
Amazon Product Recommendations Using Collaborative Filtering and Natural Language Processing

Nikhil Aourpally
Nikhil.Aourpally@asu.edu

Anthony Helmstetter
anthony.helmstetter@asu.edu

Prajwal Paudyal
ppaudyal@asu.edu

Anirudh Som
Anirudh.Som@asu.edu

Abstract

In this project we will implement a Recommender System to provide suggestions based on Amazon user reviews across Amazon Instant Video, Music, and Book product categories. The Recommender System will be implemented using collaborative filtering based on product ratings. To increase recommendation accuracy, we will also perform natural language processing on user written reviews to better inform the recommender system of user preferences. We will assess the accuracy of the Recommender System with and without additional textual information contained in user reviews, and evaluate the sensitivity of our Recommender System by varying the number of reviews by user, and number of reviews per product.

1 Introduction

Electronic retailers and content providers offer a huge selection of products, with unprecedented opportunities to meet a variety of special needs and tastes. Due to this, consumers are inundated with choices. Thus, matching consumers with most appropriate products is key to enhancing user satisfaction and loyalty.

1.1 Recommender Systems

–Thus– Most large-scale retailers have become interested in recommender systems, which are techniques to analyze patterns of user interests in products to provide personalized recommendations that suit a that specific user’s tastes. Personalized recommendations like this, can add another dimension to the user experience by providing recommendations that help users discover items that she otherwise would not –possibly– have found. Following this trend, Amazon.com, Netflix and other e-commerce leaders have made recommender systems a salient part of their websites. With the amount of data available ever-growing, recommender systems have become an inherent way for information filtering. To fulfill these expectations, a recommender system needs to complete two tasks:

- Use User’s past behavior to predict ratings on items that the user has not sampled before.

- Then, generate a list of recommendation items for the user based on the result of the previous step, and a number of factors for each item recommended, such as diversity, novelty, and credits.

The first part is the key for generating user specific ratings since history is taken as an important factor in future recommendations. The second part, i.e. the factors in a movie recommendation case will probably be the movie genre, actors/ directors etc. but in practice, since we rarely have all this fine-grained information, the factors are more likely a combination of these which is not known.

1.2 Collaborative Filtering

There are two types of recommendation systems that are generally used a) The content filtering approach creates a profile for each user or product to characterize its nature. The profiles allow programs to associate users with matching products, however this strategy requires gathering external information that might not be available or easy to collect. b) The 2nd approach is known as collaborative filtering, which relies on past user behavior i.e. previous transactions or product ratings without requiring the creation of explicit profiles. 2) Collaborative filtering analyzes relationships between users and interdependencies among products to identify new user-item associations. 3) The goal of collaborative filtering is to suggest the new items to users by predicting a score of an item (the score can be 0/1 or 0-5, etc.) for a particular user based on his consume log of some old items as well as the log from other users with similar tastes. 4) Memory-Based collaborative filtering algorithm has two models, one is user-based and the other is item-based. 5) The basic idea of user-based CF is that if the user A like item a, the user B like item a,b,c, user C like item a and c, so that user A and user B and C are similar and c is recommended to the user A. 6) The item-based CF calculates the similarity between items with all the history of the user's preference data and the user is recommended with similar items. 7) Model-based CF are developed using data mining, machine learning algorithms to find patterns based on training data. These are used to make predictions for real data. 8) Many algorithms are derived from collaborative filtering. 9) A major appeal of collaborative filtering is that, it is domain free, yet it can address data aspects that often elusive and difficult to profile using the content filtering. However, it suffers from cold start problems, due to its inability to address the system's new products and users. In this aspect, content filtering is superior.

