MAUT Based Recommendation

A Project Report

submitted by

BHARATH REDDY A

in partial fulfilment of the requirements
for the award of the degrees of

MASTER OF TECHNOLOGY
&
BACHELOR OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

April 2014

THESIS CERTIFICATE

This is to certify that the thesis titled **MAUT Based Recommendation** submitted by **Bharath Reddy A**, to the Indian Institute of Technology, Madras, for the award of the degrees of **Bachelor of Technology and Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Sutanu Chakraborti
Assistant Professor
Dept. of Computer Science and Engineering
IIT-Madras, 600 036

Place: Chennai

Date:

ACKNOWLEDGEMENTS

I am very thankful to Dr. Sutanu Chakraborti ¡fill-TODO¿

I am ever grateful to my friends Skanda Raj, Shubranshu Shekar, Saurabh Gupta and Dileep for helping me a lot during the project.

I thank the Department of Computer Science as whole. I am immensely happy to say that I have got such a good experience during the last 5 years. For this, the credit must go to remarkable teachers like Dr. Madhu Mutyam, Dr. Shankar Balachandran, Dr. Sutanu Chakraborti, Dr. Ravindran and Dr. C Pandu Rangan

ABSTRACT

Most commercial recommender systems in practice use collaborative filtering (CF) techniques that rely heavily on user-ratings to make recommendations. However, CF may not perform well in high-risk product domains like cars, cameras, houses etc. where there a low number of ratings.Knowledge based Recommenders are used to provide recommendations in these scenarios. Users often want to define their requirements explicitly - "The maximum price of PC should be *x* and HDD capacity should be atleast *500 GB*." and engage in an interaction with the system. Thus, the recommendation process of a knowledge based recommender is highly interactive, and thus they are also characterized as *conversational recommender systems*. Conversational recommender systems mimic the kind of dialog that takes place between a customer and shopkeeper involving multiple interactions and where the user can give feedback at every interaction. *Critiquing* is a popular form of feedback in conversational recommendation systems.

Dynamic generation of appropriate compound critiques in each cycle is a critical issue for critique-based conversational recommender systems. In earlier research, Apriori algorithm and MAUT (Multi Attribute Utility Theory) based generation of compound critiques have been proposed. MAUT based recommendation has been shown to be slightly superior to Apriori Algorithm based recommendation in offline experiments and live user studies. "Average number of interaction cycles per recommendation session" is a measure that is often used to measure the goodness of a recommendation algorithm. Lower the number of cycles, better is the performance of the algorithm. In this project, we propose several modifications to the MAUT based generation of compound critiques and report the improvements in performance caused by each of these modifications.

CHAPTER 1

Introduction

"Which digital camera should I buy? Which movie should I rent? Which book should I buy for my next vacation?" These are some situations where people have to make decisions about how they are going to spend money, or in a broader level, about their future. Traditionally, people have used a variety of strategies to solve such decision making problems: conversations with friends, obtaining information from a trusted third party, hiring an expert team or simply follow the crowd. It would be great to have an affordable personal advisor who helps us make good decisions efficiently. The construction of systems that supports a user in his (online) decision-making is the main goal of the field of recommender systems. In particular, the goal of recommender systems is to provide easily accessible, high-quality recommendations for a large user community. Recommender Systems are broadly classified into three categories: Collaborative, Content Based and Knowledge Based.

1.1 Collaborative Recommender Systems

The main idea in these systems is that if users share the same interests in the past - if they viewed or bought the same books - they will also have similar tastes in the future. This technique is also called as *Collaborative Filtering*. Pure CF based approaches do not require the additional knowledge about underlying users/items. Hence, the algorithms are usually domain independent. Most commercial recommender sytems use collaborative filtering for recommending items. There are two approaches to do CF: Memory Based Approaches and Model Based Approaches

1.1.1 Memory Based Approaches

In this approach, the original rating matrix is held in memory and directly used to generate predicted ratings and recommendations. There are two popular memory based approaches:

User based Nearest Neighbor(NN) Recommendation: Given a user u, the system computes top K similar users to u according to a pre-defined similarity measure. It recommends those items to user that haven't been rated/purchased by u but liked by the top K similar users.

Item based NN Recommendation: Given a user u, the system recommends items that have received similar ratings to the ones that u has previously liked.

