A SOCIAL CONTEXT BASED PRE-FILTERING ALGORITHM FOR CONTEXT AWARE RECOMMENDER SYSTEMS

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree of

INTEGRATED DUAL DEGREE

(Bachelor of Technology & Master of Technology)

in

COMPUTER SCIENCE AND ENGINEERING

(With Specialization in Information Technology)

By

ABHINAV DUGGAL

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JUNE, 2012

CANDIDATE'S DECLARATION

I hereby declare that the work being presented in the dissertation work entitled "**A Social Context Based Pre-filtering algorithm for Context Aware Recommender Systems**" towards the partial fulfillment of the requirement for the award of the degree of **Integrated Dual Degree in Computer Science and Engineering (with specialization in Information Technology)** and submitted to the **Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period from May, 2011 to June, 2012 under the guidance and provision of **Dr. Manoj Misra, Professor, Department of Electronics and Computer Engineering, IIT Roorkee and Mr. Ramesh Srinivasaraghavan, Adobe Systems, Bangalore** I have not submitted the matter embodied in this dissertation work for the award of any other degree and diploma.

Date: June, 2012
Place: Roorkee

Abhinav

(ABHINAV DUGGAL)

CERTIFICATE

This to certify that the declaration made by the candidate above is correct to the best of my knowledge and belief.

Date: June, 2012
Place: Roorkee

Ramesh

Mr. Ramesh Srinivasaraghavan
Adobe Systems,
Bangalore, India

M Misra

Dr. Manoj Misra
Professor, E&CE Dept.
IIT Roorkee, India

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my heartfelt gratitude to my guides and mentors **Dr. Manoj Misra**, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee and **Mr. Ramesh Srinivasaraghavan**, Adobe Systems, Bangalore, for their trust in my work, able guidance, regular source of encouragement and assistance throughout this dissertation work. I would state that the dissertation work would not have been in the present shape without their inspirational support and I consider myself fortunate to have done my dissertation under them.

I also extend my sincere thanks to **Dr. Padam Kumar**, Professor and Head of the Department of Electronics and Computer Engineering for providing facilities for the work.

I would like to thank all my friends who supported and encouraged me to finish this work.

Finally, I would like to say that I am indebted to my parents for everything that they have given to me. I thank them for sacrifices they made so that I could grow up in a learning environment. They have always stood by me in everything I have done, providing constant support, encouragement, and love.

Aduggal

ABHINAV DUGGAL

ABSTRACT

Context Aware computing has been one of the most challenging and interesting developments from the past decade. The term context may be defined as, "Any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."[3] Context awareness refers to the idea that computers can both sense, and react based on their environment. The importance of contextual information has been recognized by researchers and practitioners in many disciplines, including e-commerce personalization, information retrieval, ubiquitous and mobile computing, data mining, marketing, and management.

Along with context, the other topic of relevance to this work is that of recommender systems. Recommender Systems have been around for quite a while now with the most popular examples being that of Amazon[23] and Netflix that use collaborative filtering techniques to generate recommendations for their users. While a substantial amount of research has already been performed in the area of recommender systems, most existing approaches focus on recommending the most relevant items to users without taking into account any additional contextual information, such as time, location, or social circle etc. Despite some attempts being made at utilizing contextual information for generating recommendations, the problem remains largely unaddressed and tightly coupled with the base functionality of the service being provided.

In this work, we discuss how existing context aware systems exploit context and emphasize the relevance of this contextual information in recommender systems. We discuss the concepts of short-term and long-term context and how each of them can prove individually useful in the contextual pre-filtering and post-filtering processes of a context aware recommender system. We then discuss the notion of social context and introduce a novel pre-filtering algorithm using collaborative filtering techniques which exploits a user's social context, and provides a set of like-minded users to be used for generating recommendations. We will compare the performance of this algorithm with some existing techniques by evaluating the similarity of the set of users obtained in each case.

CONTENTS

CHAPTER 1: INTRODUCTION 1
1.1 Three major concepts 1
1.2 Problem Statement 2
1.2.1 Problem Description 2
1.3 Organization of Dissertation 3

CHAPTER 2: LITERATURE REVIEW 4
2.1 Context Awareness 4
2.1.1 Early Work in Context Awareness 4
2.1.2 Survey of Existing Context Frameworks and Systems 5
2.2 Recommender Systems 7
2.2.1 Approaches 7
2.2.2 Collaborative Filtering Algorithms 9
2.2.3 Examples 13
2.3 Context Aware 3D Recommender Systems 15
2.3.1 Modeling Contextual Information in Recommender Systems 16
2.3.2 Paradigms for incorporating Context 18
2.3.2.1 Contextual Pre-Filtering 18
2.3.2.2 Contextual Post-Filtering 19
2.3.4 Prior Work 19
2.4 Research Gaps 21

CHAPTER 3: PROPOSED SOLUTION 22

3.1 Context Definition and Hierarchy 22

3.1.1 Notion of Short-Term and Long-Term Context 24

3.2 Scheme for providing Context Aware Recommendations 25

3.3 Algorithms for Contextual Pre-Filtering 28

3.3.1 Maximum Mutual Items 28

3.3.2 Cosine Similarity 29

3.3.3 Proposed Page Popularity Rating Algorithm 30

3.3.4 Proposed Multidimensional Approach Algorithm 31

CHAPTER 4: IMPLEMENTATION DETAILS 32

4.1 Resources Used 32

4.1.1 Facebook Platform Javascript SDK and PHP SDK 32

4.1.2 Facebook Graph API 32

4.1.3 Apache Server and MySQL 33

4.2 Application Module: Check Mate 33

4.3 Data Gathering Module 35

4.3.1 Implicit Data Collection 35

4.3.2 Building a real dataset using Facebook 36

4.3.3 Dataset statistics 38

4.4 Evaluation Module 38

4.4.1 Explicit Data Collection 38

4.4.2 Evaluation Paradigms 39

CHAPTER 5: RESULTS AND DISCUSSION 40

5.1 System Configuration 40
5.2 Evaluation Metrics 40
5.3 Performance Evaluation 41
5.3.1 User Set Evaluation 41
5.3.1.1 Maximum Mutual Items 41
5.3.1.2 Cosine Similarity 42
5.3.1.3 Page Rating 42
5.3.1.4 Multidimensional Approach 43
5.3.1.5 Average MAE comparison of all sets 43
5.3.2 Prediction Evaluation 44

CHAPTER 6: CONCLUSION AND FUTURE WORK 45
6.1 Suggestions for Future Work 45

REFERENCES 48
PUBLICATIONS 52
APPENDIX A: EXPLICIT DATA COLLECTED A-1

LIST OF FIGURES AND TABLES

Figure 2.1	Collaborative Filtering Process	11
Figure 2.2	Multidimensional model for the User × Item × Time recommendation space.	17
Figure 2.3	Paradigms for Incorporating Context	18
Figure 3.1	Context Hierarchy	22
Figure 3.2	Workflow of Proposed Contextual Pre-filtering	26
Figure 3.3	Proposed Context-Aware Recommender Scheme	27
Figure 3.4	Screenshot from 'Discover Facebook Pages'	28
Figure 4.1	Fetch Object using Graph API	33
Figure 4.2	Variation of no. of mutual likes with no. of mutual friends	33
Figure 4.3	Screenshot of Application Module : Check Mate deployed on Facebook	34
Figure 4.4	Information Flow in Checkmate	35
Figure 4.5	Dataset Structure	36
Figure 4.6	Workflow of Incremental Update	37
Figure 4.7	Categorization of Page objects fetch using Check Mate	38
Figure 5.1	Per User MAE for Maximum Mutual Likes	41
Figure 5.2	Per User MAE for Cosine Similarity	42
Figure 5.3	Per User MAE for Page Rating	42
Figure 5.4	Per User MAE for Multidimensional approach	43
Figure 5.5	Comparison of Average MAE of Top-N users using 4 algorithms	44
Figure 5.6	Comparison prediction performance of 4 algorithms	44
Figure 6.1	Building Context Ontology	46
Figure 6.2	A partial publish/consume Context API	47

Table 2.1	Comparison of Context-Aware Systems	6
Table 2.2	Overview of Memory-based Collaborative Filtering Algorithms	10
Table 2.3	Comparison of popular Similarity Measures	13
Table 2.4	Recommender System Examples	15
Table 4.1	Dataset Statistics	38
Table 4.2	User-Rating Matrix	39

CHAPTER 1 INTRODUCTION

1.1 Three Major Concepts

In our work, we shall be using three major concepts of Context Awareness, Recommender Systems and the more recent concept of 3D context aware recommender systems.

Context Awareness - The way humans interact with other humans is influenced not only by the content being discussed but also by the way one human perceives the other, as well as the current situational information in which they are conversing. In other words, context plays an enormous role in determining the contents of any interaction in our daily life. In the same way, a human-computer dialogue can also be significantly improved if computers are able to access user context intensively. Context awareness refers to the idea that computers can both sense, and react based on their environment. Devices may have information about the circumstances under which they are able to operate and based on rules, or an intelligent stimulus, react accordingly. As it happens, for computers, unlike humans, context has to be defined explicitly. While there have been many attempts at defining context [1-3], most of them limit the scope of context definition by restricting it to a particular example. The term context-awareness in ubiquitous computing was introduced by Schilit [1] in 1994. With the new generation of social networking, people are making available a lot of personal information. There is a need to exploit this information using which context can be mined from entities such as status updates, tweets etc.

Recommender systems - They became an important research area since the appearance of the first papers on collaborative filtering since the mid-1990s [30]. There has been much work done both in the industry and academia on developing new approaches to recommender systems over the last decade. The interest in this area still remains high because it constitutes a problem rich research area and because of the abundance of practical applications that help users to deal with information overload and provide personalized recommendations, content and services to them. Examples of such applications include recommending books, CDs and other products at sites such as Amazon.com [23].

Contextually Aware Recommendations- While a substantial amount of research has already been performed in the area of recommender systems, most existing approaches focus on recommending the most relevant items to users without taking into account any additional contextual information, such as time, location, or the company of other people. Alexander et al.[5] argue that relevant contextual information does matter in recommender systems and that it is important to take this information into account when providing recommendations. Context Aware Recommendations are generally provided in two paradigms. Pre-filtering, where data is pre-filtered based on user's context variable and Post-filtering, where recommendations provided using a traditional recommender system are made contextually relevant.

1.2 Problem Statement

The aim is to develop and implement a novel social context based pre-filtering algorithm for context aware recommender systems and propose a recommender system using notions of short-term and long-term context.

1.2.1 Problem Description

As it turns out, context is a poorly used source of information which, most of the times is utilized inefficiently due to impoverished understanding of the term, an example is this is the inability of context aware systems to efficiently exploit social networks. At the same time, recommender systems suffer from two major issues – scalability and quality of recommendations. We will use a user's long-term social context as a pre-filtering step to provide context aware recommendations. In the process, we will be addressing issues of the need to propose a new context hierarchy and significantly reducing user-space for generating recommendations thereby addressing scalability while observing that quality of recommendations does not suffer. A novel algorithm will be implemented and evaluated to determine the set of most similar users to be used for generating recommendations.

Hence the goal stated in the problem statement can be divided into three sub-problems:

- To propose a holistic context hierarchy that can exploit short-term and long-term context.
- Proposing a scheme for providing context aware recommendations using notions of short-term and long-term context.
- Proposing a novel algorithm for social context based pre-filtering to address various issues in context aware recommender systems.

1.3 Organization of Dissertation

This report comprises of six chapters including this chapter that introduces the topic and states the problem. The rest of the dissertation report is organized as follows.

Chapter 2 gives a literature survey of the three concepts we have discussed till now. Firstly, we will look at the work done in context awareness and a survey performed will show how systems have exploited context till date. Then we discuss recommender systems, the approaches they follow and some important examples. We discuss collaborative filtering in detail. Then we move onto Context Aware 3D Recommender Systems and discuss vital processes of Contextual Post-filtering and Contextual Pre-filtering.

Chapter 3 provides a detailed description of the assumed context hierarchy, system framework and the proposed scheme for providing context aware recommendations and the four algorithms used for contextual pre-filtering.

Chapter 4 gives the brief description of the implementation of the proposed scheme.

Chapter 5 discusses the results and including discussion on them.

Chapter 6 concludes the work and gives the directions for future work.

CHAPTER 2
LITERATURE REVIEW

2.1 Context Awareness

Several researchers have tried to give a holistic definition of what context entails, though no widely accepted definition exists, some of the popular definitions describe context as follows:

"Context is the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user."[2]

"Any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and application themselves."[3]

The applications that use context information are called Context aware applications and the property is called Context Awareness. Context can be of several types. The most significant types are listed below: User Context, Physical Context and Time Context [3]

2.1.1 Early Work in Context Awareness

Work on context awareness started as an extension of mobile computing when Schilit[1] started with an initial idea that a person's location could be used as his context and could significantly enhance his experience. His idea, though limited to just the location of a person sparked a lot of interest among researchers who set out to define and model user context. Schilit[1] later on categorized context as Computing, User and Physical Context. Chen and Kotz [2] talk about Active and Passive Context. Device and Service Context [7] were also introduced. Other important categories that came into being with time were Time Context and Activity Context.

But the problem with these categories of context was that most of them were situational (based on sensor data) and failed to carry information past a certain timeframe. To be able to talk about a user effectively, taking into account only the current sensor data might prove vague and insufficient. So Jameson[6] talks about modeling both the context and the user whereby we take into account a user's behavior and his longer term preferences along with the features of his current situation. Therefore, the problem of context representation was tightly coupled with that of user modeling.

