**Title: Analysis of Best Techniques for Restaurant Recommender System**

**Abstract**

Predictive analysis is an active topic in the field of Data Mining. The rise in social network, and number of users using the social network has resulted in urge of research in this field. This is the primary motivation of the project. In this report, we have implemented a recommender system for restaurants. The model predicts the ratings a particular user will give to a particular restaurant. We have implemented various content based filtering, collaborative filtering and matrix factorization methods. Depending on their respective accuracy, we have ensembled different algorithms to increase the accuracy of the model.

**Introduction**

Recommendation systems provide personalized, relevant recommendations to users and have been used in various domains, such as retail, movie-going, etc. They represent user preferences for suggesting items to purchase or examine. They have become fundamental applications in electronic commerce and information access, providing suggestions that effectively prune large information spaces so that users are directed toward those items that best meet their needs and preferences. A variety of techniques have been proposed for performing recommendation, including content-based, collaborative, knowledge-based and other techniques. To improve performance, these methods have sometimes been combined in hybrid recommenders. For example, a collaborative system and a content-based system might be combined so that the content-based component can compensate for the cold-start problem, providing recommendations to new users whose profiles are too small to give the collaborative technique any traction, and the collaborative component can work its statistical magic by finding peer users who share unexpected niches in the preference space that no knowledge engineer could have predicted.

We have made use of datasets that we obtained from UCI to perform analysis. This dataset was put under rigorous data processing to clean and filter the data which was used as input dataset.We intend to predict the rating that the user would give to a restaurant by making use of restaurant information data, user details data and previous rating data.

**Problem Definition and Methods**

**Task Definition**

The objective of the project is to predict the rating that the user would give to a restaurant by making use of restaurant information data, user details data and previous data rating. We have conducted several combination of algorithms in order to find an ensembled algorithm that gives us the desired accuracy. We check if food and service rating in the regression model would improve the overall accuracy significantly.

**Algorithms and Methods**

Content-based Methods: The system generates recommendations from two sources: the features associated with products and the ratings that a user has given them. Content-based recommenders treat recommendation as a user-specific classification problem and learn a classifier for the user's likes and dislikes based on product features. Following are the two content based methods used in the project:

Random Forest:

Random forests [7] or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Gradient Boosting:

Gradient boosting [8] is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

The idea of gradient boosting originated in the observation by Leo Breiman[1] that boosting can be interpreted as an optimization algorithm on a suitable cost function. Explicit regression gradient boosting algorithms were subsequently developed by Jerome H. Friedman[2][3] simultaneously with the more general functional gradient boosting perspective of Llew Mason, Jonathan Baxter, Peter Bartlett and Marcus Frean.[4][5] The latter two papers introduced the abstract view of boosting algorithms as iterative functional gradient descent algorithms. That is, algorithms that optimize a cost function over function space by iteratively choosing a function (weak hypothesis) that points in the negative gradient direction. This functional gradient view of boosting has led to the development of boosting algorithms in many areas of machine learning and statistics beyond regression and classification.

Collaborative filtering (CF): The system generates recommendations using only information about rating profiles for different users. Collaborative systems locate peer users with a rating history similar to the current user and generate recommendations using this neighborhood. We have used the following two collaborative filtering methods:

User-based Collaborative Filtering (UCF):

UCF [9] predicts a test user’s interest in a test item based on rating information from similar user profiles. Each user profile (row vector) is sorted by its dis-similarity towards the test user’s profile. Instead of computing the similarity between two restaurants, we focus on the similarity between two customers. We use correlation-based method of computing similarity between two customers. Ratings by more similar users contribute more to predicting the test item rating. The set of similar users can be identified by employing a threshold or selecting top-N.