1.1.2 Model Based Approaches

As opposed to memory based approaches that use the ratings matrix to directly generate predictions, model based approaches learn models corresponding to each item and each user from ratings matrix and the learned models are used to make predictions at run time. Model based approaches perform well in practice for large datasets. Matrix factorization is a popular model based approach. The superiority of matrix factorization techniques over traditional CF in improving prediciton accuracy was clearly seen during The Netflix prize competition. Broadly speaking, matrix factorization methods derive a set of latent(hidden) factors from the rating patterns and characterize each item and user as vectors of these factors. In the movie domain, such latent factors can correspond to some aspects of a movie like genre, but most of them are completely uninterpretable (Koren *et al.* (2009))

1.1.3 Limitations of CF

In real-world applications, the user-item ratings matrix tends to be very sparse, as customers generally provide ratings for very few items. CF systems have a limitation called the "cold-start problem". It has two aspects - how to recommend items who has given zero/very low number of ratings; and how to recommend those items that haven't received ratings yet.

1.2 Content based recommender systems

Content based recommender systems rely on item descriptions and user profiles to recommend items. If the system knows that "Harry Potter" is a fantasy novel and the user *Alice* has always like fantasy novels, the system can recommend the new "Harry Potter" book right away. So the system need not rely on the existence of a large user base to generate recommendations. It overcomes the cold-start problem described in the section 1.1.3. However, item characteristics are hard to acquire normally and hence, they have to be entered manually into the system, which can be potentially expensive for some domains.

1.3 Knowledge based recommender systems

Typically, we do not buy a house, a car or a computer very frequently. In such a scenario, a pure CF system will not perform well because of the low number of available ratings In more complex and high-risk product domains such as cars, customers often want to define their requirements explicitly - for example, "the maximum price of the car is x and the color should be *black*". In knowledge based systems, recommendations are made taking into account the explicit user preferences and the rich knowledge base available.

Recommendation process of knowledge-based recommender applications is highly interactive, a foundational property that is a reason for their characterization as ***conversational systems***. The recommender system that we consider in this project is a conversational system.
Conversational systems assume that a user's initial query is merely a starting point for search, perhaps even an unreliable starting point. The job of the recommender system is to help the user refine his initial preference query as the interactions proceed.

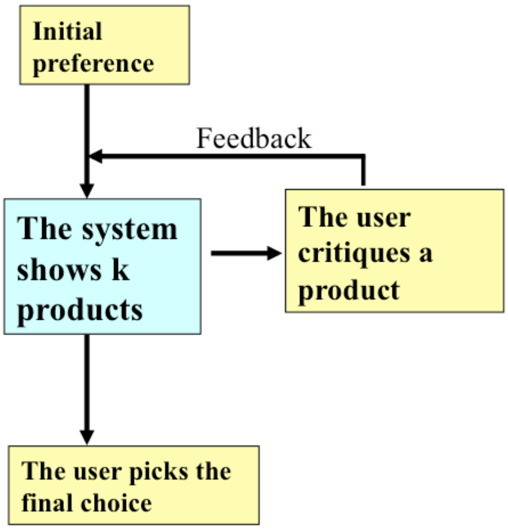

Figure 1.1: Critiquing

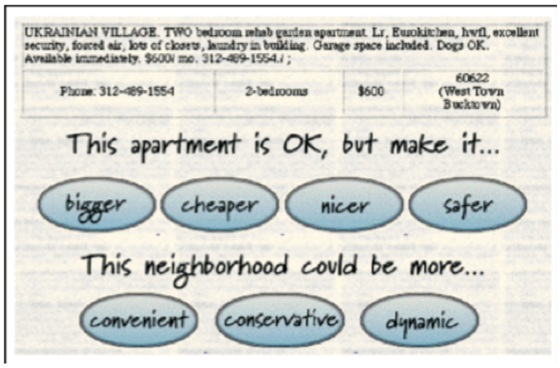

Figure 1.2: RentMe Recommender System: Burke (2000)

1.3.1 Critiquing

Critiquing is one of the most popular forms of feedback in conversational recommender systems. In each interaction cycle, the user is presented with a list of products. User selects a product and expresses directional preference(s) over one or more item feature values. For example, one might indicate that he/she is looking for a less expensive restaurant or a more formal setting. These are two individual critiques, first critique being on the *price* attribute and the second critique on the *setting* attribute. The recommender updates it's user model according to this feedback provides another set of products and proceeds to the next recommendation cycle. This continues till the user finally chooses a product.

Unit critiques allow users to express their preference over one attribute in each interaction cycle. ***Compound critiques*** enable users to input their preferences on several attributes at a time. This can potentially shorten the number of interaction cycles in finding a target product.

1.4 Our contribution: Improvements to MAUT based recommendation

1.5 Organization of the Thesis

REFERENCES

1. **Burke, R.** (2000). Knowledge-based recommender systems. *Encyclopedia of library and information systems*, **69**(Supplement 32), 175–186.

2. **Koren, Y.**, **R. Bell**, and **C. Volinsky** (2009). Matrix factorization techniques for recommender systems. *Computer*, **42**(8), 30–37.