SI 650 Project Proposal Oct-12-2022

Our team is composed of 2 people: Yeeun Jang and Jiangyue Mao. The goal of our project is to construct a joke recommendation system that can recommend jokes to users based on the similarity among users' ratings using collaborative filtering and other advanced techniques such as Neural collaborative filtering. Since the jokes are in text format, we are also considering using word vectors. We are planning to split the dataset into training and testing and measure our success by evaluating the system on the test set using NDCG and hit rate. We think the workload is appropriate for 2 people because we incorporate both traditional and advanced recommendation engine construction methods, and it will take a fair amount of time for us to learn different collaborative filtering methods and even take deep learning neural networks into account. The dataset description and the methods that we are planning to use are proposed below.

The dataset we are going to use is from https://eigentaste.berkeley.edu/dataset/ dataset 3, which contains 150 jokes and around 2.3 million ratings. There are two separate CSV files, with the first one being the content (text) of the 150 jokes and the second one being the ratings. The rows of the second file represent users, and the columns represent jokes' IDs. 22 of these jokes have few ratings as they were removed as of May 2009 and deemed to be out of date. Their ids are: 1, 2, 3, 4, 5, 6, 9, 10, 11, 12, 14, 20, 27, 31, 43, 51, 52, 61, 73, 80, 100, 116. Each rating is from -10.00 to +10.00 and 99 corresponds to a null rating (the user did not rate that joke). If some users did not rate most of the jokes, we might need to remove those users to minimize their impact on the recommendation system.

Although there are a few recommender online that's using the same dataset, the conclusions that the authors drew are not very satisfying since most of them are using basic collaborative filtering techniques and/or even only picking the top 10 jokes with the overall highest ratings and recommending them to everyone, so the recommender is not customized on users' information. Therefore, we believe that using both collaborative filtering and advanced techniques involving neural networks will help the recommendation system become more developed and well-rounded. In addition, since these methods haven't been covered in class, we believe learning these techniques will also contribute to our personal data science development.

Moreover, we believe this recommendation system is very helpful for people who would love to enrich their lives by listening to more jokes and non-native English speakers and foreigners who are often asked to tell jokes in team-related events while working. Both authors of this project are non-native speakers and have encountered that situation during the internships, and we believe such recommender systems can greatly relieve our pressure of not understanding native jokes and/or thinking of a new joke that everybody can easily understand.

In recommendation systems, user-item interaction is often treated as key information. User-item interaction is the act of users consuming items. ‘Consume’ includes purchase, rating, watch, and click. Implicit feedback is data that implicitly shows user preferences. Examples of implicit feedback include click logs and purchases. On the other hand, explicit feedback is data that explicitly shows user preferences, such as scores and ratings. Our dataset contains user ratings of the jokes, which are items. A number of research studies in recommendation systems leverage user-item interaction data, adding some auxiliary information. Therefore, the dataset we are using is appropriate and reasonable for recommendation systems and its implementation and application will be feasible. We will continue to do research on related topics, and ask for feedback and help as much as we can for improvement.

Our proposed timeline is as below:

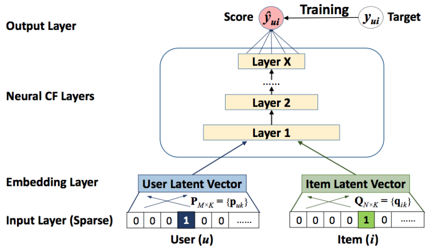
EDA	collaborative filtering	Result evaluation and model revision	Neural collaborative filtering / DeepFM + evaluation	Final report	
10/12 - 10/20	10/21 - 11/5	(around) 11/6 - 11/14	11/15-12/9	12/10-12/14	

The biggest challenge for our project would be to capture the underlying features of the user's preference as much as possible. In our dataset, an item corresponds to a joke and a user corresponds to a user who rated a joke. However, unlike normal items like books, videos, and movies, jokes are mostly in long text format. There would be multiple approaches possible to leverage the text data itself, and the most straightforward approach would be mapping each word into word vectors, using pre-trained word embeddings such as word2vec. However, in this case the corresponding vectors will be in high dimensions, which is auxiliary information. Adding auxiliary information in recommendation systems will be challenging to handle, as it depends on the dataset and the structure of the model we are going to use. Another challenge is how to map the words. Joke often has implicit meaning, which is hard to catch when it is simply embedded into word vectors. When a joke is splitted into words, it often loses its meaning. Similarly, the similarity among jokes does not heavily rely on the meanings of the words itself. Therefore, it must be challenging to find the best way to leverage the joke as a text.

For the methods that we are trying to use, first of all, we will try collaborative filtering as a baseline approach. Collaborative Filtering is a recommender system based on user's preference information. Matrix Factorization(MF)[1] is the most popular implementation. It captures the pattern between users and items by mapping both users and items to the integrated latent space, where user-item interaction is internally modeled. In the following steps, we will try to apply more advanced methods. As of now, we are planning to try deep neural networks based models such as Neural collaborative filtering or DeepFM.

The structure of Neural Collaborative Filtering(NCF)[2] is a combination of two key features - Generalized Matrix Factorization(GMF) and Multi-layer perceptron (MLP), as shown in figure 1. MLP can comparatively express more complex relationships than matrix factorization-based structures because it is non-linear and flexible. NCF is a remarkable approach that can learn both linear and non-linear representation. However, it also has some limitations that it is not using auxiliary information, and is not able to deal with the cold-start problem.

The second model we are planning to use, DeepFM[3], is a model that predicts Click-Through-Rate(CTR) by combining Factorization Machine(FM) structure and Multi-Layer-Perceptron(MLP) structure. FM component captures low-order feature interaction and MLP component captures high-order feature interaction, as shown graphically on figure 2. However, DeepFM has the same limitations as NCF.



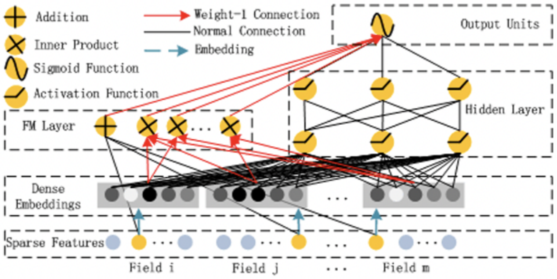


Figure 1. Structure of NCF

Figure 2. Structure of DeepFM

For evaluation, we will use Normalized Discounted Cumulative gain (NDCG) and hit-rate. (include definition / formula)

Normalized Discounted Cumulative Gain(NDCG) is defined as :

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p},$$

where DCG stands for Discounted cumulative gain, and IDCG stands for Ideal Discounted Cumulative Gain.

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \quad \mathrm{IDCG_p} = \sum_{i=1}^{|REL_p|} \frac{rel_i}{\log_2(i+1)}$$

Hit ratio@n is defined as:

$$HR = \frac{hits}{hits + misses}$$

where n indicates the total number of misses and hits we retrieve. (n = hits + misses)

In conclusion, we will try collaborative filtering and other advanced techniques such as Neural collaborative filtering to build the recommendation system and use NDCG and hit-rate as our evaluation metrics. The end goal of this project is to build a successful recommendation machine that can recommend jokes based on individual user's rating habits and we will make sure to consult professors, GSIs, peers, and online resources such as papers and blogs for help if necessary.

References

[1] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender
systems." Computer 42.8 (2009): 30-37.
[2] He, Xiangnan, et al. "Neural collaborative filtering." Proceedings of the 26th international conference on
world wide web. 2017.
[3] Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." arXiv
preprint arXiv:1703.04247 (2017).