Survey on selected Developments of Recommender Systems

Lilac Lai (Yinmeng2@illinois.edu)

Recommender systems are used widely throughout the internet, with applications that many are familiar with such as recommending the right YouTube videos, Facebook Ads, Netflix movies, and many other online shopping items. Good recommendations benefit both the business or advertisers and consumers. Recommender systems' aims are to make predictions about a user's preference or rating of unseen items and then recommend the ones that are most likely to be the ones users are interested in.

Popular recommender system approaches typically fall into the category of Content-based approaches, which build individual user profiles independently, and Collaborative Filtering approaches, which consider all users and items together. Classic recommender system includes ones that we have learned in class such as Matrix Factorization and Neighborhood Collaborative Filtering.

Recommender systems face a few general problems: the sparsity problem, the cold-start problem, the scalability problem, and the rating bias problem. The development of recommender systems typically falls into the category of solving one of the above problems and increasing recommendation effectiveness. The recent development of recommendation systems leverages the power of machine learning and more complex user-item interactions for improvements. The machine learning techniques used include the use of neuro networks, deep learning, stacked autoencoders, and transfer learning. The user-item interactions are traditionally represented by a single numeric rating, and recent advancements include the increased use of multi-criteria user ratings which could include explicit user ratings or implicit ratings extracted from user-generated texts.

In the following discussion, we will be examining some of the state-of-the-art progress in recommender systems.

EDMF: Efficient Deep Matrix Factorization With Review Feature Learning for Industrial Recommender System [1]

There has been much research done on how to improve the performance of Collaborative Filtering recommendation systems. One approach uses user-generated text reviews on top of the classic numeric rating that summarizes a user's overall opinion of an item. Methods to extract user preferences from text reviews can be classified into two main categories— convolutional neural network (CNN) models and bags of words models. While CNN models achieved higher accuracy, their training time and parameter complexity are very high. Thus, the high time complexity and space complexity of the CNN models makes them unsuitable for industrial use. Matrix Factorization (MF) is a classic and popular collaborative filtering-based approach to approach recommendation problems. Efficient Deep Matrix Factorization (EDMF) is the new tool this paper introduces, which builds on top of the Matrix Factorization technique and CNN.

The EDMF framework utilizes the interactive characteristics of user text reviews while being able to achieve higher training efficiency and good accuracy compared. Big picture-wise, EDMF extracts the review features from users to produce a onefold review feature representation vector and considers user-items matrix interactivity and sparsity as priors to the maximum a posteriori (MAP) of latent factors to use in combination with MF techniques.

The EDMF tool can be summarized as three main steps: review feature representation learning, latent factor representation learning, and rating prediction based on sparsity constraints. Review feature representation learning is where we extract the information from user text reviews and represent them in a feature vector. EMDF achieves this by using a CNN with a word attention mechanism to create a feature vector. In latent factor representation learning, a Hadmard product is taken of the users' and items' latent feature vectors, and the Hadmard product is partially constrained by the review text feature vector from the previous step. In constructing the latent factor pool, the latent factors of users and items are assumed to follow a Gaussian distribution that follows a mean of zero and variances of the training dataset variance. A MAP estimation of the users' and items' latent factors is then calculated. Finally, EDMF performs rating prediction based on spasticity constraints by representing review features with an L0 norm prior probability and includes parameters that act as spasticity constraints of the L0 norm. The authors also propose a further improvement to EDMF, calling it EDMF+, which considers the length of the review and includes a log of long term in the original MAP estimation setup.

The performance of EDMF is evaluated against other CNN-based recommendation systems, such as CARL and DeepConNN. The author ran the evaluations on four public datasets that are on automotive, movies, video games, and Yelp user-items interaction. In terms of training accuracy, EDMF achieves better accuracy than all of the five other CNN-based recommendation systems with lower root mean square error (RMSE) and mean absolute error (MAE). In terms of training time, EDMF also made huge improvements again the other CNN-based datasets, with an improvement in training time decrease of as high as 71.7%.

Utility-based multi-criteria recommender systems [2]

One category of multi-criteria recommender systems is that of using explicit user preference. Explicit user preference means that users are asked to explicitly give numerical ratings to multiple characteristics of an item– for example, users may be asked to rate a restaurant based on the tastiness of the food, environment, quality of service, etc. Zheng's paper proposes one explicit user preference multi-criteria recommender system, which predicts user ratings by building a utility function and then ranking items based on their calculated utility.