As the umbrella of context spread, one important issue that needed to be addressed was as to how we would come up with a framework that would gather this information and deliver personalization services. So, now, we address the various frameworks that were built to tackle this problem.

2.1.2 Existing Context Frameworks and Systems

The pioneering context-aware systems like Active Badge [8] and Xerox PARC [9] were basically location aware systems as they were only aware of the locality. Tour guide systems like GUIDE [10] and Cyberguide [11] evolved the concept of context by adding temporal information in addition to spatial information. These systems are primarily context-aware application designed to provide better and customized services to their users. The Context Toolkit [4] provides an Application Programming Interface (API) to develop context-aware applications but is limited to tightly coupled 'Widgets' that directly access the hardware contextual data sensors. But all in all, these older context aware systems neither address the issue of taking user preference into account nor do they talk about the importance of contextual history in making decisions.

The recent context-aware systems are frameworks that provide context-awareness through rich ontology based context representation. This rich context ontology considers parameters relevant to an interaction as the context. Gaia [12] (a CORBA based distributed operating system) and CAMUS [13] (a JINI based service oriented framework) provides context-aware service delivery limited only to context-aware applications. CoBrA [14] is a mobile agent based framework that dispatches mobile agents to gather context information from the sensors in the environment. CAPEUS [15] uses a document based approach that exchanges context-aware packets that describe service requests. CAPP [16] is a service oriented architecture that provides context-aware service discovery for mobile users.

The history of information is maintained as a contextual database in SOCAM [17], Gaia, CASS [18] and CoBrA. CAML has been proposed to highlight that context adaptation should be the research issue rather than context awareness [19]. SODA is a decentralized system designed as a Service Oriented Architecture (SOA) [20]. Context Management Framework (CMF) and Hydrogen both lack history of information as well as an Ontology based information representation technique [21 and 22]. CAPP is also a centralized architecture but lacks both history of information and conflict resolution [16].

Table 2.1 depicts the comparison of these systems on the basis of the way in which these systems exploit context. All these systems have in one way or the other, talked about a use case that best fits the description of the framework provided by them.

Table 2.1 – Comparison of Context-Aware Systems

	ACQUIRED CONTEXT	RICH CONTEXT	EXAMPLE USE CASES
ACTIVE BADGE [8]	Location	No	Friend Finder
GUIDE[10]/CYBERGUIDE[11]	Location, Orientation	No	Tour Guide, Maps
CONTEXT TOOLKIT[4]	Location, Identity, Time	Yes	In/out, DUMMBO*
CAPEUS[15]	Location, Proximity	Yes	Triggered printing
GAIA[12]	Location, Device, Activity, Temperature, Sound, Social	Yes	User-centric Active Space, CFS**
COBRA[14]	Physical Sensor Data, Situational roles and beliefs	Yes	Intelligent Meeting Room
HYDROGEN[15]	Physical, Logical	Yes	Context Aware Postbox
SOCAM[17]	Sensed Low Level, Manipulated High Level	Yes	Smartphone Profiles
CASS[18]	Low Level, High Level	Yes	MALLET***
CAML[19]	Identity, Spatio-Temporal, Facility, Activity, Learner, Community	Yes	Synchronous Messaging
CAMUS[13]	Service, Semantic, Client	Yes	Service Provisioning
SODA[20]	Sensor Data, High Level Abstractions	Yes	Real-time logistics
CAPP[16]	Who, What, Where, When	Yes	Service Provisioning

*DUMMBO – Dynamic Ubiquitous Mobile Meeting Board.

**CFS - Context File System

***MALLET - Maintenance Assignment Listing Lightweight Electronic Tool

2.2 Recommender Systems

The other topic of research of importance to us in our work is that of recommender systems. Recommender systems apply knowledge discovery techniques to the problem of making personalized recommendations for information, products or services during a live interaction [24]. These systems have become extremely popular in recent years after their use by E-commerce giants such as Amazon.com [23] and Netflix. The products can be recommended based on the top overall sellers on a site, based on the demographics of the customer, or based on an analysis of the past buying behavior of the customer as a prediction for future buying behavior. Broadly, these techniques are part of personalization on a site, because they help the site adapt itself to each customer. Broadly there are three ways [25] in which recommender systems enhance E-commerce sales:

1) **Browsers into buyers** – Visitors to a Web site often look over the site without ever purchasing anything. Recommender systems can help customers find products they wish to purchase.
2) **Cross-sell** - Recommender systems improve cross-sell by suggesting additional products for the customer to purchase. If the recommendations are good, the average order size should increase.
3) **Loyalty** - Recommender systems improve loyalty by creating a value-added relationship between the site and the customer. Sites invest in learning about their users, use recommender systems to operationalize that learning, and present custom interfaces that match customer needs.

Let us now discuss the various approaches followed by recommender systems in brief.

2.2.1 Approaches

Recommender systems use mostly two approaches in providing recommendations – Collaborative Filtering or Content-based filtering.

1) **Content-Based Filtering** - Content-based recommender systems make recommendations by analyzing the content of textual information and finding regularities in the content [26]. Content-based filtering methods are based on information about and characteristics of the items that are going to be recommended. In other words, these algorithms try to

recommend items that are similar to those that a user liked in the past (or is examining in the present). In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended. There are two major ways of doing Content-Based Filtering:

a. **Bayesian Classifiers** - Bayesian networks create a model based on a training set with a decision tree at each node and edges representing user information. The model can be built online over a matter of hours or days [24]. Bayesian networks may prove practical for environments in which knowledge of user preferences changes slowly with respect to the time needed to build the model but are not suitable for environments in which user preference models must be updated rapidly or frequently.

b. **Clustering** - Clustering techniques work by identifying groups of users who appear to have similar preferences.[24] Once the clusters are created, predictions for an individual can be made by averaging the opinions of the other users in that cluster. Some clustering techniques represent each user with partial participation in several clusters. The prediction is then an average across the clusters, weighted by degree of participation. Clustering techniques usually produce less-personal recommendations than other methods, and in some cases, the clusters have worse accuracy than nearest neighbor algorithms.

A key issue with content-based filtering is whether the system is able to learn user preferences from user's actions regarding one content source and use them across other content types. When the system is limited to recommending content of the same type as the user is already using, the value from the recommendation system is significantly less than when other content types from other services can be recommended. For example, recommending news articles based on browsing of news is useful, but it's much more useful when music, videos, products, discussions etc. from different services can be recommended based on news browsing.

2) **Collaborative Filtering** – Collaborative Filtering is the most widely used approach for providing online recommendations and this approach is what we'll be using for our work as well. Collaborative filtering methods are based on collecting and analyzing a large amount of information on users' behaviors, activities or preferences and predicting what

users will like based on their similarity to other users. User-based collaborative filtering attempts to model the social process of asking a friend for a recommendation. A key advantage of the collaborative filtering approach is that it does not rely on machine analyzable content and therefore it is capable of accurately recommending complex items such as movies without requiring an "understanding" of the item itself. One of the most famous examples of Collaborative Filtering is item-to-item collaborative filtering (people who buy x also buy y), an algorithm used by Amazon.com's recommender system [23].

Collaborative filtering techniques use a database of preferences for items by users to predict additional topics or products a new user might like. In a typical CF scenario, there is a list of m users $\{u_1, u_2 \ldots u_m\}$ and a list of n items $\{i_1, i_2, \ldots, i_n\}$, and each user, u_i, has a list of items, Iu_i, which the user has rated, or about which their preferences have been inferred through their behaviors.

The major difference between CF and content-based recommender systems is that CF only uses the user-item ratings data to make predictions and recommendations, while content-based recommender systems rely on the features of users and items for predictions. Both content-based recommender systems and CF systems have limitations. While CF systems do not explicitly incorporate feature information, content-based systems do not necessarily incorporate the information in preference similarity across individuals [27]. Also, content-based systems are generally more complex since they require an understanding of the item itself.

For our work, we will focus on using collaborative filtering techniques and therefore discuss various CF-based algorithms in the following section.

2.2.2 Collaborative Filtering Algorithms

Collaborative Filtering algorithms primarily fall into three categories [26]:

1) **Memory Based Collaborative Filtering** - Memory-based CF algorithms use the entire or a sample of the user-item database to generate a prediction. Every user is part of a group of people with similar interests. By identifying the so-called neighbors of a new user (or active user), a prediction of preferences on new items for him or her can be produced.

2) **Model Based Collaborative Filtering** – Rely on design and development of models (such as machine learning, data mining algorithms), which can allow the system to learn to recognize complex patterns based on the training data, and then make intelligent predictions for the collaborative filtering tasks for test data or real-world data, based on the learned models.
3) **Hybrid recommenders** - Hybrid CF systems combine CF with other recommendation techniques (typically with content-based systems) to make predictions or recommendations. We will discuss some examples of these in section 2.2.3

Table 2.2 depicts an overview of these algorithms. For our purpose we'll be discussing the Memory Based Collaborative Filtering technique in detail.

Table 2.2 – Overview of Memory-based Collaborative Filtering Algorithms [26]

CF categories	Representative techniques	Main advantages	Main shortcomings
Memory-based CF	-Neighbor-based CF (item-based/user-based CF algorithms with Pearson/vector cosine correlation) -Item-based/user-based top-N recommendations	-easy implementation -new data can be added easily and incrementally -need not consider the content of the items being recommended -scale well with co-rated items	-are dependent on human ratings -performance decrease when data are sparse -cannot recommend for new users and items -have limited scalability for large datasets
Model-based CF	-Bayesian belief nets CF -clustering CF -MDP-based CF -latent semantic CF -sparse factor analysis	-better address the sparsity, scalability and other problems -give an intuitive rationale for Recommendations -improve prediction performance	-expensive model building -lose useful information for dimensionality reduction techniques -have trade-off between prediction performance and scalability
Hybrid Recommenders	-content-based CF recommender -content-boosted CF -hybrid CF combining memory-based and model-based CF algorithms, for example, Personality Diagnosis	-overcome limitations of CF and content-based or other Recommenders -improve prediction performance -overcome CF problems such as sparsity	-have increased complexity and expense for implementation -need external information that usually not available

Memory Based Collaborative Filtering - By identifying the so-called neighbors of a new user (or active user), a prediction of preferences on new items for him or her can be produced.

The neighborhood-based CF algorithm, a prevalent memory-based CF algorithm, uses the following steps: (i) calculate the similarity or weight, $sim_{i,j}$, which reflects distance, correlation, or weight, between two users or two items, i and j; (ii) produce a prediction for the active user by taking the weighted average of all the ratings of the user or item on a certain item or user, or using a simple weighted average [24]

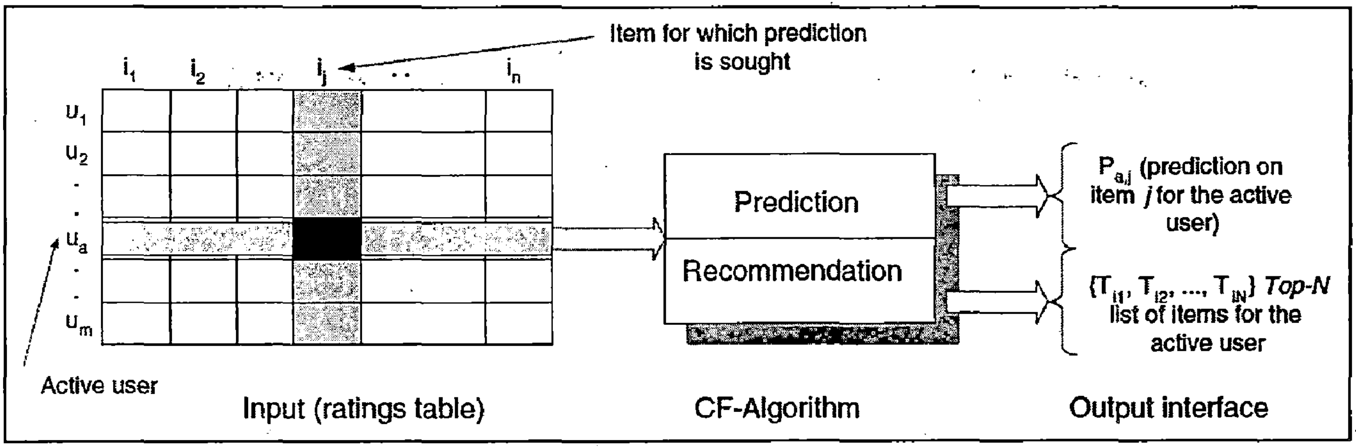


Figure 2.1 – Collaborative Filtering process [24]

As Figure 2.1 shows, the Collaborative filtering process can do two things, predict the rating a user is likely to give to a certain item, or generate a top N list of items a user is most likely to purchase. To be able to do both, the following steps are essential:

1) Similarity Computation - Similarity computation between items or users is a critical step in memory-based collaborative filtering algorithms. For item-based CF algorithms, the basic idea of the similarity computation between item i and item j is first to work on the users who have rated both of these items and then to apply a similarity computation to determine the similarity, $sim_{i,j}$, between the two co-rated items of the users [24]. For a user-based CF algorithm, we first calculate the similarity, $sim_{u,v}$ between the users u and v who have both rated the same items. There are many different methods to compute similarity or weight between users or items.
   a. Correlation Based Similarity - In this case, similarity $sim_{u,v}$ between two users u and v, is measured by computing the Pearson correlation or other correlation-based similarities. Pearson correlation measures the extent to which two variables linearly relate with each other. For the user based algorithm, the Pearson

correlation between users u and v is given by Equation (2.1). $r_{u,i}$ is the rating given by user u to item i and $\bar{r_u}$ is the average rating by user u, I is the whole set of items. For the item-based algorithm the set of items I is replaced by the set of users U and users u and v are replaced by items i and j.

