RECOMMENDATION SYSTEM

MemoryLeak

October 5, 2013

Team Members:

Bahtiyar Onur Geyik, 1745975, b.onurgeyik@gmail.com

Emre Ercan, 1745918, emre.ercan.91@gmail.com

Gökçe Aksu, 1745694, gokceaksu2@gmail.com

Ege Bozkurt, 1745819, bozkurtege@gmail.com

Problem Definition and Background Information

Websites and applications that offer their users or customers an item or a social element, have been trying to recommend them relevant content according to items/elements which they are interested in. Increasing the time that user spends on the website and increasing the interest of the user to the items in the website are the main reasons why the recommender systems are being used. Accuracy and time-efficiency are the most common problems of recommender systems. We will be trying to design an accurate and fast algorithm which will solve these problems.

People often complain about the irrelevant recommendations of the websites they are using. Users sometimes even complain about websites like Amazon or Netflix, even though they are considered as having the best recommender systems. Since most of the global websites that offer their users an item, the problem can be considered as a worldwide problem.

Some of the global websites which are using a recommender system have been using the insourcing method to get the system. However, outsourcing is the most common way among most websites. There are many software companies and university labs which are working on recommender systems. Websites like Amazon, Netflix, Google, Facebook have their own recommender systems. Also, there are researchers in universities like Stanford who are working on recommender systems. Even competitions have been organized to get the most accurate and fast recommender system worldwide. Netflix Prize is the best known competition. It began on October 2, 2006 and on 21 September 2009; the grand prize of

US$1,000,000 was given to the BellKor's Pragmatic Chaos team which bested Netflix's own algorithm for predicting ratings by 10.06%.

For people who are totally unaware of how a recommendation system works, we can give definitions of some of the basic concepts related to the evaluation of the effectiveness of a recommender system, which is one of the most important issues to understand the topic. As it is explained in the webpage of Creighton University, "recall", "precision" and "DCG" (Discounted Cumulative Gain) are the common metrics to assess the quality of the recommendation method. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage too. DCG measures the usefulness, or gain, of a document based on its position in the result list. The gain is accumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks. These are the basic concepts used in calculating recommendations.

Significance of the Problem and Motivation

The main challenge of this problem is dealing with big data. Like many other environments which are using recommendation systems, our project will be dealing with millions of users and products. Therefore we need to come up with a large amount of computational power to make recommendations. The other problem is that we actually need big data to make accurate recommendations. Although the amount of metadata is huge, we also need ratings and feedback from the user to be able to perform collaborative filtering. In general, even most active users will have been expected to rate a very small portion of the overall database, therefore most of the items in the database remain unrated. The last problem we want to mention is that, we have to find the most efficient and effective way of combining these two data, which are metadata and data about users, as most of the applications today need to be compatible with both content-based filtering and collaborative filtering.

We thought that it would be better for us to choose a project which is supported by a company and a professor from our university. Among those projects, this one was the only project which all of our team members are interested in and excited to work on. Since we are living in a world with incredible amount of data, we think that it would be useful for us to learn new technologies like Mahout, Map-reduce, NOSQL and neo4j for our future career after graduation.

A recommendation system is a system such that, always a faster and more accurate solution can be found. We already have some solutions to this problem on different applications, but each application needs a different algorithm and different solution depending on the data type. Therefore, for our application, we need to come up with a new algorithm and we may improve some of the existing algorithms to make them faster and more accurate.

In recommendation systems, there is no magic formula, which means that the solutions differ based on the item type, how the data is stored and the needs of the customer. Therefore, missing part of the existing solutions are that they are developed to fulfill the needs of their own customers, which means that we also need to provide different solutions to our customers based on previously mentioned criteria.

When the problem is solved, the customers using our recommendation system will be able to provide better recommendations to its users. Therefore, the users of our customers will feel more satisfied.