A utility function is trained and built for each unique user. It is defined as the difference between the vector of the user rating vector that is calculated from the multi-criteria ratings and the vector of user expectations that contains the expected values for each item's rating based on the training data. Zheng used three different ways to evaluate utility: the difference between the rating and expectation vector, which are Pearson correlation, cosine similarity, and Euclidean distance. After building the utility function, utility scores for each item are calculated, and a ranking of items is created for each user using the optimization idea of Learning-to-rank. Zheng also used multiple ways to perform the ranking calculation: Pointwise ranking, Pairwise Ranking, and Listwise Ranking.

The utility-based recommender is evaluated with each of the three similarity evaluation methods and each of the three ranking calculations against other recommendation systems, including the classic Matrix Factorization and three other models, such as the Hybrid Context Model (HCM). The evaluations were done using two datasets from TripAdvisor and Yahoo! Movies. The effectiveness of the models is evaluated in terms of precision and the normalized discounted cumulative gain (NDCG). The results of utility-based recommender outperform the baselines in terms of both metrics. Based on this evaluation, the utility calculated using the Pearson correlation measure and using Listwise ranking produced the best performance.

User preference learning in multi-criteria recommendations using stacked autoencoders [3]

Other approaches to improve recommendations using explicit user preferences include those involving machine learning and neural networks. The authors of this paper propose a Collaborative Filtering recommendation method using stacked autoencoders to produce rating predictions for all unknown user-item pairs. Autoencoders learn a compact representation of the input data. They are known for their applications in feature learning, dimensionality reduction, and denoising, which are all goals in the Collaborative Filtering recommendation problem. Due to the increase of dimensionality data when using multiple-criteria user preferences compared to single rating recommender systems, the author chooses to use stacked autoencoders which is especially great for finding lower dimensional representations.

In more detail, the authors used stacked autoencoders and added extra input and output layers to deal with the multi-criteria user ratings. The model is named Extended Stacked Autoencoders (Extended_SAE). The structure of this stacked autoencoder can be described as: one input layer that maps multi-criteria input to each item neuron, followed by N encoding layers to encode the latent representation, the N decoder layers to decode the previously obtained latent factors, and finally, an output layer to predict all of the ratings. The loss function of this stacked autoencoder network is defined as the summed square error of known items rating. In this design, items without a rating are "deactivated" and not considered in the loss function.

The evaluation of these stacked autoencoders (Extended_SAE) is also done on the TripAdvisor and Yahoo! Movies datasets. Extended_SAE with N = 3 (3 encoder layer, 3 decoder layers) and N = 5 are re-evaluated with the classic Matrix Factorization and six more advanced and recent recommenders from 2011, 2016, and 2017. Out of all of the datasets evaluated, Extended_SAE models outperformed all of the other models in terms of all of the accuracy matrix– Mean Absolute Error (MAE), the F1 metric (a metric commonly used to evaluate collaborative filtering precision), Good Items MAE (GIMAE), and Good Predicted Items MAE (GPIMAE).

DNNRec: A novel deep learning-based hybrid recommender system [4]

Deep Neuro Networks (DNN) are also used to improve other aspects of recommender systems, such as improving the "cold-start" problem. Due to the linear nature of many recommender systems– such as the classic Matrix Factorization (MF)-- there is limited information that can be extracted by the model. However, DNN can learn non-linear characteristics due to their multiple layers architecture and can be further enhanced by choosing non-linear activation functions such as ReLU. Thus, by using a deep neural network, the author hopes to learn non-linear latent factors and thus improve overall prediction accuracy, including those in the cold-start problems category. The model proposes, DNNRec, is a hybrid recommender model as it can capture not only user-items interactions but also each user's preferences like those seen in Content-based recommendation approaches.

DNNRec is designed to learn from specially crafted users and item embeddings during an intense network and uses decreasing learning rate with increasing weight decay. The network uses a non-linear activation function called the LeakyRELU, stochastic gradient descent with restarts, and weight decay concepts to optimize the solution. The authors tuned the hyperparameters of leaning rate, weight decay, and dropout.

Instead of the typical representation of user-item ratings in a single matrix where the matrix tends to be sparse, the authors propose the representation of users and items as embeddings. Embeggings are lookup matrices that are a size where each row represents the embedding array of each particular user (or item) and is size K, where the raw user (or item) data of that particular user (or item) is mapped into this vector space of the fixed size K. The use of embeddings has been shown to be successful in both natural language modeling approaches and in YouTube recommendations.

