RECOMMENDATION SYSTEM

Dcengo Unchained

October 8, 2013

Team Members:

1) Duygu Kabakcı, 1746064, duygukabakci@gmail.com
2) Işınsu Katırcıoğlu, 1819432, isinsu.katircioglu@gmail.com
3) Sıla Kaya, 1746122, silakaya91@gmail.com
4) Mehmet Koray Kocakaya, 1746189, mkoraykocakaya@hotmail.com

1. Problem Definition and Background Information

Recommendation systems are software tools or knowledge discovery techniques that provide suggestions for items to a user. Items can be music, books, movies, people or social groups. The first purpose of this system is to recommend items that a user is likely to be interested in. The second purpose is to learn more about their preferences and constraints. The main source of our data is user behavior. Data collection from user behavior can be classified as either implicit or explicit [2]. Implicit data collection consists of observing the items that a user views in an online store, analyzing item/user viewing times, keeping a record of the items that a user purchases online, obtaining a list of items that a user has listened to or watched on his/her computer and analyzing the user's social network and discovering similar likes and dislikes. On the other hand, explicit data collection consists of asking a user to rate an item on a sliding scale, asking a user to rank a collection of items from favorite to least favorite, presenting two items to a user and asking him/her to choose the better one of them and asking a user to create a list of items that he/she likes [1]. When a user is asked to give an opinion through a rating scale, the following forms can be considered: Numerical ratings such as 1-5 stars, ordinal ratings such as "strongly agree, agree, neutral, disagree, strongly disagree", binary ratings that model choices in which the user is simply asked to decide if a certain item is good or bad, unary ratings that can indicate whether a user has observed or purchased an item. In unary cases, the absence of a rating indicates that we have no information relating the user to the item. The recommendations are usually personalized so that each user receives specific suggestions. Besides, there are non-personalized recommendations which gather information based on general popularity. This is done through Active Learning (AL), Data mining, Machine Learning and Artificial Intelligence which provide diverse knowledge for constructing probabilistic results. The aim is to help users become more aware of their preferences and feed new data into the system for further analysis. The vast majority of today's enterprises make use of recommender systems in the fields of commerce, Internet streaming and social networking services. The leading projects and contributions about recommendation systems belong to Amazon, Netflix, Pandora Radio, YouTube, Facebook, Last.fm, 8tracks, eBay, LinkedIn, MySpace, Internet Movie Database (IMDb) and so forth. Some of their recommendation approach will be discussed later on.

There are two main paradigms studied for the problem of recommending items: Content-based and collaborative filtering paradigms. Content-based filtering paradigm is based on matching the characteristics of an item to the preferences of a user. In order to come up with an accurate match, attributes of the item must be specified. An attribute defines a unique feature that enables us to classify an item. The attributes are then represented in a weighted vector for each item. The weight corresponds to the degree of importance that a user assigns to a feature. The next thing to consider is the user profile. User profile describes the detailed model of tastes and preferences of a user. In addition to the user profile, the user is expected to give direct feedback through rating the item offered by the system. This can be used to assign higher or lower weights to certain attributes of the item. The entire system using this approach improves as better personalization is maintained [2]. A simple movie example can illustrate the content-based approach. If a user rates a movie from Sci-Fi genre or includes this genre in his/her preferences then a movie from this genre is recommended to the user. The well-known example of content-based filtering is “Music Genome Project” of Pandora Radio [5]. Pandora’s recommendations are based on the inherent qualities of the music. This comprehensive project uses mathematical algorithms to organize the songs according to 400-450 attributes such as melody, harmony, lyrics, orchestration, vocal character, gender of the lead vocalist, level of distortion on a specific instrument and so on. Pandora calls these musical attributes “genes” and each “gene” is assigned to a number between 0-5. The database of Pandora consists of thousands of songs classified according to hundreds of such attributes.