1.3 Matrix Factorization

1) The two primary areas of collaborative filtering are the neighborhood methods and latent factor models. 2) Neighborhood methods are centered on computing the relationships between items or, alternatively, between users. The item oriented approach evaluates a user's preference for an oriented item based on ratings of neighborhood items by the same user. A product's neighborhood are other products that tend to get similar ratings when rated by the same user. 3) Latent factor models are an alternative approach that tried to explain the ratings by characterizing both items and users on, say 20 to 100 factors inferred from the ratings patterns. 4) Some of the most successful realizations of latent factor models are based on matrix factorization (MF). 5) The matrix factorization technique is gradually applied to personalized recommendation algorithm. It does a global optimization on object, so MF performs better than neighborhood methods in Root Mean Square Error (RMSE). 6) In the basic form, matrix factorization characterizes both items and users by vectors of factors inferred from item rating patterns. High correspondence between item and user factors leads to a recommendation. 7) One strength of matrix factorization is that it allows incorporation of additional information. When explicit feedback is not available, recommender systems can infer user preferences using implicit feedback, which indirectly reflects opinion by observing user behavior including purchase history, browsing history, search patterns, or even mouse movements. Implicit feedback usually denotes the presence or absence of an event, so it is typically represented by a densely filled matrix. 8) DESCRIPTION: Page 44

1.4 Alternating Least Squares

1) We have two approaches to minimize equation2 ? a) stochastic gradient descent b) alternating least squares - ALS 2) Both q_i and p_u are unknowns, equation2(Page 44) is not convex. 3) By fixing one of the unknowns, the optimization problem becomes quadratic and can be solved optimally. Thus the ALS techniques rotate between fixing the q_i 's and fixing the p_u 's. When all p_u 's are fixed, the system recomputes the q_i 's by solving a least-squares problem, and next when all the q_i 's are fixed, the system recomputes the p_u 's by solving the least squares problem. This ensures that each step decreases equation2 until convergence. 4) In general, stochastic gradient descent is easier and faster than ALS. However, ALS is favorable in at least two cases ? a) since the system computes q_i and p_u independently of the other user factors respectively, the system can utilize massive parallelization of the algorithm. b) when the system is centered on implicit data, because the training set cannot be considered sparse, looping over each single training case as gradient descent does would not be practical.

1.5 Natural Language Processing

1) We have two approaches to minimize equation2 ? a) stochastic gradient descent and b) alternating least squares (ALS) 2) Both q_i and p_u are unknowns, equation2(Page 44) is not convex. 3) By fixing one of the unknowns, the optimization problem becomes quadratic and can be solved optimally. Thus the ALS techniques rotate between fixing the q_i 's and fixing the p_u 's. When all p_u 's are fixed, the system recomputes the q_i 's by solving a least-squares problem, and next when all the q_i 's are fixed, the system recomputes the p_u 's by solving the least squares problem. This ensures that each step decreases equation2 until convergence. 4) In general, stochastic gradient descent is easier and faster than ALS. However, ALS is favorable in at least two cases ? a) since the system computes q_i and p_u independently of the other user factors respectively, the system can utilize massive parallelization of the algorithm. b) when the system is centered on implicit data, because the training set cannot be considered sparse, looping over each single training case as gradient descent does would not be practical.

1.6 Combination

2 Problem Description

1) The rapid growth of internet and e-commerce has brought the problem of information overload. 2) Excessive information makes it difficult for a user to pick up what they really want and reduces the efficiency of information usage. 3) Several search engines or e-commerce online platforms use personalized recommender systems in searching results. 4) Personalized recommender systems not only performs better than traditional ones but also adds another dimension to the user's experience, by analyzing patterns of the user's interest in products to suit their taste. 5) Using Amazon product reviews, we implemented a recommender system to suggest new movies to Amazon users. 6) The suggested movies are determined by highly rated products as rated by similar users. 7) Similarity of users will be determined by the degree to which the users have rated identical products similarly. 8) Our recommender system falls under the collaborative filtering category, in which we filter based on similarity of users and not similarity of products. 9) To enhance the performance of our recommender system, we will also incorporate additional information derived from text reviews written by users using Natural Language Processing, to develop a more accurate metric of user similarity. 10) We will evaluate the performance of our recommender system both with and without the incorporation of additional textual information and compare the results.

3 Methodology

4 Results

5 Conclusions