$$sim_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r_u})(r_{v,i} - \bar{r_v})}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r_u})^2} \sqrt{\sum_{i \in I}(r_{v,i} - \bar{r_v})^2}} \qquad (2.1)$$

b. Cosine Based Similarity - In this case, two items are thought of as two vectors in the m dimensional user-space. The similarity between them is measured by computing the cosine of the angle between these two vectors $\vec{u}$ and $\vec{v}$. Formally, in the m×n ratings matrix in Figure 2.1, similarity between users u and v, denoted by $sim_{i,j}$ is given by Equation (2.2). Here again, to perform Item-based CF, similarity can be calculated by taking i and j vectors of products I and j, instead of user vectors u and v.

$$sim_{u,v} = \cos(\vec{u}, \vec{v}) = \frac{\vec{u}.\vec{v}}{||\vec{u}|| * ||\vec{v}||} \qquad (2.2)$$

c. Adjusted Cosine Similarity – This measure is mostly used in item-based CF algorithms. One fundamental difference between the similarity computation in user-based CF and item-based CF is that in case of user-based CF the similarity is computed along the rows of the matrix but in case of the item-based CF the similarity is computed along the columns, i.e., each pair in the co-rated set corresponds to a different user. Computing similarity using basic cosine measure in item-based case has one important drawback; the differences in rating scale between different users are not taken into account. The adjusted cosine similarity offsets this drawback by subtracting the corresponding user average from each co-rated pair. Formally, the similarity between items i and j using this scheme is given by Equation (2.3)

$$sim_{i,j} = \frac{\sum_{u \in U}(r_{u,i} - \bar{r_u})(r_{u,j} - \bar{r_u})}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r_u})^2} \sqrt{\sum_{u \in U}(r_{u,j} - \bar{r_u})^2}} \qquad (2.3)$$

Table 2.3 – Comparison of popular similarity measures.

Measure	Expression	Remarks
Correlation	$sim_{u,v} = \frac{\sum_{i\in I}(r_{u,i} - \bar{r_u})(r_{v,i} - \bar{r_v})}{\sqrt{\sum_{i\in I}(r_{u,i} - \bar{r_u})^2}\sqrt{\sum_{i\in I}(r_{v,i} - \bar{r_v})^2}}$	Does not satisfy Van Eck [41] conditions for similarity measures.
Cosine	$sim_{u,v} = \cos(\vec{u}, \vec{v}) = \frac{\vec{u}.\vec{v}}{\lVert\vec{u}\rVert * \lVert\vec{v}\rVert}$	Fairly simple and can be used for both binary and non-binary data.
Adjusted Cosine	$sim_{i,j} = \frac{\sum_{u\in U}(r_{u,i} - \bar{r_u})(r_{u,j} - \bar{r_u})}{\sqrt{\sum_{u\in U}(r_{u,i} - \bar{r_u})^2}\sqrt{\sum_{u\in U}(r_{u,j} - \bar{r_u})^2}}$	Does not satisfy Van Eck [41] conditions for similarity measures.
Jaccard	$sim_{u,v} = \frac{\lvert U \cap V\rvert}{\lvert U \cup V\rvert}$	Useful only for binary data. Cannot be used with our algorithms.
Dice	$sim_{u,v} = \frac{2.\lvert U \cap V\rvert}{\lvert U\rvert + \lvert V\rvert}$	Useful only for binary data. Cannot be used with our algorithms.
Jensen-Shennon	$sim_{u,v} = 1 - \frac{1}{2}\left(\sum_{i\in I} r_{u,i}.\log\frac{r_{u,i}}{\bar{r_u}}\right) - \frac{1}{2}\left(\sum_{i\in I} r_{v,i}.\log\frac{r_{v,i}}{\bar{r_v}}\right)$	High time complexity. Not useful for binary data.

2) Prediction Computation - The most important step in a collaborative filtering system is to generate the output interface in terms of prediction. Once we isolate the set of most similar items based on the similarity measures, the next step is to look into the target users ratings and use a technique to obtain predictions. Here we consider most commonly used technique [24] of using weighted sum.

$$P_{a,i} = \bar{r_a} + \frac{\sum_{u\in U}(r_{u,i} - \bar{r_u}).sim_{a,u}}{\sum_{u\in U}\lvert sim_{a,u}\rvert} \qquad (2.4)$$

For our purpose, as we shall discus later, we have used Cosine based similarity and user-based Collaborative Filtering algorithms. The prime reason for choosing cosine based similarity being binary matrices generated using social network mining.

2.2.3 Examples

We now discuss some recommender systems and then later discuss the type of technology used to provide recommendations.

1) **Amazon.com –**

   a. Customers who bought also bought - The Customers Who Bought feature is found on the information page for each book in their catalog [25]. It is in fact two separate recommendation lists. The first recommends books frequently purchased by customers who purchased the selected book. The second recommends authors whose books are frequently purchased by customers.

b. Eyes - The Eyes feature allows customers to be notified via email of new items added to the Amazon.com catalog.

c. Amazon.com Delivers - Amazon.com Delivers is a variation on the Eyes feature. Customers select checkboxes to choose from a list of specific categories/genres (Oprah books, biographies). Periodically the editors at Amazon.com send emails to notify subscribers of their latest recommendations in the subscribed categories.

d. Book Matcher - The Book Matcher feature allows customers to give direct feedback about books they have read. Customers rate books they have read on a 5-point scale from “hated it” to “loved it.”

e. Customer Comments - The Customer Comments feature allows customers to receive text recommendations based on the opinions of other customers.

2) CDNOW

a. Album Advisor – In the Album Advisor feature of CDNOW (www.cdnow.com) customers locate the information page for a given album. The system recommends 10 other albums related to the album in question.

b. My CDNOW: My CDNOW enables customers to set up their own music store, based on albums and artists they like.

3) eBay

a. Feedback Profile - The Feedback Profile feature at eBay.com™ (www.ebay.com) allows both buyers and sellers to contribute to feedback profiles of other customers with whom they have done business. The feedback consists of a satisfaction rating (satisfied/neutral/dissatisfied) as well as a specific comment about the other customer.

4) Moviefinder.com

a. Match Maker - Moviefinder.com’s Match Maker (www.moviefinder.com) allows customers to locate movies with a similar “mood, theme, genre or cast” to a given movie.

b. We Predict - We Predict recommends movies to customers based on their previously indicated interests. Customers enter a rating on a 5-point scale -- from

A to F – for movies they have viewed. These ratings are used in two different ways. Most simply, as they continue, the information page for non-rated movies contains a personalized textual prediction (go see it – forget it).

Table 2.4 below illustrates the interface and technology used by above discussed systems.

Table 2.4 – Recommender System Examples

Business/Application	Recommendation Interface	Recommendation Technology
AMAZON.COM		
Customers who Bought	Similar Item	Item to Item CF
Eyes	Email	Attribute Based
Amazon.com Delivers	Email	Attribute Based
Book Matcher	Top N List	People to People CF
Customer Comments	Average Rating	Aggregated Rating
CDNOW		
Album Advisor	Similar Item (Top N List)	Item to Item CF
My CDNOW	Top N List	People to People CF
eBay		
Feedback Profile	Average Rating	Text Comments Aggregated Rating
Moviefinder.com		
Match Maker	Similar Item	Item to Item Correlation
We Predict	Top N List	People to People Correlation

2.3 Context Aware 3D Recommender Systems

The majority of existing approaches to recommender systems focus on recommending the most relevant items to individual users and do not take into consideration any contextual information[5], such as time, place and the company of other people (e.g., for watching movies or dining out). In other words, traditionally recommender systems deal with applications having only two types of entities, users and items, and do not put them into a context when providing recommendations.

However, in many applications, such as recommending a vacation package, personalized content on a Web site, or a movie, it may not be sufficient to consider only users and items – it is also important to incorporate the contextual information into the recommendation process in order to recommend items to users under certain circumstances. For example, using the temporal context, a travel recommender system would provide a vacation recommendation in the winter that can be very different from the one in the summer. Therefore, accurate prediction of consumer preferences undoubtedly depends upon the degree to which the recommender system has incorporated the relevant contextual information into a recommendation method [5].

2.3.1 Modeling Contextual Information in Context Aware recommender Systems

Traditionally, recommender systems use the function R given in (2.5) to estimate the rating of an item for a user. These systems deal with the User×Item space and therefore are considered as 2 dimensional restricted in user and item dimensions.

$$R : User \times Item \rightarrow Rating \qquad (2.5)$$

Context Aware recommender systems introduce an additional dimension to the computation as shown in (2.6).

$$R : User \times Item \times Context \rightarrow Rating \qquad (2.6)$$

Example 2.1 [5]: Consider the application for recommending movies to users, where users and movies are described as relations having the following attributes:

• Movie: the set of all the movies that can be recommended; it is defined as

Movie(MovieID, Title, Length, ReleaseYear, Director, Genre).

• User: the people to whom movies are recommended; it is defined as

User(UserID, Name, Address, Age, Gender, Profession).

Further, the contextual information consists of the following three types that are also defined as relations having the following attributes:

• Theater: the movie theaters showing the movies; it is defined as

Theater(TheaterID, Name, Address, Capacity, City, State, Country).

• Time: the time when the movie can be or has been seen; it is defined as

Time(Date, DayOfWeek, TimeOfWeek, Month, Quarter, Year).

Here, attribute DayOfWeek has values Mon, Tue, Wed, Thu, Fri, Sat, Sun, and attribute TimeOfWeek has values "Weekday" and "Weekend".

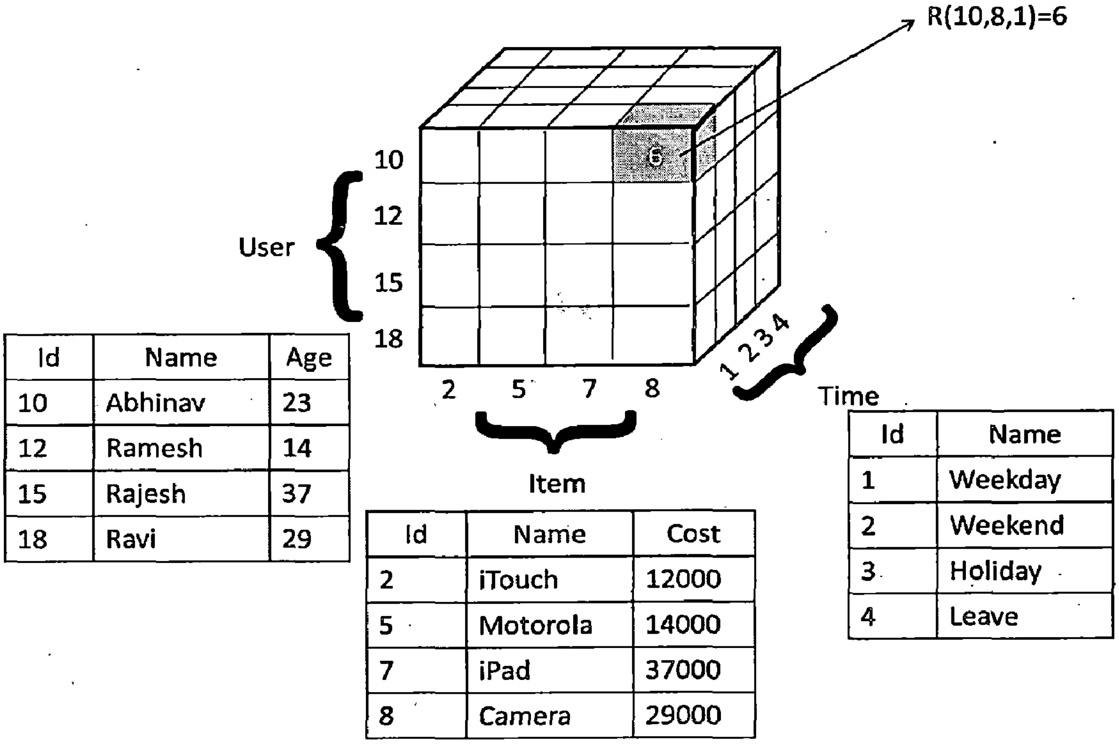
• Companion: represents a person or a group of persons with whom one can see a movie. It is defined as

Companion (companionType).

Here attribute companionType has values "alone", "friends", "family", "co-workers", and "others".

Then the rating assigned to a movie by a person also depends on where and how the movie has been seen, with whom, and at what time. For example, the type of movie to recommend to college student Abhinav can differ significantly depending on whether she is planning to see it on a Saturday night with her friends vs. on a weekday with her parents.