The second well-known filtering method is collaborative filtering. It is based on collecting and analyzing huge amount of information on users’ behaviors, activities, preferences and similarities to other users. Therefore, it is referred as people-to-people correlation. Similar users are determined according to the similarities between their evaluations of certain attributes for a specific group of items. Analysis concerning user’s past behavior has extensive applications such as previously purchased item, recently watched movie (or music preference) and search history. One of the most famous examples of collaborative filtering is item-to-item collaborative filtering (people who buy x also buy y), an algorithm popularized by Amazon.com recommendation system. It is based on customizing the browsing experiences of users instead of matching similar users. Amazon.com algorithm can support massive data sets and millions of customer information. The algorithm matches each of the user’s purchased and rated items and puts them in a recommendation list [4]. From this list, it is more likely to construct a product-to-product matrix which scans through all pairs and assigns a similarity metric to them. Last.fm is another example of the collaborative filtering system. Last.fm recommends music through comparing the listening habits of similar users. It collects the profile information of each user by keeping track of the songs that the user has listened. Their data set is also rich in tag information. Moreover, Facebook, MySpace, LinkedIn and other social networks use collaborative filtering to recommend new friends, groups and other social connections through examining the network of connections between a user and his/her friends.

It has been experienced that both the content-based and the collaborative approaches have some downsides. The scope of content-based approach is limited and it may suffer from little information of user. The collaborative approach requires large amount of data to be accurate but this might be a problem at the very beginning of the system when the data set is insufficient. Therefore, hybrid models are widely used today. They provide combination of previously mentioned approaches. We will be using hybrid models in order to contribute to the existing systems.

There is a wide range of methods proposed to solve and improve the existing recommendation systems. One such method is constructing a rating matrix R that stores the vote for an item for each user-item pair. Accordingly, there is a non-negative number r in the cells of the matrix. However, there might be many missing values due to the insufficient rating. Therefore, predictions of ratings are calculated by finding the similarities between users. A rating of some user u for an item i is deduced from ratings of that item by users with high similarity to the user u [1]. For this purpose, neighborhood based model and regularized matrix factorization model are used together in hybrid methods [8]. Neighborhood formation scheme usually uses Pearson correlation which is currently weak for large and sparse databases [7]. Another recent method is Association Rule Mining which is a technique in data mining. The purpose is to find relation among different items in databases such as the books borrowed together from a library or the items bought together in a supermarket [3]. Apart from these methods, Bayesian classifier, uncertainty-based active learning, error-based active learning [6], cluster analysis, decision trees and artificial neural networks are used to group similar items and estimate the probability that a user will prefer them. Netflix Prize, which is an open competition for the best collaborative filtering algorithm to predict user ratings for films, encourages the teams to improve the existing methods and reduce the root mean square error to yield more accurate predictions.

In this project, we will be working on the recommendation system of TTNET Müzik. TTNET Müzik is a music downloading and listening service supported by TTNET. This service provides TTNET ADSL users with the legal right of accessing music files. It also enables the users to download music in mp3 format in case of purchasing music packages. The service provides the opportunity of displaying latest news and music lists along with going over the lyrics and album information. Their recommendation system allows creating and sharing personal music lists. A user can also view the music lists generated by other members. This is done through suggesting relevant lists to users. Our goal is to use and extend the existing recommender system of TTNET Müzik. Our targeted user profile is ordinary web or mobile users. According to 2012 data, it was expected to affect more than 3.5 million TTNET Müzik users monthly with a more efficient recommendation system. This data shows that there are several millions of target users for this end product. Moreover, another expectation was to increase the number of times a track is played up to 67 million. TTNET Müzik web application is only available in Turkey. Due to the IP restrictions, its web application is not accessible from other countries. But its mobile application has no such restrictions and can be used worldwide.

2. Significance of the Problem and Motivation

Recommendation systems suffer from three major problems; cold start, huge data and sparsity. Cold start is a potential problem in computer-based information systems which involve a degree of automated data modelling. Specifically, it concerns the issue that the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information. There are basically two types of this problem. The first one occurs when a new user starts using the system. The system knows little about their preferences and it is necessary to pick some training points for rating so that the system begins learning what the user wants [6]. The second type occurs when a new product is introduced to the system. Again it is essential to rate this item in order to quickly improve the prediction accuracy about it [6]. The next problem is huge data. In many of the environments that these systems make recommendations in, there are millions of users and products. It is quite a challenge to produce high quality recommendations and perform many recommendations per second for millions of customers and items. Thus, a large amount of computation power is often necessary to calculate recommendations. The runtime and memory usage of existing algorithms can still be improved. The third problem is sparsity. Users that are very active contribute to the rating for a few number of items available in the database and even very popular items are rated by only a few number of users available in the database. Because of sparsity, it is possible that the similarity between two users cannot be defined, rendering collaborative filtering useless. The rating matrix R, which is mentioned in the previous section, might be full of missing entries since many users vote for just a few of the items. It is obvious that the companies having excellent recommendation systems are those with a lot of user data: Google, Amazon and Netflix and so forth.