Item-based Collaborative Filtering (ICF) :

ICF[10] approaches apply the same idea as UCF, but use similarity between items instead of users. The unknown rating of a test item by a test user can be predicted by averaging the ratings of other similar items rated by this test user. Again, each item (column vector) is sorted and re-indexed per its dis-similarity towards the test item in the user-item matrix, and, ratings from more similar items are weighted stronger. One the most similar restaurants are found, the prediction is then composed by taking a weighted average of the target user’s ratings on these similar restaurants.

Matrix factorization (MF):

Matrix Factorization characterizes both restaurants and users by vectors of factors inferred from item on rating patterns. High correspondence between restaurants and user factor leads to prediction. Following are the two matrix factorization methods that we used:

Single Value Decomposition (SVD):

The SVD [11] approach models both users and restaurants by giving them coordinates in a low dimensional feature space i.e. each user and each restaurant has a feature vector. And each rating (known or unknown) is modeled as the inner product of the corresponding user and restaurant feature vectors. In other words, we assume there exist a small number of (unknown) factors that determine (or dominate) ratings, and try to determine the values (instead of their meanings) of these factors based on training data. Mathematically, based on the training data (sparse data of a huge matrix), we try to find a low-rank approximation of the user-restuarant matrix A. This approach is called sparse SVD algorithm.

Non-Negative Matrix Factorization (NMF):

NMF[12] is a recent technique for linear dimensionality reduction and data analysis that yields a parts based, sparse non-negative representation for non-negative input data. Essentially, NMF is an unsupervised learning algorithm coming from linear algebra that not only reduces data dimensionality, but also performs clustering simultaneously.

Having a non-negative data matrix containing a set of items with a fixed number of features, NMF algorithm will produce a factorization of the data matrix and reveal the interesting latent factors underlying the interactions between items and their features. We can consider that each item can be described using these latent factors, and, using the information about how much each of latent factors is expressed for each item, we can cluster the items that share the same latent features. If you are familiar with statistical factor analysis you will find that one of the resulting matrices will look just like the one you would get by running a factor analysis – only all factor weights will be non-negative.

Probabilistic Matrix Factorization (PMF)**:** The PMF [4] model scales linearly with the number of observations and performs well on the large, sparse, and very imbalanced dataset. Suppose we have M restaurants, N users, and integer rating values from 1 to 5. Let Rij represent the rating of user i for restaurant j, U ∈ RD×N and V ∈ RD×M be latent user and movie restaurant matrices, with column vectors Ui and Vj representing user-specific and restaurant-specific latent feature vectors respectively. Since model performance is measured by computing the root mean squared error (RMSE) on the test set we first adopt a probabilistic linear model with Gaussian observation noise.

**3. Experimental Evaluation**

This section presents an evaluation of the various collaborative filtering, matrix factorization, content based filtering algorithms and the top performing algorithms of these three types are ensembled together and analysis was carried on a real-world dataset.

**3.1 Data Description**

We make use of restaurant dataset provided by the University of California, Irvine. It consists of ratings between July 2010 to February 2011. There are over 1,153 ratings collected for 128 restaurants that are in three main areas of Mexico– Ciudad Victoria, San Luis Potosi, and Cuernavaca. These ratings are provided by 138 users and each user on an average has rated 8.4 restaurant and each restaurant on an average has received ratings from 9 users, the maximum number of restaurants that a particular user has rated is 18 and the rating sparsity is 92.75%. There are three rating values– 0(dissatisfied), 1(neutral) and 2(satisfied). Apart from the user-restaurant rating data, there are many other data like price, ambience, smoke, parking, payment, cuisine, longitude, latitude, etc. Due to the privacy issue, it is extremely difficult to get a larger restaurant data which contains multi-aspect ratings, the detailed user profile and restaurant profile information []. Even the data we use in this paper has small size, it possesses characteristics of commonly-used recommendation data. A small number of users, restaurants or cities dominate the ratings whereas many users, restaurants or cities obtain only a few. Those observations present the power law distribution, which is usually found on recommendation data [].