Contextual information was also defined in [30] as follows. In addition to the classical User and Item dimensions, additional contextual dimensions, such as Time, Location, etc., were. Formally, let $D_1, D_2, \ldots, D_n$ be dimensions, two of these dimensions being User and Item, and the rest being contextual. Each dimension D_i is a subset of a Cartesian product of some attributes (or fields) A_{ij}, ($j = 1, \ldots, k_i$), i.e., $D_i \subseteq A_{i1} \times A_{i2} \times \ldots \times A_{iki}$, where each attribute defines a domain (or a set) of values. Moreover, one or several attributes form a key, i.e., they uniquely define the rest of the attribute. In some cases, a dimension can be defined by a single attribute, and $k_i = 1$ in such cases. For example, consider the three-dimensional recommendation space User×Item×Time. Figure 2.2 clearly illustrates the difference in information modeling with traditional recommender systems as shown in Figure 2.1



Id	Name	Age
10	Abhinav	23
12	Ramesh	14
15	Rajesh	37
18	Ravi	29

Id	Name	Cost
2	iTouch	12000
5	Motorola	14000
7	iPad	37000
8	Camera	29000

Id	Name
1	Weekday
2	Weekend
3	Holiday
4	Leave

Figure 2.2 – Multidimensional model for the *User* × *Item* × *Time* recommendation space [5]

2.3.2 Paradigms for Incorporating Context

2.3.2.1 Contextual Pre-Filtering

The contextual pre-filtering approach uses contextual information to select or construct the most relevant 2D (User × Item) data for generating recommendations. As shown in Figure 2.3(a), the major advantage of this approach is that it allows deployment of any of the numerous traditional recommendation techniques previously proposed in the literature [30]. In particular, in one possible use of this approach, context c essentially serves as a query for selecting (filtering) relevant ratings data. An example of a contextual data filter for a movie recommender system would be: if a person wants to see a movie on Saturday, only the Saturday rating data is used to recommend movies. Note that this example represents an exact pre-filter. In other words, the data filtering query has been constructed using exactly the specified context.

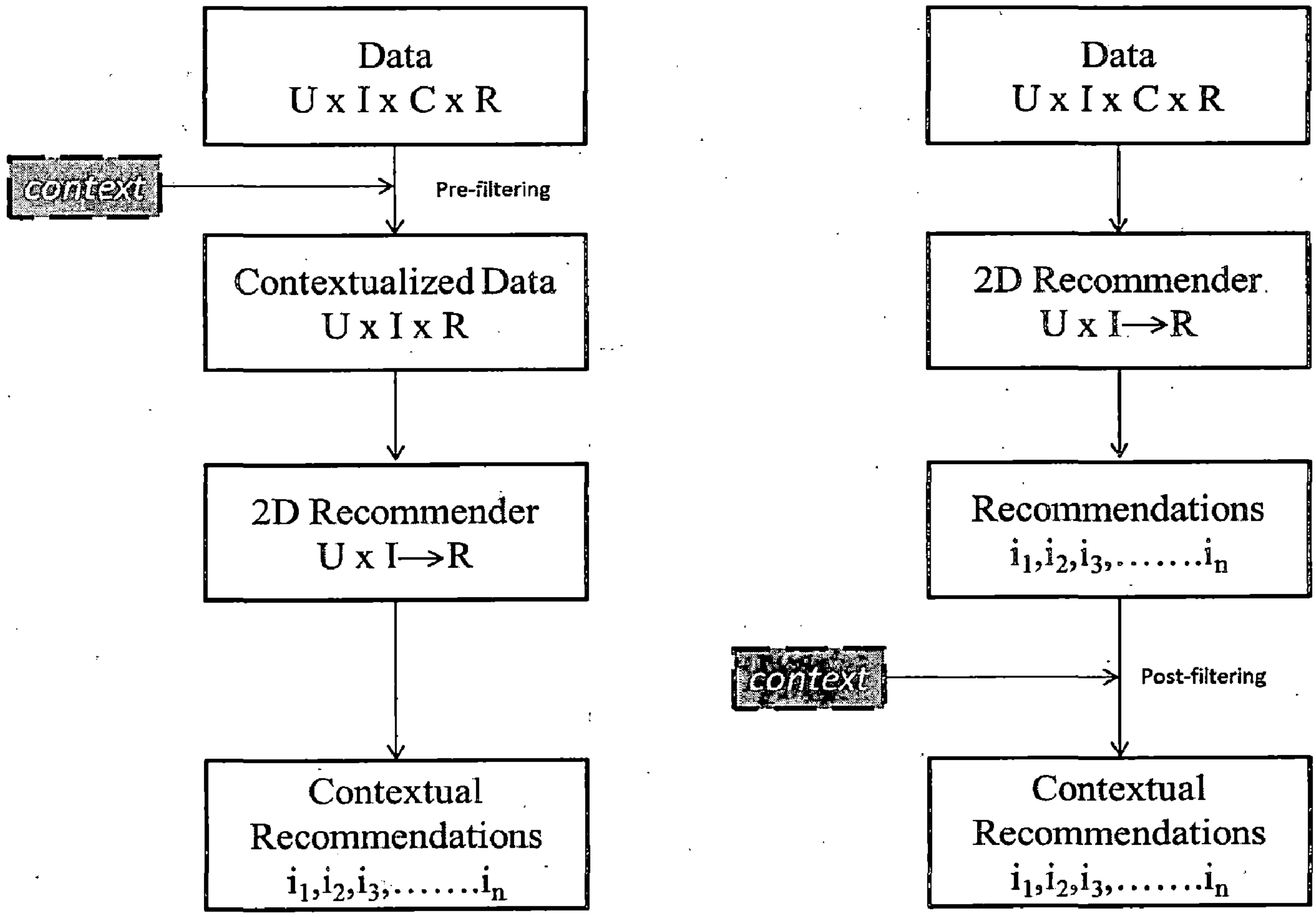


Figure 2.3 – Paradigms for incorporating context [5]

(a) Contextual pre-filtering (b) Contextual post-filtering

In our case, we will be using the contextual data filter by narrowing down the set of users to be considered for collaborative filtering to the set of Top N similar users in a person's social context.

2.3.2.2 Contextual Post-Filtering

As shown in Figure 2.3b, the contextual post-filtering approach ignores context information in the input data when generating recommendations, i.e., when generating the ranked list of all candidate items from which any number of top-N recommendations can be made, depending on specific values of N. Then, the contextual post-filtering approach adjusts the obtained recommendation list for each user using contextual information [5]. The recommendation list adjustments can be made by:

• Filtering out recommendations that are irrelevant (in a given context), or

• Adjusting the ranking of recommendations on the list (based on a given context).

For example[5] , in a movie recommendation application, if a person wants to see a movie on a weekend, and on weekends she only watches comedies, the system can filter out all non-comedies from the recommended movie list. More generally, the basic idea for contextual post-filtering approaches is to analyze the contextual preference data for a given user in a given context to find specific item usage patterns (e.g., user Abhinav watches only comedies on weekends) and then use these patterns to adjust the item list, resulting in more "contextual" recommendations.

2.3.3 Prior Work

Prior work on Context Aware Recommendation systems is relatively scarce and this field remains open to a lot of improvements. In this section, we will discuss in brief some research that addresses this issue.

1) **COMPASS [31]** – It is a context-aware mobile tourist application that adapts its services to the user's needs based on the user's current context. In order to provide context-aware recommendations, a recommender system is integrated with a context-aware application platform. For example, a tourist expressing an interest in history and architecture is served with information about nearby monuments built before 1890. A tourist expressing

the wish to find a place for the night gets a list of hotels and campsites in and around town that match his preferences for accommodations. For both the above cases, the context taken into account is his current location, and his interests are collected explicitly as inputs for the system.
Contextual Post-Filtering: Based on Location Context + Explicit user input.

2) **TCCF [28]** – Reyn et al. introduce the concept of Tag Base Contextual Collaborative Filtering whereby they take into commonly tagged items by different users and the context in which they tagged those items. They give weighted scores to these items where greater weight is given to a common tag with common context than just a simple common tag between two users. They introduce their own method to calculate similarity scores essentially based on the idea of cosine similarity with slight modifications.
Contextual Pre-Filtering: Based on social tagging systems.

3) **Context-Aware Media Recommendations [33]** – This research addresses the issue of recommendation systems that can handle all three context categories—user preference, situation context, and capability context. However, the work is primarily focused on improving the post filtering; hence leaving most of the research gaps of recommender systems such as scalability unaddressed. It also proposes context ontology for post-filtering. Also, the work on user preference is not very convincing as there is no method proposed as to how we will actually obtain user preferences. The work only focuses on once we have the preferences, how we can use them to modify similarity scores and thereby contextualize the recommendations.
Contextual Post-Filtering: User Preference, Situation Context.

As can be noticed from the work above, and as was seen in other work in this field, most of the systems talk about doing either one of contextual pre-filtering or post-filtering and tend to use the same context information for both processes even if they talk about both. An interesting idea that might be explored is that it is possible the system yields better results if use different types of contextual information for these two processes.

2.4 Research Gaps

We discussed two categories of adaptive systems in sections 2.3 and section 2.4 i.e. Context – Aware Systems and Recommender systems. Each of these fields have their own research gaps. Listed following are the gaps that will be addressed in this work:

1. As Table 2.1 shows, most of the systems have not exploited a user's social context. Nowadays, most of the user's social life is available via social networks, we can extract this information and utilize it. Performing social network collaborative filtering is one of the fastest emerging trends in this field.
2. The problem of taking user's preferences and long-term properties has not been taken into account effectively by any of the context-aware systems. Context needs to be categorized in a temporal way based on longer term context which varies relatively less frequently as compared to shorter term context which is mostly obtained from sensor data.
3. One of the major gaps in recommender systems is the issue of scalability. Millions of people use e-commerce websites these days and running O(mn) algorithms on such huge databases create scalability issues. These can be addressed by using contextual pre-filtering to narrow the user space.
4. Another major issue coupled with (3) is ensuring the quality of contextual recommendations. Limiting the user space tends to decrease the quality of recommendations. There is a need for a novel algorithm to ensure the quality of recommendations does not suffer in such cases.
5. A user's feedback is very important to the system. Context Aware Systems as well as Recommender Systems tend to go overboard in delivering personalized service, therefore a user feedback is required to successfully evaluate such systems.

CHAPTER 3
PROPOSED SOLUTION

In this chapter, we first discuss the context hierarchy used for the system, then we discuss how we assume the context aware recommendation system architecture to be. In the third part, we propose a novel contextual pre-filtering algorithm to address the scalability-quality of recommendation tradeoff.

3.1 Context Definition and Hierarchy

We categorize into three major categories i.e. User Context, Situation Context and Resource Context. We define these categories and illustrate the scenarios where each comes into picture.

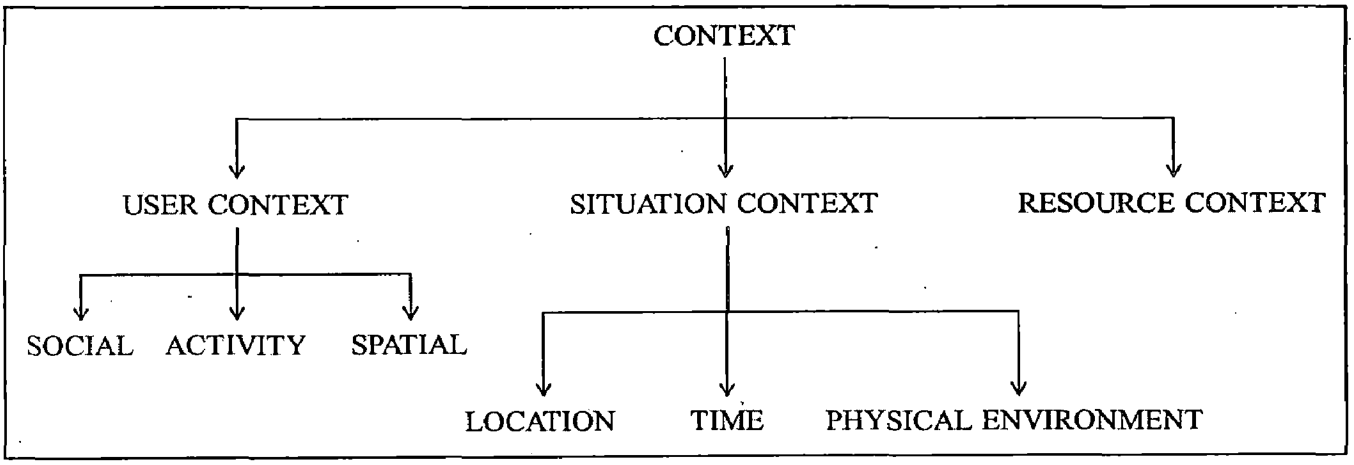


Figure 3.1 – Context Hierarchy

1) **User Context** - This is essentially the context information that varies from individual to individual. This figures out as the most important part of context awareness, as the idea of context awareness is primarily to provide personalized tailoring of services. Following are some important subtypes of the same
    a. Social Context - It implies the personal likes and dislikes of a person, information he shares with people, the way he behaves in a group, who his friends are etc. With the advent of social networking, it has become really feasible to obtain the interests, activities and preferences of a user. Users are now willingly sharing their interests, activities and preferences, which enable systems to ingest, mine and use this social context. Social Context is used extensively by several websites

and service providers to provide context-aware services to a user such as friend suggestions, advertisements etc. Walmart Labs is working on the Social Genome [34] project which aims to make product purchase recommendations based on person's social context.

b. Activity Context - The activity performed by a certain individual such as running, dancing, walking etc. or even browsing, listening to music etc. is a very rich source of behavioral information that defines the characteristics of a person. Activity context is a high-level rich context that is inferred using raw sensor data from various sensors such as accelerometer, gyro meter etc. An example use case for activity context can be an Express News Reader that customizes the length of the news article to be displayed based on whether the person is driving (short), walking(medium) or at rest(full). CenceMe[35] is one of the systems to makes use of inferred activity information to provide context aware results.

c. Spatial Context - Holds information regarding proximity to a user and describes the relevance of a certain location to a user (e.g. Home , Office etc.), and not just the coordinates of a place. This is how it is essentially different from just Location Context which would be the same for all users sharing the same location. Here, the same location might be 'home' for one user and 'office' for the other. An Automated Meeting Planner can use a person's current location and determine which of his friends are in his proximity and can suggest a relevant meeting place close to each of their homes/offices.