We would like to work on recommendation systems because it is currently a hot topic. Of the main reasons why we chose this project, two in particular stand out. To begin with, a lot of web or mobile applications use this system to improve their popularity in social area. User-oriented technologies are more important parts of the social area. The new trend is to get feedback from the users and perform active learning on their behaviors or habits. This type of interaction resembles real world and it is more efficient to increase the popularity of items through making suggestions. The second reason is academic. We are interested in machine learning, artificial intelligence and data mining, so we would like to gain experience in these fields. The improvements in recommendation systems include a closer look at certain algorithms and mathematical notions. We believe that this subject will be a milestone for a quick introduction to academic researches.

There are many big companies which handle this problem effectively as we explained in the first part. The conflict for the existing systems is the big data. Having a large dataset is important to have more accurate predictions. But big data means a more complex set of computations. At this point, the existing algorithms suffer from the big data. Besides, the root mean square error for the predictions can be further improved, therefore there are many open problems in this area such as context-awareness. Apart from this, some companies still don't have effective recommendation system such as TTNET Müzik website and mobile application. We are planning to improve TTNET's recommendation features. After this project is completed, TTNET Müzik will have better suggestions to its users. In this way, both the company and users are affected positively. TTNET Müzik will be more successful at urging people to buy and download songs legally. We will also benefit from learning many new algorithms and platforms.

The data of our project are retrieved from an existing commercial product. We aim to develop the recommendation algorithm of this product. The main steps we will follow are literature survey, documentation, algorithm design, algorithm improvement, testing and deployment. After the project is completed, we want to turn it into an academic work if we can come a long way and bring a new perspective to this field.

3. Draft Project Plan

Our aim in this project is to improve the existing recommendation system in such a way that the major problems discussed in the previous section will almost be overcome. The existing system is at the moment TTNET Müzik application but it will become definite after the meeting with the advisors. In order to keep track of contemporary models and software tools, we will be using Neo4j, NoSQL, and Mahout. Neo4j is an open-source graph database implemented in Java. Its community edition is licensed under GNU General Public License. It is much faster than relational databases. NoSQL provides is a mechanism for storage and retrieval of data that employs less constrained consistency models than traditional relational databases. We will use it because it is highly optimized for big data and real-time web applications. Apache Mahout is a tool for collaborative filtering, classification and clustering. It consists of machine learning algorithms. The suggested model for this project is MapReduce. It is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster. We will also study Google's map-reduce algorithm, other learning algorithms such as SVM and as a starting point we will examine the neighborhood based models and hybrid models. Later on, we will study graph algorithms and databases which have evolved recently. Finally, we are aiming to make contributions to the problem of higher dimensions due to context.

The end product won't be a standalone product. TTNET already uses a recommendation system, but we will develop it because the current algorithm is not sufficient. After we complete the project, there will not be a major user interface oriented difference in TTNET Müzik application but the users will receive more realistic and reliable recommendations.

The distribution of the major tasks among project members is shown in the table below:

Task	**Members**
Literature Survey & Documentation	Duygu Kabakcı, Işınsu Katırcıoğlu, Sıla Kaya, Koray Kocakaya
Database Design	Duygu Kabakcı, Işınsu Katırcıoğlu
UI Interactions	Sıla Kaya, Koray Kocakaya
Algorithm Design	Duygu Kabakcı, Işınsu Katırcıoğlu, Sıla Kaya, Koray Kocakaya
Testing and Deployment	Duygu Kabakcı, Işınsu Katırcıoğlu, Sıla Kaya, Koray Kocakaya
Error Estimation	Duygu Kabakcı, Işınsu Katırcıoğlu, Sıla Kaya, Koray Kocakaya