After training the network, the network's performance is evaluated against ten other rating prediction methods using the classic accuracy measurements, which are MSE, RMSE, MAE, and R-squared on five different datasets. The result of DNNRec is promising, with better performance than all the datasets' baselines. What is especially notable is DNNRec's prediction performance on cold-start cases. On all of the datasets, DNNRec's prediction for cold cases outperformed the following best techniques by as much as 64% improvement in R-squared. The performance improvement between DNNRec and the next-best technique is typically between 2%-12% in terms of MSE, RMSE, MAE, and R-Squared.

Transfer Learning via Contextual Invariants for One-to-Many Cross-Domain Recommendation [5]

Other improvements or recommendation systems include those that address the sparsity problem. In the Transfer Learning paper by Krishnan et al, the authors aim to address the spasticity problem by using domain-invariant components across dense and sparse domains. This paper is in the category of context-aware recommendation and produces a Neuro Collaborative

Filtering technique. Compared to past methods that have attempted to use cross-domain methods to improve inference quality, the technique proposed made new advancements, including allowing one-to-many transfer between dense and sparse domains. Furthermore, this technique does not require overlaps of user and item information. Another benefit of this research is its scalability– which the authors demonstrated through experiments run on massive datasets. The authors can achieve the improvement of user-item representation through the use of contextual invariances across domains, extracting context features using context combination layers. Big picture-wise, multi-linear contextual invariances are extracted from dense source domains and applied to the sparse domains. The transformation to the sparse domains is achieved by direct layer transfer and two different approaches of layer adaptation– Annealing (deals with parameter adaptation) and Distributionally Regularized Reisiduals (deals with input adaptation). The overall recommender architecture includes the user and item embeddings, a context module, reclustering module of pooled context and embeddings, and a ranking module. The context modules, reclustering module, and ranking module can all be transferred from the source dense domain to the target sparse domain.

The model is evaluated on two datasets, Google Local Review Dataset and Yelp Challenge Dataset. Both datasets consist of explicit user feedback. The authors pre-processed the data by extracting only user reviews of restaurants based on geographic locations. This approach creates dense and sparse regions in the dataset and is known to have the context invariant of location. Then, the models are trained on high-density regions and transferred to lower-density regions. This approach is shown to be effective and efficient. This approach is implemented in three models– the main Multi-Linear Module Transfer Network, a variant with feedforward ReLU layers, and a variant with information-gain terms included. The three models' single-domain recommendation performance is evaluated with seven state-of-the-art recommendation models. Multi-Linear Module Transfer Network models pretty consistently outperform the other state-of-the-art models in terms of RMSE and MAE with only a few close exceptions.

Research opportunities and challenges

From the survey of research papers above, I have learned about the recent advancements in recommender systems and gained knowledge on future research directions on recommender systems. Although already a area under research, sentiment analysis could be an approach to enhance recommendations. Accurately extracting user opinions on the characteristics of items would provide even more information-rich user-items interaction than the current multi-criteria rating systems.

Other research directions could be tackling the cold-start problem. More user-items interaction features or outside information (such as user location) could potentially be used to improve predictions during a cold-start problem. For instance, the research could be done on using third-party cookies data to aid the recommendations during the cold-start phase. Another potential approach to improve cold-start recommendations could include techniques such as knowledge-based approaches.

One current limitation and challenge to recommender systems include computing power–effective but costly models are not widely used in the industry.

References

[1] H. Liu, et al., "EDMF: Efficient Deep Matrix Factorization With Review Feature Learning for Industrial Recommender System," in IEEE Transactions on Industrial Informatics, vol. 18, no. 7, pp. 4361-4371, July 2022, doi: 10.1109/TII.2021.3128240.

[2] Y. Zheng, "Utility-based multi-criteria recommender systems," in Proc. 34th ACM/SIGAPP Symp. Appl. Comput., 2019, pp. 2529-2531.

[3] D. Tallapally, R. S. Sreepada, B. K. Patra, and K. S. Babu, "User preference learning in multi-criteria recommendations using stacked autoencoders," in Proc. 12th ACM Conf. Recommender Syst., Vancouver, BC, Canada, 2018.

[4] K. R, P. Kumar, and B. Bhasker, "DNNRec: A novel deep learning based hybrid recommender system," Expert Systems with Applications, vol. 144, 2020, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2019.113054.

[5] A. Krishnan, M. Das, M. Bendre, H. Yang, and H. Sundaram, "Transfer Learning via Contextual Invariants for One-to-Many Cross-Domain Recommendation," in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), Association for Computing Machinery, New York, NY, USA, 2020, pp. 1081-1090.