**3.2 Evaluation Metrics**

We make use of two metrics to measure how close forecasts or predictions are to the eventual outcomes for different algorithms: (1) Mean Absolute Error (MAE)– MAE is an average of the absolute errors and is a scale-dependent accuracy measure, it’s given by, MAE = ∑i,j |yij − xij|/N; (2) Mean Square Error (MSE)– MSE represents the sample variance of the differences between predicted values and observed values and it’s given by MSE = ∑i,j (yij − xij)2/N, where yij and xij denote the observed overall rating and the predicted overall rating respectively, and N denotes the total number of the tested data. The smaller the value of MAE or MSE, the more accurate is the recommendation of the algorithm.

**3.3 Approach**

We evaluate the recommendation performances of various top performing algorithms on our dataset separately from each of the category: collaborative filtering (ICF, UCF), matrix factorization (PMF, NMF, SVD) and content based filtering (RF, GBM). The top performing CF/MF algorithm’s MSE value is set as benchmark. We then ensemble several of these CF and MF algorithm using different combinations and evaluate the resulting MSE score with that of our benchmark. Next, we evaluate the accuracy of content based techniques- both individual and ensembled ones with the benchmark score. Now the food and service ratings are included in the content based techniques and we evaluate how that impacts the accuracy score of predicting the overall rating. Finally we take the best technique from the previous step and try to ensemble with CF and MF techniques. We use the same test and train data set throughout our evaluation to avoid any sort of bias, and we find the technique that performs the best for the considered dataset by evaluating against the benchmark score set.

**3.4 Results**

**4 Related Work**

The problem of recommender system is a widely studied topic. Over the years, many recommender system techniques have been proposed. One such approach was a Hybrid Web recommender system []. It gives us the different approaches used for designing an accurate Web recommender system. The current technology trend requires such Hybrid systems[] which try to understand the user better and give more accurate results.

Another good article was the survey on different recommender techniques[]. In a restaurant recommender system, it is extremely important to learn more about user preferences. A lot of extensive research has been performed in [] which covers a lot about user preference learning. Matrix Factorization is a vast field and is extensively pursued. While performing several of our permutations amongst algorithms to achieve accuracy we found the research done in [] and [] has helped in improving our understanding.

**Conclusion**

We performed various filtering methods on our data and calculated their respective accuracies.

Depending on their respective accuracy, we have ensembled different algorithms to increase the accuracy of the model.

An ensembled model was observed to be more accurate than any methods implemented individually.

**References**

**[1] Burke, Robin. "Hybrid recommender systems: Survey and experiments." *User modeling and user-adapted interaction* 12.4 (2002): 331-370.**

**[2] Burke, Robin. "Hybrid web recommender systems." *The adaptive web* (2007): 377-408.**

**[3] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 42.8 (2009).**

**[4] Salakhutdinov, Ruslan, and Andriy Mnih. "Probabilistic Matrix Factorization." *Nips*. Vol. 1. No. 1. 2007.**

**[5] Agarwal, Deepak, and Bee-Chung Chen. "Regression-based latent factor models." *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009.**

**[6] Fu, Yanjie, et al. "User preference learning with multiple information fusion for restaurant recommendation." *Proceedings of the 2014 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2014.**

**[7] Breiman, Leo. "Random forests." *Machine learning* 45.1 (2001): 5-32.**

**[8] Wikipedia contributors. "Gradient boosting." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 30 Apr. 2017. Web. 8 May. 2017.**

**[9] Yao, Guanwen, and Lifeng Cai. "User-Based and Item-Based Collaborative Filtering Recommendation Algorithms Design."**

**[10] Sarwar, Badrul, et al. "Item-based collaborative filtering recommendation algorithms." *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001.**

**[11] Kalman, Dan. "A singularly valuable decomposition: the SVD of a matrix." *The college mathematics journal* 27.1 (1996): 2-23.**

**[12] Lee, Daniel D., and H. Sebastian Seung. "Algorithms for non-negative matrix factorization." *Advances in neural information processing systems*. 2001.**