The diagrams below are all designed according to the putative TTNET Müzik dataset and the intended recommendation system. TTNET Müzik provides users to play songs and to apply basic functionalities on these songs (Figure 1). Users can rate an item and like an item to create their own profile sufficiently so that she/he can get accurate recommendations. Users can also search items by keywords such as album title, singer or released year. If users want to purchase an item, he/she can download it legally on the system. Moreover, TTNET Müzik gives an opportunity to tag items. Users can generate lists by gathering songs together which provide significant data to our algorithm for classifying items and estimating similar songs. These music lists can also be viewed by other users to discover new songs. All these transactions give insight about user's taste of music so that the system makes more reliable recommendations. On the other hand, operators can add or remove items (songs or albums) to and from the system. They can also classify items according to their contents before uploading them to the system. Another important operator mission is collecting data from user activities for recommendation algorithm. They gather all the data from user transactions such as rating, liking, purchasing an item so on. Operators make suggestions to users based on the results of our estimation system.

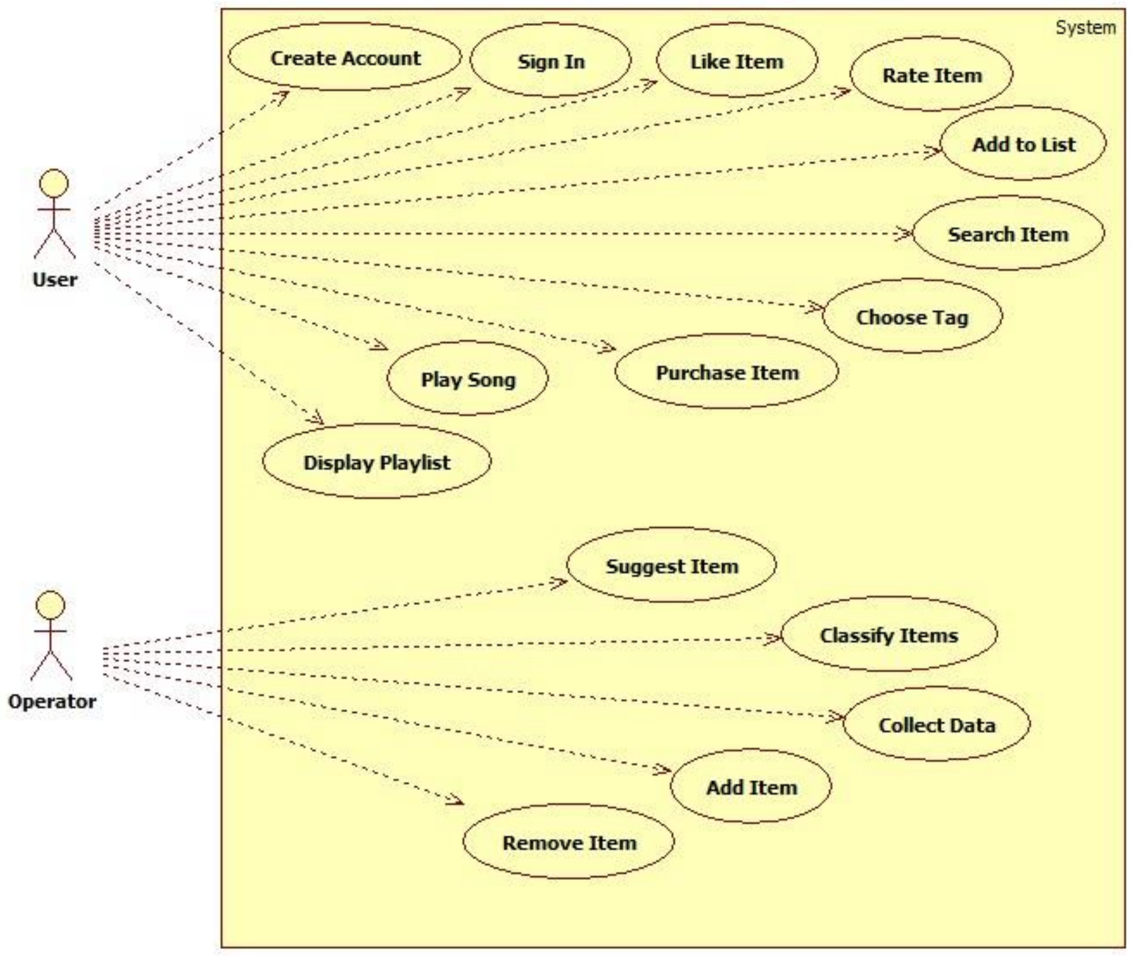

Figure 1: Use Case Diagram

The UI package of the product that our recommendation system will be integrated to already exists and we will not be dealing with UI design unless we encounter some incompatibility. But we will be dealing with UI and Business Logic interactions. The View module must be consistent with PC, Smart Phone and Tablet PC. The MVC