Alternating Least Squares with Incremental Learning Bias

Than Htike Aung
Computer Science Department
Assumption University
Bangkok, Thailand
g5619309@au.edu

Rachsuda Jiamthapthaksin
Computer Science Department
Assumption University
Bang Na-Trad Km. 26, Bangsaothong,
Samuthprakarn, Thailand
rachsuda@scitech.au.edu

***Abstract*— Recommender systems provide personalized suggestions for every individual user in the system. Many recommender systems use collaborative filtering approach in which the system collects and analyzes users' past behaviors, activities or preferences to produce high quality recommendations for the users. Among various collaborative recommendation techniques, model-based approaches are more scalable than memory-based approaches for large scale data sets in spite of large offline computation and difficulty to update the model in real time. In this paper, we introduce *Alternating Least Squares with Incremental Learning Bias (ALS++)* algorithm to improve over existing matrix factorization algorithms. These learning biases are treated as additional dimensions in our algorithm rather than as additional weights. As the learning process begins after regularized matrix factorization, the algorithm can update incrementally over the preference changes of the data set in constant time without rebuilding the new model again. We set up two different experiments using three different data sets to measure the performance of our new algorithm.**

Keywords-recommender system; collaborative filtering; algorithms

I. INTRODUCTION

Information over the internet has been growing over the time. Facing large amount of information, it becomes more difficult to obtain correct and precise information to match individual's preference. Recommender systems become the tool to address that kind of challenge. The systems are based on two different approaches or combination of both.

The first one is content based approach which makes a profile for each user or product to characterize its nature using their intrinsic factors. For example, a profile for a movie is based upon its genre, directors and actors, popularity and other factors. A user profile may include age, sex, location, etc. The systems associate the users with suitable products by using their respective profiles.

Another approach is collaborative filtering that use past preferences of users and items in order to predict new user-item preferences. For example, collaborative filtering identifies similar users with similar history of ratings to find out relationships between users. The recommendation of new ratings for users is based upon known preferences of a group of users [6]. The advantage of collaborative filtering is that it relies only on the past preferences of users making it as domain independent approach. However, collaborative filtering suffers from cold start problem due to inability to get past history of new users or products where content based approach would be more suitable. According to [2], collaborative filtering algorithms can be classified into two groups: memory-based and model-based. The user-based approaches are memory-based because the original rating database is stored in memory and used directly for generating the recommendations. It constructs the relationships between users or items to build neighborhood by their similarities and makes predictions based on known ratings by the active user's neighbors or those on the active item's neighbors.

In model-based approaches, the ratings data are first processed offline which requires high computation. At run time, only the precomputed model is required to make predictions. In latent factor model, it transforms both users and items in latent factor space, characterizes each entity with a number of feature vectors which are inferred from the existing ratings. Prediction making for unknown ratings is done by inner products of the corresponding vector pairs. One successful kind of approaches for latent factor model is based on matrix factorization. Some non-negative matrix factorization approaches [14] use factorization by minimizing least square errors from user-item ratings matrix. Alternating Least Squares (ALS) introduced in [10] is considered regularized low-rank matrix factorization.

A*lternating least squares with weighted-λ-regularization* can perform parallelizable algorithm with good accuracy as discussed in [1]. However, these matrix factorization techniques require a batch-training process based on static training data set. But in real world, this scenario is less likely to be happened. In modern e-commerce systems, user feedbacks vary at every millisecond. One approach to catch up this rapid data expansion is to rebuild the model if the data growth exceeds a pre-defined threshold. During the interval between old model and new model, the system cannot capture changes over behaviors of users and items. This lack of efficient update mechanism causes scalability problem for large data set in real time recommender systems. The desirable approach to cope with such situations is to enable the system to update incrementally in accordance with the newly arriving data. In this paper, we will introduce incremental learning bias which not only improve accuracy but also can update incrementally in constant time. We setup two