2) **Situation Context** - This part of the context hierarchy varies from one situation to the other and is the same for all individuals sharing the same situation unlike User Context. The major aspects of Situation Context are:

   a. Location - Location based services have been in use for a long time now. Location Context is very useful and has been used in many tourist guide application such as GUIDE [10].

   b. Time - Time, day, month, year etc. can be used to customize things very easily. For example, a context aware clock can change background themes depending on whether its day or night. A calendar can be customized to show varying wallpapers based on the season in which the month lies.

There are several others situational parameters that can be exploited productively. E.g. Sound, Lighting, Temperature etc. Gellersen [36] introduced the term situational context and has worked on inferring higher level information based on above mentioned parameters.

3) **Resource Context** - This encompasses the quality of contextual information obtained from the device or the resource used by the user. There are many resource constraints that vary from person to person based on the devices used by them. Some of these are network connectivity, storage space, computational power, accuracy of the sensor data etc. Based on the resource context, applications determine the level of personalization to be delivered. Also, a user might wish to withhold certain information for privacy, which is then accommodated by the resource context.

3.1.1 Notion of Short-term and Long-Term Context

After discussing the categories of context in detail, we can make a common observation. The above categories cover the context information that can be obtained from sensor data. In other words, it is short-term information that is likely to change each time sensor readings for users vary. If we want to holistically model a user, we also need to take into account the properties that are not entirely dependent on his current scenario i.e. long-term properties that reflect about the person's nature, behavior etc. To address this, we now discuss the concepts of short-term and long-term context.

1) Short Term Context - Short Term context refers to any information that affects the interaction between a user and an application based on user's current situation. A user's short term context changes relatively quickly and is mostly obtained from sensors deployed in an active space.

   Say a user U in a scenario S. Now the entities of S that characterize the behavior of U with an application A constitutes his short-term context. Let us discuss a use case and we shall see how both types of contexts will prove useful separately. Say an application A delivers Personalized News services to a user. By detecting a user U"s current city (a feature of his current situation S and hence his short-term context), we can remark that he would be interested in the news of his current city i.e. local news. A classic example of this is any Weather website.

2) Long Term Context - Long Term context characterizes a user's longer term properties that are unlikely to change quickly irrespective of his current situation. Long term context helps in moderating the relevance of short term context and can be used with higher dependability if the sources of short-term context cannot be relied upon.

Now let us take the same case as above. For a user U, Personalized News Service A delivers customized results based on his current city. But in case U is currently visiting a city for just one or two days and resides somewhere else, he would be more interested in the news of his hometown than this local city. Here is where the Long-term context comes into picture. By seeing the location trends of a user over a period of time, the city where he most frequently.

3.2 Scheme for providing Context-Aware Recommendations

For providing context-aware recommendations, this work proposes that the schemes of contextual pre-filtering and contextual post-filtering need to be isolated from each other and looked as processes addressing separate gaps.

Here is what we shall be addressing using the contextual pre-filtering process:

1) **Context Exploitation** - As we have shown earlier, context aware systems till now have not been able to really exploit social context. We will be exploiting social context using social network collaborative filtering. Also, the notion of long-term and short-term context will be seen to significantly improve the scheme of providing recommendations
2) **Scalability** - The problem of scalability in recommender systems will be addressed using social context. The probability of a user being similar to another random user is significantly lesser than his probability of being similar to his friends and people he knows. This is assumed from the fact a person's social circle not only influences his interests and preferences but a person is also insatiably inclined towards trying new things that his friends are trying and he would not have tried otherwise. As [5] says, aim of contextual pre-filtering is the need to reduce test space. Therefore, we will not only be addressing the issue of scalability by reducing the user space from the whole universe of users to a set of friends but also be making the process inherently context aware. However, this leaves the issue of scalability vs. quality of recommendations unaddressed.
3) **Quality of Recommendations** – As shown in [25], reducing the user space is bound to decrease the quality of recommendations. Therefore, to address this we propose a novel

algorithm and then a further refinement of that algorithm, that we will show gives better results than existing algorithms. To do this, we shall evaluate the set of similar user, rather friends obtained in each case.

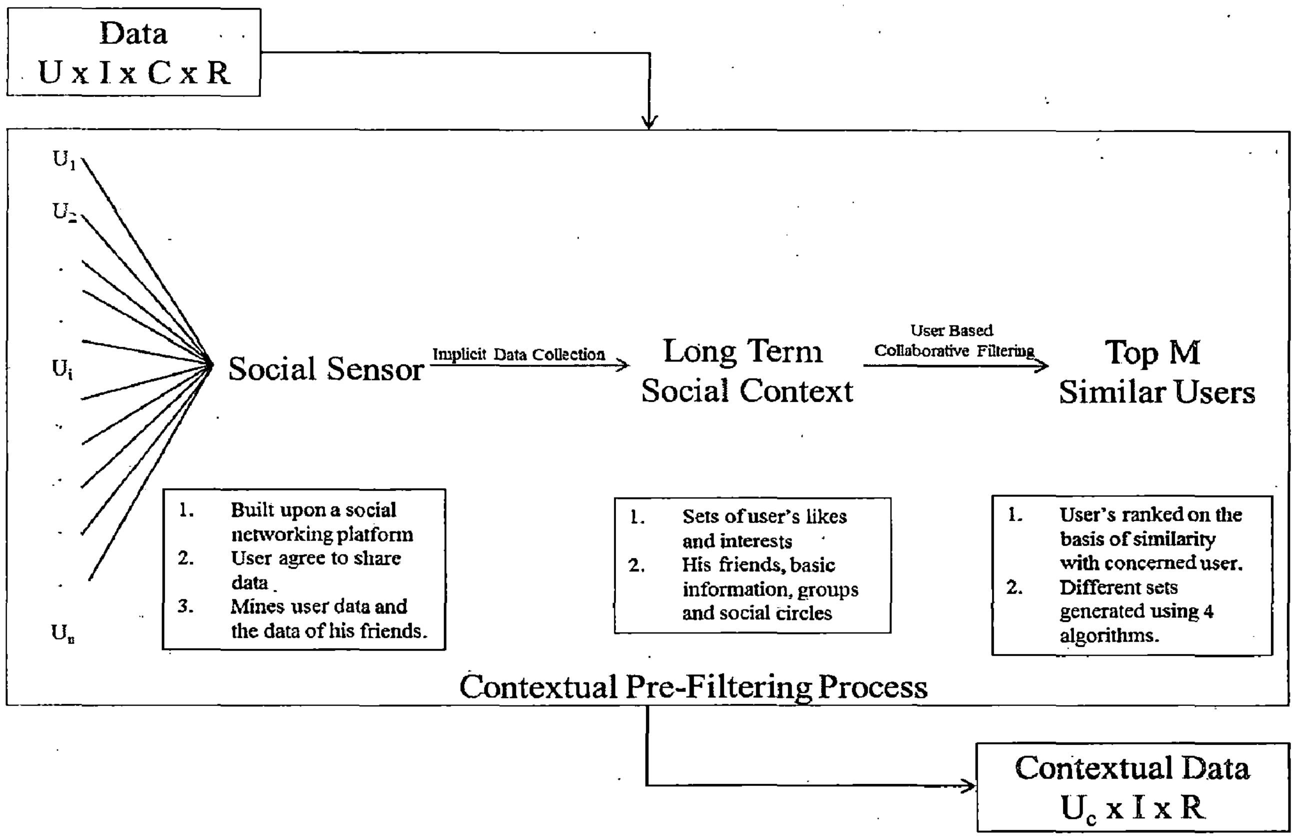


Figure 3.2 – Workflow of proposed Contextual Pre-filtering

As Figure 3.2 depicts, the contextual pre-filtering process narrows the user space by filtering out the set of most similar users to the concerned user based on his long-term social context. The most important question to be addressed is:

1) Why Long Term Context?

   Long-Term Context (essentially user likes, interests and friends) has been used for the pre-filtering process because pre-filtering is essentially a pre-computation step to the recommendation process. As discussed earlier, a user's long-term context varies rather less frequently therefore saving us computational constraints of calculating the 'most similar set' each time we provide the user with recommendations.

2) Why Social Context?

   Again, as discussed earlier, social context has been not really exploited by context-aware systems. With users increasingly sharing data and making information public on social networks, we can exploit this and make the personalization process better. More so, it save us the hassle of doing explicit data collection as in most of the recommender systems where users have to start rating things before recommendations can be provided. Here, we are implicitly collecting the information the user has already provided on his social networks.

Moving on to the next part, as Figure 2.3 (a) shows, the step that follows is that of 2D-recommendations. In [24], Sarwar et al. show how Item-Based Collaborative Filtering processes outperform User-Based process when it comes to recommending items to a user. Therefore, Item-based CF can now be used to filter set of Top N items from the set of Top M users. As for the Contextual-Post Filtering part, we have proposed a context hierarchy in section 3.1 which can be used to filter the results obtained from the above process. Short-Term context is used to filter information here since this information has to reflect the current needs of the user. E.g- If a set of Top N movies for a user has been found out, they can be sorted in the order of how the user will like them in his current mood or location. Figure 3.3 shows the proposed scheme.

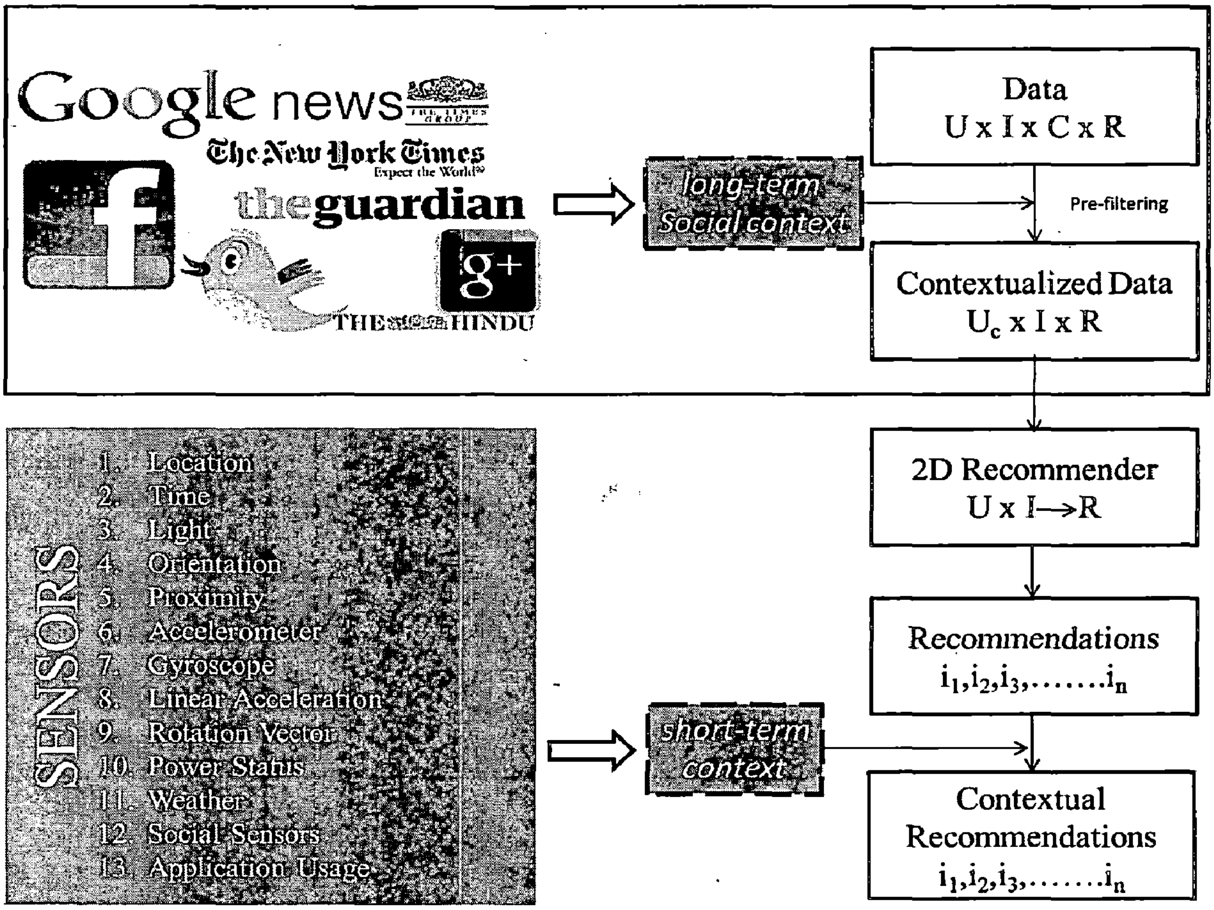


Figure 3.3 – Proposed Context-Aware Recommender Scheme

3.3 Algorithms for Contextual Pre-Filtering

3.3.1 Maximum Mutual Items

Maximum Mutual Items – This algorithm determines most similar users by counting the number of items that have been liked or purchased by both users in question. The user who has purchased most items in common with another user is most similar to that user. This algorithm is used by Discover Facebook Pages [37], which displays a set of 'Friends similar to you' as shown in Figure 3.4 based on most mutual page likes.