architecture is roughly shown in Figure 2. The Business Logic package mainly consists of our algorithms. Since we will be working on big data, graph algorithm component is necessary. It provides faster computation. Machine learning algorithm component allows us to learn the behavior of user and train the user data to construct a rule. This component might include Bayesian statistics, SVM, decision trees or nearest neighbor approach. Accuracy estimation algorithm component allows to derive the degree of belief in our method. Mahout library component provides clustering, classification and pattern mining algorithms. All of these algorithms can deal with big data that reside on our NoSQL database in the database package. In case that there are previously constructed relational databases, we will convert them into non-relational databases. Because they can express queries as traversals and support semi-structured information as well. We will use Neo4j as a NoSQL database and we are planning to implement our algorithm packages in Java unless otherwise specified.

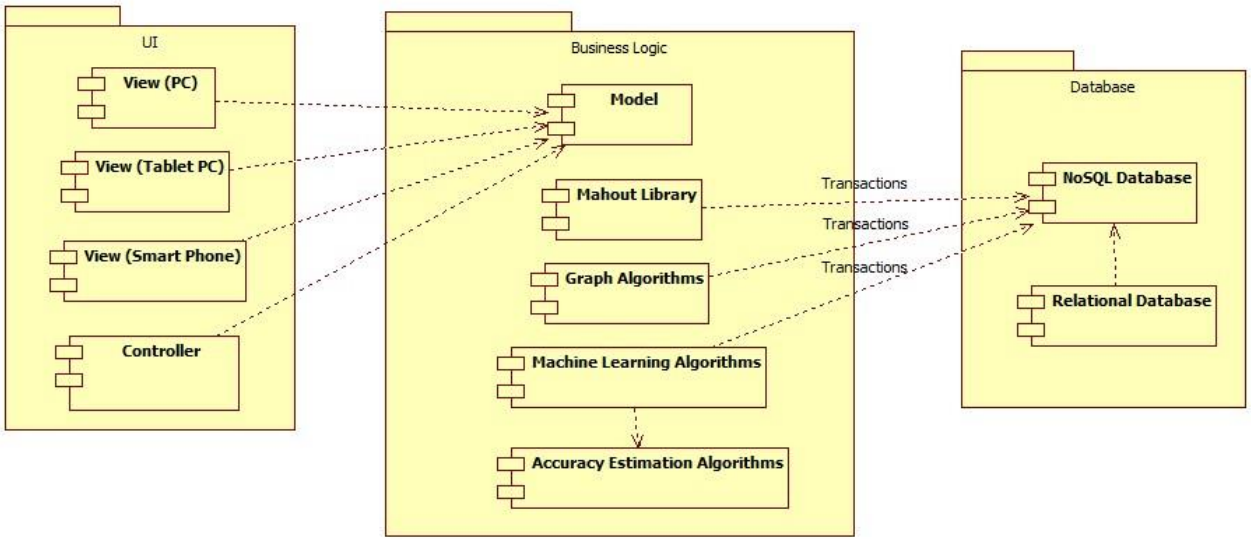

Figure 2: Component Diagram

We can use TTNET Müzik application via PC, Tablet PC and Smart Phone (Android and IOS). Our final algorithm will be compatible with these three hardware components (Figure 3). The PC product must be compatible with Unix/Windows operating systems. We are going to use map-reduce as a programming model, Neo4j as a graph database, Mahout to produce free implementations and NoSQL to store big data. Our large scale databases will be located in the server. Due to the big data size, we will be provided a computer cluster to which other hardware components belong and interact. Moreover, we are going to implement our project with Java unless otherwise specified.