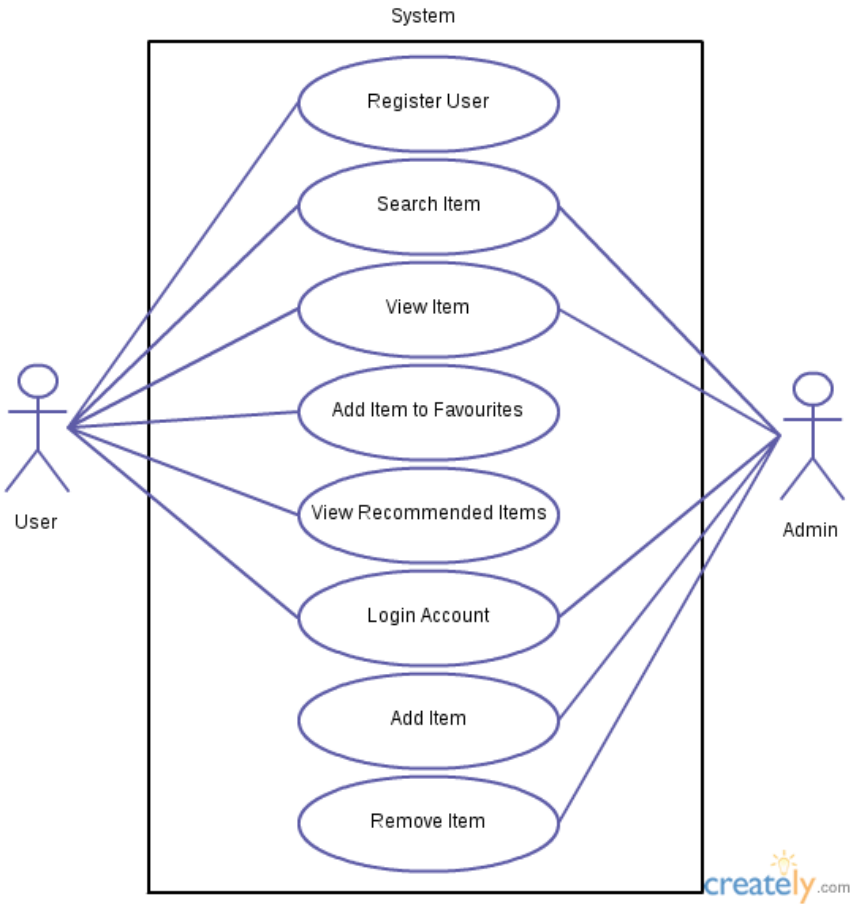
The project will mainly be a recommendation algorithm, which will turn into a commercial product, when finished. The steps that MemoryLeak is planning to take are that first, both satisfying the target customer's needs, TTNET Müzik's needs and providing them the algorithm which they desire, then we will sell the algorithm to other companies, who desire to recommend items to their customers.

Draft Project Plan

The prediction of the rating value is made by a technique called Collaborative Filtering. Basically, the system will observe the items that a user views. It will also keep the records of the items that a registered user views in a NOSQL database. Then it will analyze item/user viewing rates. If that rating is above some threshold or in top k of the list of the items based on their ratings, it can be recommended.