Figure 3.4 – Screenshot from 'Discover Facebook Pages' using Maximum Mutual Likes as the similarity criteria.

<u>**Algorithm 1**</u> – Determine most similar users to a user U using most mutual likes.

Input: U's likes, U's friend list, Friends' likes **Output:** Top N similar users

```
For each page i in U's likes
    val <- i.pageid;
    hash[val] <- U.userid;
For each friend F in U's friends
    For each page i in F's likes
        val <- i.pageid;
        if (isset(hash[val]))
            F.mutuallike <- F.mutuallike+1;
sort(F,'mutuallike');
```

Algorithm 1 uses multisort function from util class of PHP.

3.3.2 Cosine Similarity

As is quite apparent, the flaw in the above algorithm lies in the fact that it does not consider the total no. of likes by a user before giving a similarity value i.e. if two users like 2 mutual items and like 2 items all in each, they have a 100% commonality which should be more reflective of similarity as compared to two users who like 100 items each and share 10 items from them.

So, we apply the basic user based collaborative filtering as the first refinement.

We have used cosine similarity as the similarity measure because the user rating matrix constructed here is a binary matrix i.e. either a user likes a page or doesn't like a page, there is no concept of an average rating.

We will be using Equation (2.2) from Section 2.2.2 for Algorithm 2.

Algorithm 2 – Determine most similar users to a user U using cosine similarity.

Input: U's likes, U's friend list, Friends' likes **Output:** Top N similar users

```
For each page i in U's likes
    val <- i.pageid;
    hash[val] <- U.userid;
    U.likecount <- U.likecount+1;
For each friend F in U's friends
    For each page i in F's likes
        val <- i.pageid;
        F.likecount <- F.likecount+1;
        if (isset(hash[val]))
            F.mutuallike <- F.mutuallike+1;
    F.similarity <- F.mutuallike/(sqrt(U.likecount)*sqrt(F.likecount));
sort(F,'similarity');
```

3.3.3 Proposed Page Popularity Rating Algorithm

Now Algorithm 2 uses basic user-based Collaborative Filtering techniques. This section proposes our first modification to the algorithm of social context-based prefiltering. Here, we propose a hypothesis that :

Each commonly bought item or common interest is not an equal reflection of similarity. More popular the item or more commonly purchased the item, lesser is the weight with which it reflects the similarity between two users.

As an example of the above, let us assume in the first case, two users purchased a shaving cream, which is likely to be purchased by all adult men. In the second case, two users purchased an Apple iPhone, which is likely to be purchased by only people who like Apple products, can afford the product, and require a mobile phone. As another example, say two users like a very popular personality, say Sachin Tendulkar, and two users like a relatively less popular personality, say Morne Morkel, our assumption says that in both above listed case the latter pair is more similar as compared to the former. In either case, it is a useful hypothesis and the results can either prove it or disprove it. As we will see in the results section, performance improves after taking this hypothesis into account.

To implement Page Rating, we took into account the number of likes on a page (no. of purchases in case of a product) as a measure of its popularity and gave the pages a rating on scale 1-5. The most popular were given rating 1 and the least popular pages were given rating 5. Also, we will now define the aggregate similarity as a mean of similarities taken over these dimensions.

Algorithm 3 – Determine most similar users to a user U using cosine similarity with Page Rating.

Input: U's likes, U's friend list, Friends' likes **Output:** Top N similar users

```
For each page i in U's likes
    val <- i.pageid;
    rating <- i.pagerating;
    hash[val] <- U.userid;
    U.mod <- U.mod + rating*rating;
For each friend F in U's friends
    For each page i in F's likes
        val <-  i.pageid;
        rating <- i.pagerating;
        F.mod <- F.mod+rating*rating;
        if (isset(hash[val]))
            F.mutualmod <- F.mutualmod+rating*rating;
    F.similarity <- F.mutuallike/(sqrt(U.mod)*sqrt(F.mod));
sort(F,'similarity');
```

3.3.4 Proposed Multidimensional Approach Algorithm

Now Algorithm 3 seems a logical refinement of Algorithm 2 but to make our system more effective, we make the following proposition:

Similarity cannot be viewed in a linear dimension. People's interests need to be categorized into different dimension and dimensional similarity should be accounted for.

The above lines essentially imply that people's interests can be classified into discrete well-defined categories such as movies, music, books, games etc. and each of these categories have an individual importance. e.g. – if two users U_1 and U_2 like 100 mutual pages and all those pages are of musicians, recommending U_1 a movie based on his similarity with U_2 seems far-fetched. Therefore, we categorize the pages into four broad categories – Movies, Music, Books and Other and define dimensional similarity as the Cosine Similarity with Page Rating in each dimension.

Algorithm 4 – Determine most similar users to a user U using dimensionality.

Input: U's likes, U's friend list, Friends' likes **Output:** Top N similar users

```
For each page i in U's likes
        val <-  i.pageid;
        rating <- i.pagerating;
        hash[val] <- U.userid;
        dim <- i.category
        U.dim.mod <- U.dim.mod + rating*rating;
For each friend F in U's friends
        For each page i in F's likes
                val <-  i.pageid;
                rating <- i.pagerating;
                dim <- i.category
                F.dim.mod <- F.dim.mod+rating*rating;
                if (isset(hash[val]))
                        F.dim.mutual <- F.dim.mutual+rating*rating;
dimcount <-0;
For each dim in dimensions
        F.dim.similarity <- F.dim.mutual/(sqrt(U.dim.mod)*sqrt(F.dim.mod));
        dimcount<-dimcount+1;
For each dim in dimensions
        F.aggregatesimilarity <- F.aggregatesimilarity + F.dim.similarity;
F.aggregatesimilarity <- F.aggregatesimilarity/dimcount;
sort(F,'aggregatesimilarity');
```

CHAPTER 4
IMPLEMENTATION DETAILS

Most of the deployment has been completed using Facebook as the social network. The algorithm have been tested locally by building a real dataset with Facebook APIs.

4.1 Resources Used

4.1.1 Facebook Platform Javascript SDK and PHP SDK

The PHP SDK[38] provides a rich set of server-side functionality for accessing Facebook's server-side API calls. These include all of the features of the Graph API[40], FQL, and the Deprecated REST API.

The PHP SDK is typically used to perform operations as an app administrator, but can also be used to perform operations on behalf of the current session user. By removing the need to manage access tokens manually, the PHP SDK greatly simplifies the process of authentication and authorizing users for your app.

The JavaScript SDK provides a rich set of client-side functionality for accessing Facebook's server-side API calls. These include all of the features of the REST API, Graph API, and Dialogs. Further, it provides a mechanism for rendering of the XFBML versions of our Social Plugins, and a way for Canvas pages to communicate with Facebook.

4.1.2 Facebook Graph API

At Facebook's core is the social graph; people and the connections they have to everything they care about. The Graph API presents a simple, consistent view of the Facebook social graph, uniformly representing objects in the graph (e.g., people, photos, events, and pages) and the connections between them (e.g., friend relationships, shared content, and photo tags).

Every object in the social graph has a unique ID. You can access the properties of an object by requesting https://graph.facebook.com/ID. For example, the official page for the Facebook Platform has id 19292868552, so you can fetch the object at https://graph.facebook.com/19292868552. Figure 4.1 shows the format of the fetch object:

```
{
  "name": "Facebook Platform",
  "website": "http://developers.facebook.com",
  "username": "platform",
  "founded": "May 2007",
  "company_overview": "Facebook Platform enables anyone to build...",
  "mission": "To make the web more open and social.",
  "products": "Facebook Application Programming Interface (API)...",
  "likes": 449921,
  "id": 19292868552,
  "category": "Technology"
}
```


Figure 4.1 – Fetch object using Graph API

4.1.3 Apache Server and MySQL

Wamp server running on Apache 2.2.21 was used and dataset was deployed using PHPMyAdmin. To run the algorithms on localhost, PHP and MySQL were used as the essential languages.

4.2 Application Module: Check Mate

As a prerequisite for this work, we needed a real working dataset of a considerable amount of users to test the algorithms and evaluate them. Therefore, an application module called Checkmate was built using Javascript and PHP and deployed on Facebook as an application using Heroku as the free cloud provider. Our application gives a set of most similar users based on interests users share with their friends. Currently, most applications like this treat most mutual friends or sharing common groups as an indicator but as Figure 4.2 shows, there is no concrete relation between the two.

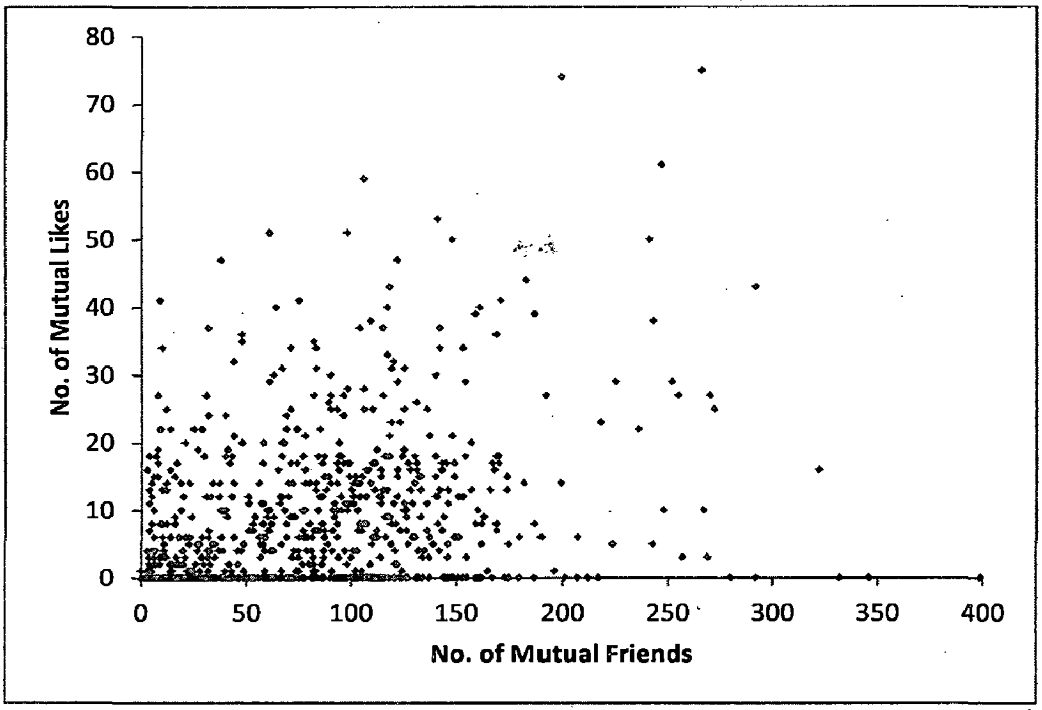


Figure 4.2 – Variation of no. of mutual likes with no. of mutual friends

The application Check Mate shown in Figure 4.3 serves a two-fold purpose:

1) **Interesting to Use** - The users who login to this application are shown a list of friends they are most similar with. It is of interest to most users as they might discover they share interests with people they might be friends with but not communicating frequently with. They discover who is the person to look forward to when they want to watch a movie, whom to go to for borrowing a good book for reading, whom they share most groups and friends with. Not many such applications exist and even if they do, they focus on getting people to know people they don't already know by using most mutual friends or common locations as criteria. Our application is purely interest-based.
2) **Data Gathering** – Users who login to this app essentially make available to us some data that is used by us for our work. This include the list of people they are friends with, what pages they have liked on Facebook and what their friends like on Facebook. Facebook is being used by over 700 million people today and therefore, its inarguably a great platform to conduct a social network analysis experiment. We will discuss in following section the need and structure of the dataset built.