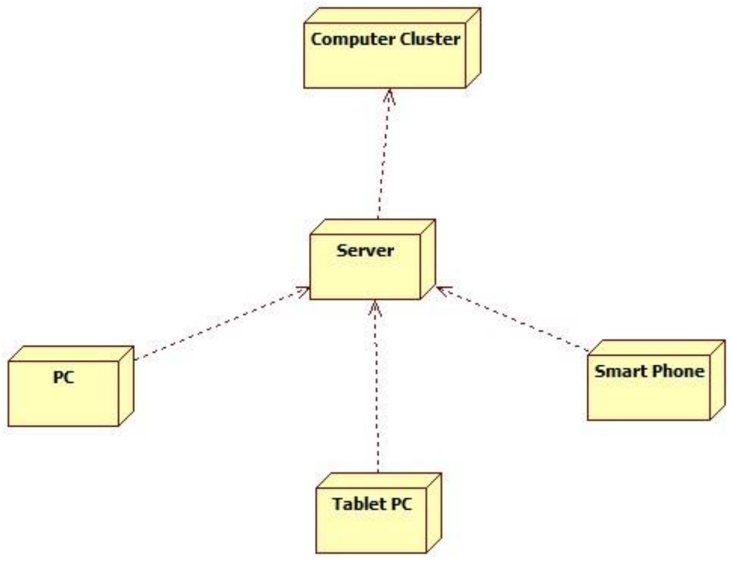

Figure 3: Deployment Diagram

4. Support

In this project, we will be supported by Prof. Dr. İsmail Hakkı Toroslu, Dr. Güven Fidan and AGMLab.

Prof. Dr. İsmail Hakkı Toroslu is a faculty member at the Department of Computer Engineering, METU. He has acknowledged expertise in Databases, Algorithms, Data Mining, Web Mining, Graph Algorithms, Logic Programming and Query Processing Algorithms. He is the leading advisor in this project and we will get academic support from him.
Contact Info: toroslu@ceng.metu.edu.tr

Dr. Güven Fidan is the Co-Founder of AGMLab. He and his colleagues in R&D team will provide us the big data that we will work on. AGMLab team will also indicate the requirements for developing the recommendation system for the given data.
Contact Info: guven.fidan@agmlab.com
Address: ODTÜ Teknokent, Silikon Binası Kat: 1
No: 25 06531 - ODTÜ | Ankara / TÜRKIYE
Tel: 0 312 210 13 40

All rights of the end product are reserved. We will make contributions to an existing recommendation system and further regulations about the intellectual property rights of the end-product will be mentioned in SRS report.

5. References

[1] Herlocker, J., Konstan, J. A., & Riedl, J. (2002). An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. *Information Retrieval, 5*(4), 287-310. doi:10.1023/A:1020443909834

[2] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems, 22*(1), 5-53. doi:10.1145/963770.963772

[3] Jafarkarimi, H., Sim, A. T. H., & Saadatdoost, R. (2012). A Naïve Recommendation Model for Large Databases. *International Journal of Information and Education Technology, 2*(2), 216-219. doi:10.7763/IJIET.2012.V2.113

[4] Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing, 7*(1), 76-80. doi:10.1109/MIC.2003.1167344

[5] Pandora. (2005). *About The Music Genome Project*. Retrieved October 8, 2013, from http://www.pandora.com/about/mgp

[6] Rubens, N., Kaplan, D., & Sugiyama, M. (2011). Active Learning in Recommender Systems. In P. B. Kantor, F. Ricci, L. Rokach & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 735-767). New York, USA: Springer.

[7] Sarwar, B. M., Karypis, G., Konstan, J., & Riedl, J. (2002). *Recommender Systems for Large-scale E-Commerce: Scalable Neighborhood Formation Using Clustering*. Paper presented at the 5th International Conference on Computer and Information Technology (ICCIT), 27-28 December 2002. Retrieved from http://grouplens.org/papers/pdf/sarwar_cluster.pdf

[8] Töcher, A., Jahrer, M., & Legenstein, R. (2008). *Improved Neighborhood-Based Algorithms for Large-Scale Recommender Systems*. Paper presented at the 2nd Netflix-KDD Workshop, 24 August 2008. Retrieved from http://dl.acm.org/citation.cfm?id=1722153