustering of attributed multi-relational graphs. In *ACM SIGKDD 2021*, page 1431–1440, 2021.
[14] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *ACM RecSys 2008*, page 259–266, 2008.
[15] R. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.
[16] C.-F. Tsai and C. Hung. Cluster ensembles in collaborative filtering recommendation. *Applied Soft Computing*, 12(4):1417–1425, 2012.
[17] U. von Luxburg. A tutorial on spectral clustering. *Stat. Comput.*, 17(4):395–416, 2007.
[18] W. Wei, C. Huang, L. Xia, and C. Zhang. Multi-modal self-supervised learning for recommendation. In *ACM WWW 2023*, pages 790–800, 2023.
[19] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-art natural language processing. In *EMNLP 2020: System Demonstrations*, pages 38–45, Online, Oct. 2020. ACL.
[20] Y. Yang, C. Huang, L. Xia, and C. Li. Knowledge graph contrastive learning for recommendation. In *ACM SIGIR 2022*, page 1434–1443, 2022.