**The Recommender System: Power of Personalized Online Experience**

There’s no doubt about it. We live in a data driven world. To be competitive in today’s world, a business must have digital access to its consumers via computer and hand-held devices. Through the collection of information regarding the consumer’s behavior, businesses can now provide a more personalized, targeted experience for its customers. It can utilize this information to measure performance, enhance marketing efforts and drive business decisions. One of the methodologies used for targeting specific users with relevant material such as a recommended purchase, subscription suggestion, or online advertisement is the recommender system.

Once we know we want to implement a recommender system, we must decide upon the approach. Following is a brief overview [2]. The two approaches for designing a recommender system are content-based filtering and collaborative filtering. The content-based approach requires specific knowledge of the features of an item and uses that information to recommend items with similar features. Collaborative filtering does not require prior knowledge of the features of an item. Collaborative filtering builds a model based on users’ past preferences and uses that model to predict future preferences. There are two types of collaborative filtering algorithms: memory-based and model-based. The memory-based algorithms find a group of users that have similar preferences as the user in question and use those preferences to make a prediction to the user. A similarity measurement is used to find a correlation between users. The model-based algorithm uses a subset of the data as “training data” to learn a predictive model. There are advantages and disadvantages to the various approaches and algorithms. Some advantages of model-based collaborative filtering are [7]:

1. No prior knowledge of the item or user is required
2. They handle the problem of sparsity (portion of matrix that is empty) better
3. Scalability. The model typically is smaller than the entire dataset so can be used even for very large datasets.
4. Are typically faster than memory-based
5. Easier to handle the problem of over-fitting

There are many model-based collaborative filtering algorithms [2]. Several research articles indicate that the algorithm with the highest accuracy is matrix factorization [3,4,6]. However, it has also been shown that the “best” algorithm for a given data set may depend on the nature of the data set itself [4].

For this project, the Dataset used will be one of the stable MovieLens datasets found at the website: <http://grouplens.org/datasets/movielens/>. The MovieLens dataset consists of movie ratings collected from the website <http://movielens.com> . Specifically, it will include the movie title, movie genre, user ID, movie ID, movie rating, a tag, and a time stamp. At the onset of the project, the smallest dataset which consists of approximately 100,000 ratings by 700 users of 9,000 movies will be used. This is a sparse data set as evidenced by the fact that if each user had rated every movie, the total ratings would be equivalent to over 6 million ratings yet the data set consists of 100,000. In “Recommender: An Analysis of Collaborative Filtering Techniques”, Christopher Aberger found that the Alternating Least Squares algorithm provided better accuracy and better scaling on the sparse dataset, MovieLens, as compared to other matrix factorization algorithms he tested [3]. Thus, the chosen algorithm for this project is W-ALS, Weighted Alternating Least Squares [1,5].

A follow on to the initial project, is to run the algorithm using Amazon Web Services and compare the compute time to that achieved on a stand-alone platform. Further experimentation would be to run the algorithm on the larger dataset that consists of 24,000,000 ratings by 260,000 users of 40,000 movies to validate scalability.

The deliverables for the project shall include two Jupyter notebooks, a slideshow presentation, and a paper. The first Jupyter notebook shall contain exploratory data in the form of plots, graphs, or histograms. The second Jupyter Notebook shall contain the Python scripts implementing the W-ALS algorithm and the recommendation output. The paper shall include a detailed description of the project and the algorithm used. The slideshow presentation shall include a briefer description of the project, graphs, plots, and a sample of a movie recommendation print out.

References:

[1] Akyildiz, Bugra, Alternating Least Squares Method for Collaborative Filtering, Machine Learning Newsletter, Published: April 19, 2014, <https://bugra.github.io/work/notes/2014-04-19/alternating-least-squares-method-for-collaborative-filtering/>

[2] Su, Xiaoyuan and Khoshgoftaar, Taghi M., “A Survey of Collaborative Filtering Techniques, Advances in Artificial Intelligence”, Volume 2009 (2009), Article ID 421425, 19 pages, Department of Computer Science and Engineering, Florida Atlantic University, Received 9 February 2009; Accepted 3 August 2009, <https://www.hindawi.com/journals/aai/2009/421425/?cm_mc_uid=14123928030314782760842&cm_mc_sid_50200000=1481837637>

[3] Aberger, Christopher R., “Recommender: An Analysis of Collaborative Filtering Techniques”, <http://cs229.stanford.edu/proj2014/Christopher%20Aberger,%20Recommender.pdf>

[4] Joonseok Lee, Mingxuan Sun, Guy Lebanon, “A Comparative Study of Collaborative Filtering Algorithms”, May 14, 2012

[5] Hu, Yifan, and Volinsky, Chris of AT&T Labs, and Koren, Yehuda of Yahoo Research, “Collaborative Filtering for Implicit Feedback Datasets”, <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=34AEEE06F0C2428083376C26C71D7CFF?doi=10.1.1.167.5120&rep=rep1&type=pdf>

[6] Hnot, Taras, “Recommender Systems Comparison”, Rpubs, June 8, 2016, <https://rpubs.com/tarashnot/recommender_comparison>

[7] Recommender Systems, A Computer Science Comprehensive Exercise, Carlton College, Northfield, MN, <http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/modelbased.html>