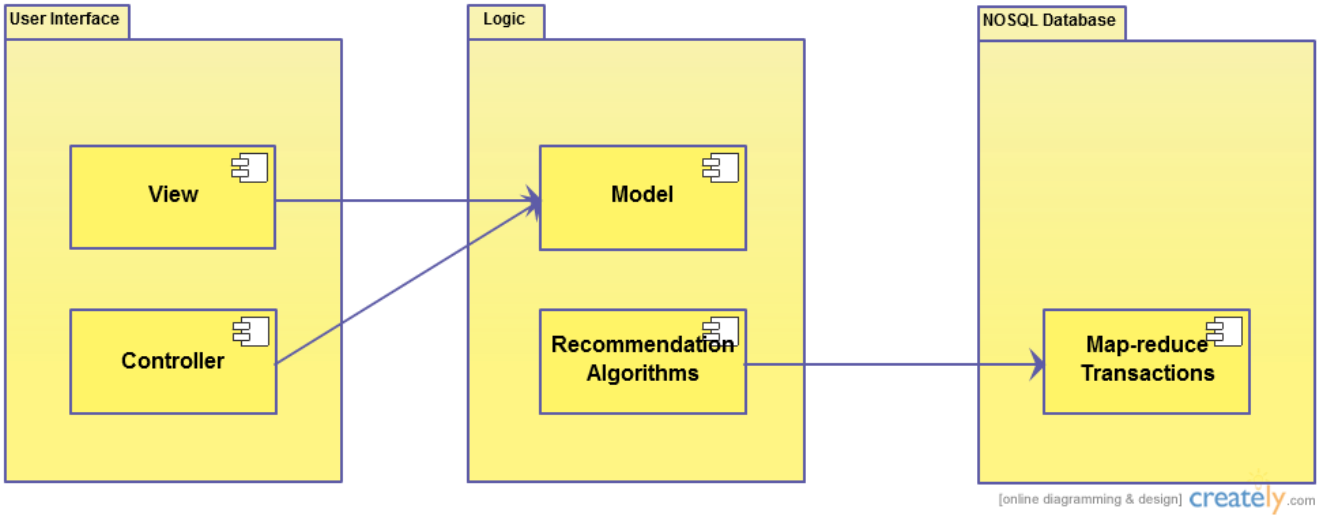
The end-product will basically be a website. There will be two different user types, which are registered and unregistered. The system will show the items in it to the both type of users. The unregistered users will see some recommended items according to the current item that they are viewing. Moreover, the registered users will see recommendations based on their past actions in the website. Amazon and Netflix can be given as examples.

Major tasks are research, backend and frontend of the project. Research will be done by all project members. Backend stage includes handling big data and designing recommendation algorithms. Each sub-stage will be assigned to two project members. Frontend task includes designing a website using Model-View-Controller structure, and one project member will be responsible for this task.



Use Case Diagram

The user will be able to register to the system by providing a username and password and then login by using those two. He/she also will be able to search for desired items by writing their names into a textbox and clicking the search button. The user will view items via clicking on the item name and add the item to the favorites list. The user will be able to see the recommendations on the screen while viewing other items. The admin can add/remove items to/from the system and also he or she has to login to the system.



Component Diagram

Our project software will contain a model-view-controller architecture. A model can be taught as a database table or an object representation of data or an event. A controller is the part which is able to change the state of the model and a view is a representation of that state of the model. In our project, user interface will enable the user to control the model by manipulating it, which will result in updates in the view. For example, user will be able to make ratings and according to his/her ratings he/she will be able to view the recommendations as a view. Moreover, user will be able to search for an item, and then he/she will be able to view the result. We are thinking of using PHP and HTML to design the user interface. One or two team members will be working on this part.

In database module, we are planning to use a NOSQL database. It can be a document database like CouchDB or MongoDB or it can be a graph database like Neo4j. We will definitely use a map-reduce algorithm to query our NOSQL database. To write the map-reduce algorithm, we will be using Java or C++ programming language and most probably Apache Mahout Library from Hadoop project. The most important task of the project is to implement recommendation algorithm in the business logic module. We haven't decided what programming language we will use to implement it yet. However, Java and C++ programming languages are our options. Two of our team members will be working on the database module and the other two will be working on the modeling & algorithms module.

Deployment Diagram

View module will be working on PC. Controller and Model modules will be working on a web server. Our recommendation algorithm will also be working on the web server. Database module will be working on the database server. Hence, logic and database part of the project needs a server to work on.

Support

The project will be supported by AGMLab, Dr. Guven Fidan and Prof. Dr. Ismail Hakki Toroslu. The contact information of the supporter is the e-mail address of Prof. Toroslu, which is toroslu@ceng.metu.edu.tr. As support, our group will get tools and the necessary data to develop the project and also consultancy from AGMLab. The intellectual property right of the end-product is still under discussion.

References

- http://www.tmsp.umd.edu/position%20papers/riedl-statement.pdf
- http://en.wikipedia.org/wiki/Recommender_system
- http://www.creighton.edu/fileadmin/user/HSL/docs/ref/Searching_-_Recall_Precision.pdf
- http://en.wikipedia.org/wiki/Discounted_Cumulative_Gain