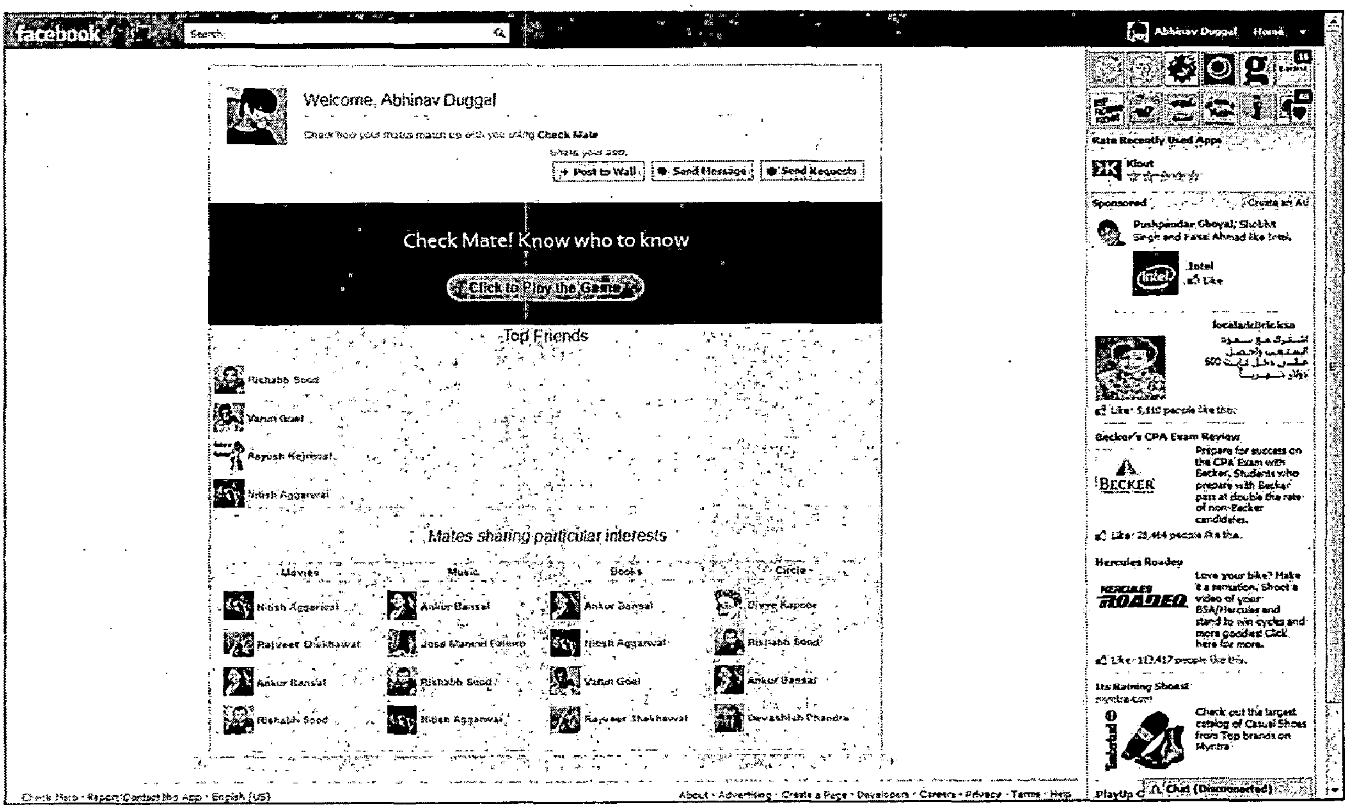


Figure 4.3 – Screenshot of Application Module : Check Mate deployed on Facebook

Figure 4.3 shows a screenshot of deployed application running on Facebook which shows results for Abhinav Duggal. It performs User based Collaborative Filtering to determine Top-4 set of users with cosine based similarity as the similarity metric. Since there were resource restrictions on using free cloud services, the application deployed on Facebook takes only 20 friends into consideration at once, the application running on localhost runs for the whole set of friends.

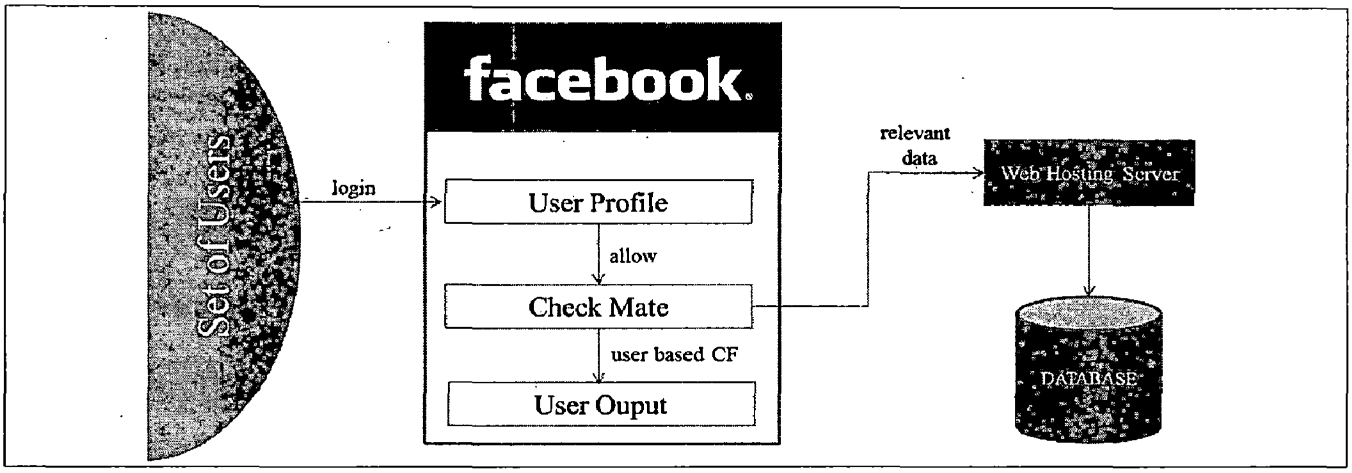


Figure 4.4 – Information Flow in Check Mate

4.3 Data Gathering Module

4.3.1 Implicit Data Collection

One of the best aspects of doing pre-filtering using social network is that we do not need the user to explicitly rate a set of items before we can start giving out recommendations. We can use what he has already made public. Following is the set of items shared by a user when he logs in with Check Mate:

1) Basic Information (ID, Name etc.).
2) Likes.
3) Friends.
4) Friends' likes.

Besides users, the other set of objects of relevance to us are the pages being liked. Following information about pages is extracted by Check Mate:

1) Page ID.
2) Page Name.
3) Category.
4) Like Count.
5) Talking About Count.

Using the data gathering module, we were able to gather data for about ***600 users, 67123 likes and 26580 distinct pages.***

4.3.2 Building a Real Dataset using Facebook

User
uid : *int* name : *string* date_updated : date

Friend
uid : *int* fid : *int* mutual_friends : *int*

Like
pid : *bigint* uid : *int*

Similarity
uid : *int* fid : *int* set0 : *float* set1 : *float* set2 : *float* set4 : *float*

Page
pid : *bigint* name : *string* category : *string* likes : *int* talking_about : *int* rating : *int*

Figure 4.5 – Dataset Structure

Figure 4.5 represents the tables used in the dataset. This data was obtained using FQL and FBML. The above representation is a logical representation of the data. Redundancy was introduced to optimize the performance. Below is a code fragment showing how data is fetched.

```
$user_id = $facebook->getUser();
if ($user_id) {
$ friends=idx($facebook->api('/me/friends'),'data',array());
for($i=0;$i<count($friends);$i++)
{
     $fname=$friends[$i]['name'];
     $fid=$friends[$i]['id'];
     $tmp=idx($facebook->api
               ('/me/mutualfriends/'.$fid),'data',array());
     $mfcount=count($temp);
     insertmf($myid,$myname,$fid,$fname,$mfcount);
}
}
function insertmf($a,$b,$c,$d,$e)
{
     $sql="INSERT INTO mutualfriends (uid,uname,fid,fname,mfcount)
     VALUES ('$a','$b','$c','$d','$e')";
     $res=mysql_query($sql);
}
```

Incremental Updation – Since the dataset collected is quite large and it is impossible to mine such data at the application runtime, there needs to be a mechanism to keep the dataset updated. In order to account for the changes in people's interests and likes, we need to account for the likes most recently created. But running such updation queries continuously or downloading the whole datasets can be hazardously time consuming.

Therefore, we will do periodic updates. The period can be chosen based on the tradeoff between accuracy of recommendation-bandwidth consumption. In order to do this periodic updates, the user table stores the time and date when the likes of that particular user were last updated. Correspondingly, FQL allows us to determine when a page like was created by a user. Therefore, all the likes that created after the date stored in the user table can be mined and inserted in the Like table and the corresponding pages can be inserted in the Page table if not already there. Figure 4.6 depicts the workflow of incremental updation, with dashed lines being database queries. For our purpose, we run this cycle after a period of seven days.

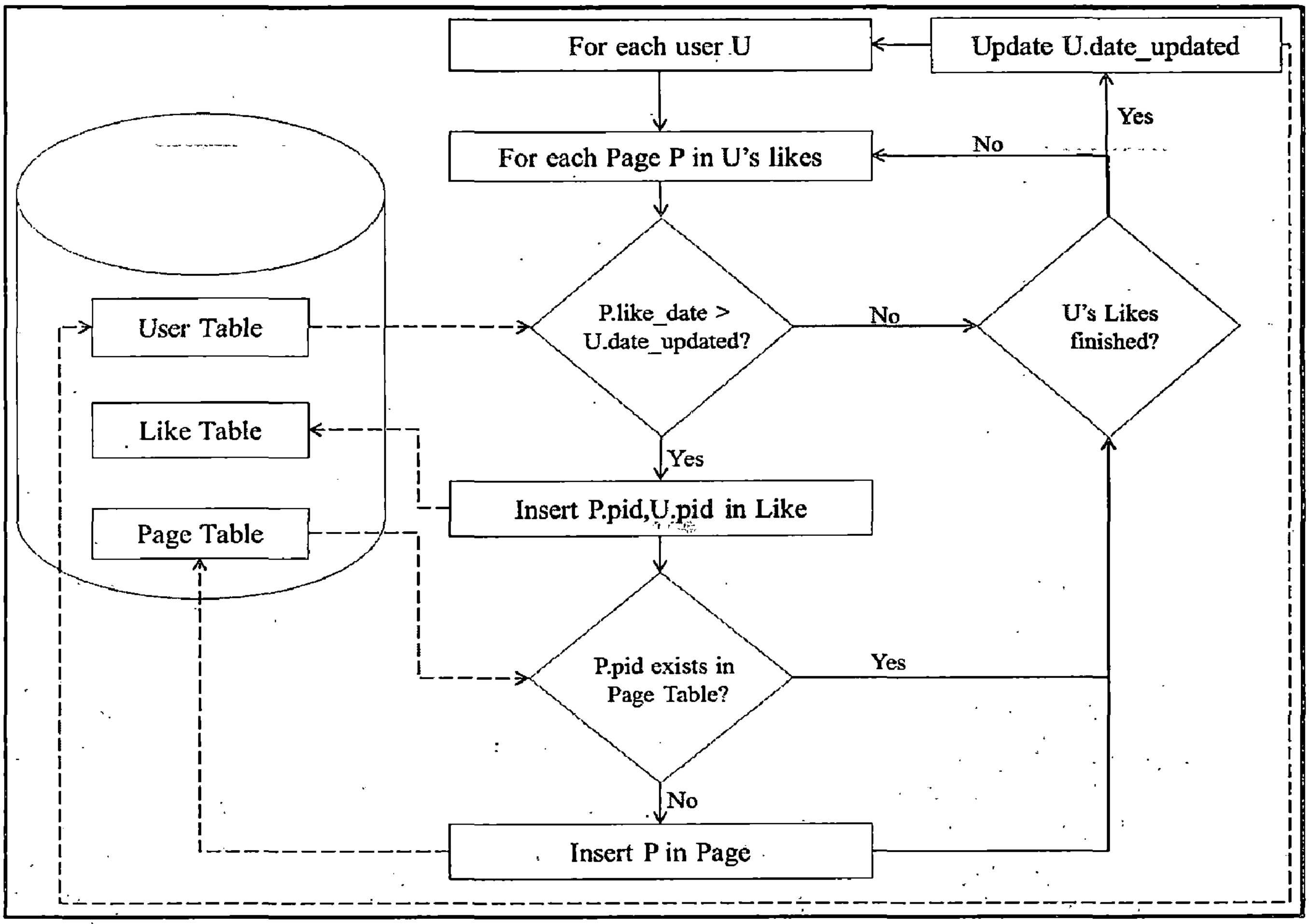


Figure 4.6 – Workflow of incremental update.

4.3.3 Dataset Statistics

Table 4.1 – Dataset Statistics

Total No. of Users	592
Total Page Likes	67123
No. of Pages	26580
Average Likes/User	113.384

All the pages were categorized into some major categories as shown in Figure 4.6 and finally four categories were chosen – Movies, Music, Books and the rest in Others.

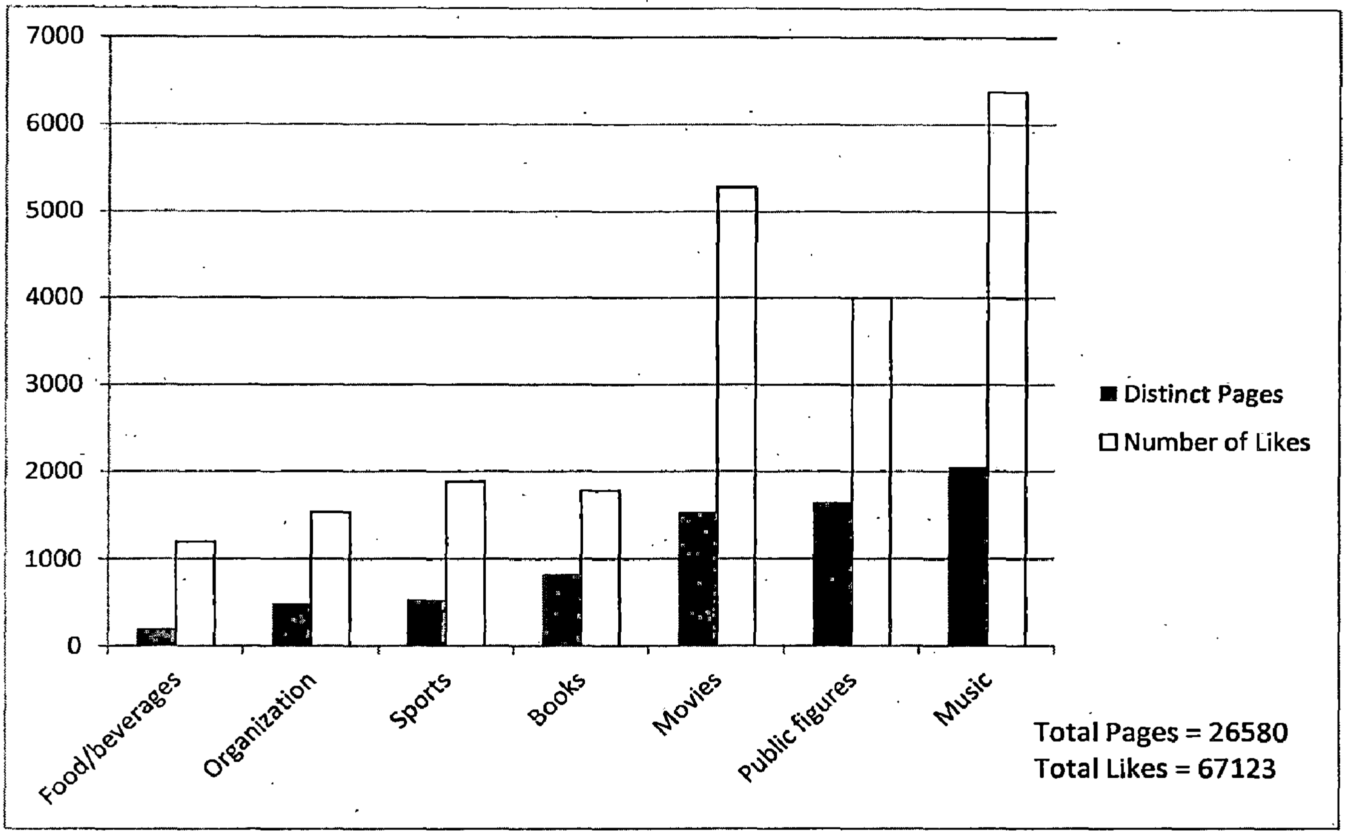


Figure 4.7 – Categorization of Page objects fetch using Check Mate

4.4 Evaluation module

4.4.1 Explicit Data Collection

For evaluation purposes, all the users in the 4 sets obtained via 4 algorithms were required to rate a certain set of items that was intentionally chosen to be a mixed bag of pages from all categories in order to be able to test the aggregate similarity of users.

This was main advantage of using a real dataset. Since all the users were in my social network, I could reach out to them and collect ratings explicitly.

Following is the list of items that were given out to be rated on a scale of 1 to 10.

1. Android
2. Sachin Tendulkar
3. New York
4. The Beatles
5. Google
6. Sholay
7. Jeffrey Archer
8. Dan Brown
9. Goldman Sachs
10. Engineering
11. Barrack Obama
12. Mashable
13. Incredible India
14. Facebook
15. Forrest Gump
16. Harry Potter
17. Anna Hazare
18. Adolf Hitler
19. AR Rehman
20. Adventure Sports

Table 4.2 shows the how the user-rating matrix was constructed. It shows ratings obtained for some 10 users.

Table 4.2 – User-Rating Matrix

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{17}	I_{18}	I_{19}	I_{20}
U_1	9	10	10	8	9	10	6	7	5	5	8	8	9	8	10	10	7	4	6	10
U_2	9	10	10	5	9	10	6	9	5	5	6	8	6	7	10	9	7	5	6	10
U_3	9	5	4	8	4	1	4	7	10	2	8	8	9	8	10	6	7	8	9	8
U_4	9	9	7	8	9	10	6	7	5	6	9	10	8	7	10	9	7	10	6	3
U_5	6	10	4	7	9	10	4	5	10	10	8	8	9	9	10	7	7	8	9	9
U_6	7	10	7	8	9	10	6	7	9	5	8	8	5	7	8	9	9	4	6	10
U_7	10	5	4	6	5	6	4	3	10	10	8	8	9	10	10	3	5	8	9	10
U_8	7	10	9	9	9	7	4	7	8	6	4	10	9	6	10	9	7	8	3	4
U_9	7	10	9	9	9	9	5	8	8	6	8	9	10	6	7	9	7	8	5	8
U_{10}	8	5	4	8	10	6	4	3	8	8	5	8	9	7	10	7	5	8	9	10

4.4.2 Evaluation Paradigms

We do two sets of evaluation on the results obtained using the four algorithms:

1) User Comparison - We compare a user with another user by calculating the mean absolute error in the ratings given by both users.
2) Prediction Comparison - We compare the prediction performance by comparing the error between predicted ratings for a user and actual ratings given by him.

CHAPTER 5
RESULTS AND DISCUSSION

The following sections discuss the performance of the proposed privacy scheme.

5.1 System Configuration

The application Check Mate was deployed on Heroku cloud service provider, to run the algorithms, Wamp server was used on a system with following configurations:

1. Processor : Intel Pentium Dual-Core T2310 1.47 Ghz
2. RAM : 2GB
3. Operating System: Windows-7 Ultimate.

5.2 Evaluation Metrics

The major evaluation metric we use for our system is Mean Absolute Error(MAE).

1) For user set evaluation, as given by equation (5.1) the Mean Absolute Error between two users u and v is defined as:

$$MAE_{u,v} = \frac{\sum_{i=1}^{N}|u_i - v_i|}{N} \qquad (5.1)$$

Here u_i and v_i are the ratings given by user u and user v to item i and N is the set of all items.

2) For prediction evaluation, the predicted rating that user a would give to an item i is given by :

$$P_{a,i} = \overline{r_a} + \frac{\sum_{u\in U}(r_{u,i} - \overline{r_u}).sim_{a,u}}{\sum_{u\in U}|sim_{a,u}|} \qquad (5.2)$$

As discussed before, equation (5.2) takes into account the mean ratings given by different users to take into account rating behaviors. We compare the prediction performance of the system by calculating the Mean Absolute Error between predicted rating for item i by user a and actual rating given by user a to item i. This prediction error is calculated as:

$$PE_a = \frac{\sum_{i=1}^{N}|p_i - a_i|}{N} \qquad (5.3)$$

Here p_i is the predicted rating user a will give to item i and a_i is the actual rating by user.

3) In figures 5.1-5.4, linear trend lines have been plotted over MAE to show the trend in increase/decrease of MAE. (depicted as Linear(Error)).

5.3 Performance Evaluation

5.3.1 User Set Evaluation

We calculate the Mean Absolute Error between the ratings provided by two users and show the error obtained in proposed algorithms is lesser. Also, we will compare the Top-N users of each algorithm between themselves to observe the gradient with which Mean Absolute Error rises. The analysis that is done in Section 5.3.1.1-5.3.1.4 represents how each individual algorithm stacks up to itself i.e. we compare the i^{th} and $i+1^{th}$ users of the same algorithm. As is intuitively clear, the MAE of i^{th} user should be lesser than $i+1^{th}$ user since the latter is less similar. Therefore, a linear trend line plotted on this graph should show a gradual upward rise. As we will see, the rate with which these points jump up and down decreases as we go from Section 5.3.1.1 to 5.3.1.4. On the X-axis, the nearest neighbor rank means the rank that a user got when the similarity measures were sorted in descending order. The Y-axis represents the MAE.

5.3.1.1 Maximum Mutual Likes

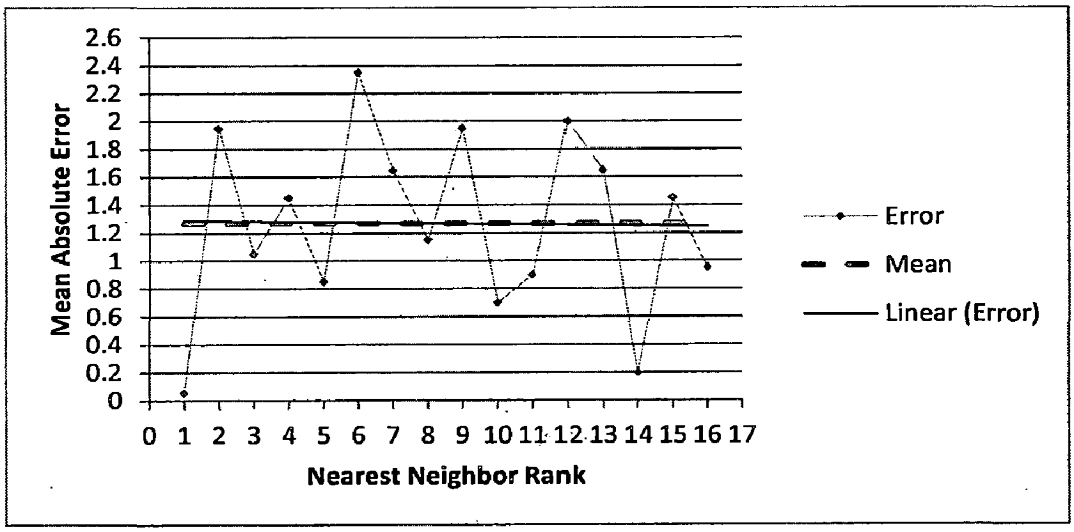


Figure 5.1 – Per User MAE for Maximum Mutual Likes

As Figure 5.1 shows, the mean absolute error for this algorithm varies a lot for the Top-N users. Ideally, the trend line plotted should move gradually upwards as the rank of user (shown on X-axis) decreases i.e. the user goes on becoming less similar. The value of Mean MAE for all users is 1.26875.We now see how the graph for cosine similarity looks like.

5.3.1.2 Cosine Similarity

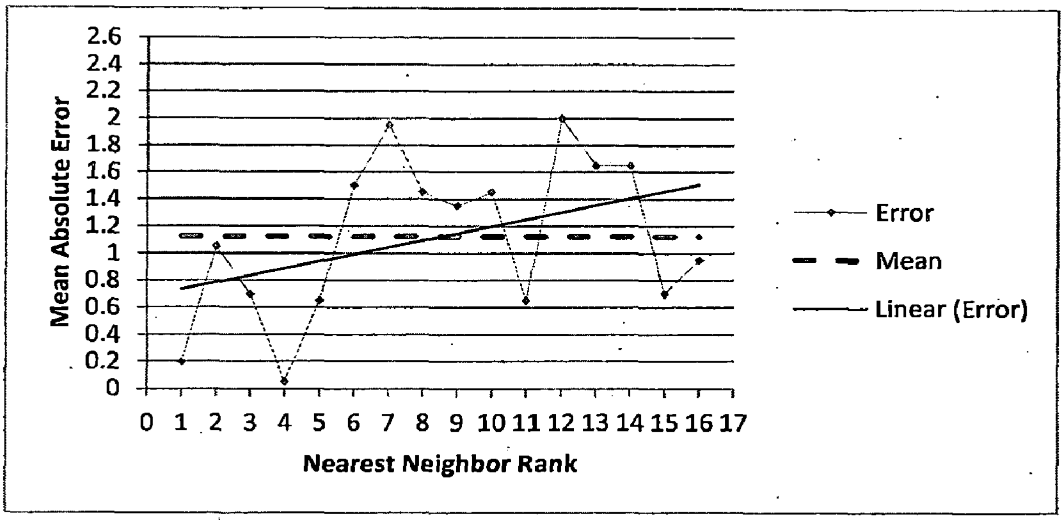


Figure 5.2 – Per User MAE for Cosine Similarity

As Figure 5.2 shows, the mean MAE has decreased, the trend line shows an upward trend. The value of Mean MAE for all users is 1.121875.

We now see how the graph for page rating looks like.

5.3.1.3 Page Rating

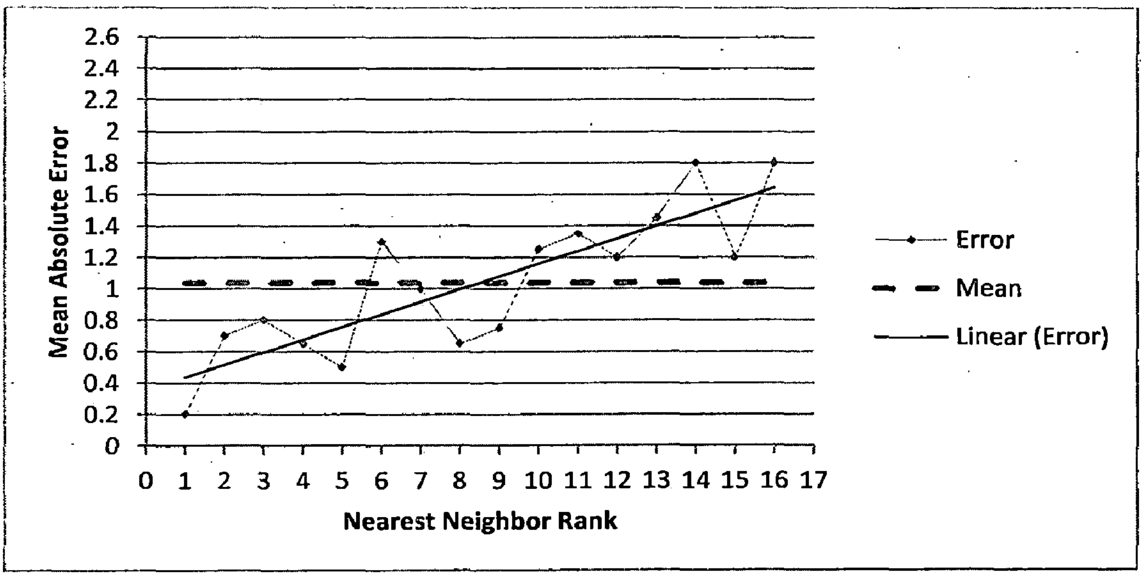


Figure 5.3 – Per User MAE for Page rating

Now we evaluate the first proposed algorithm in which the common pages with high popularity get a low rating and the less popular pages get a higher rating. As Figure 5.3 shows, the mean MAE has again decreased, the trend line shows an upward trend. The value of Mean MAE for all users is 1.0375.

5.3.1.4 Multidimensional Approach

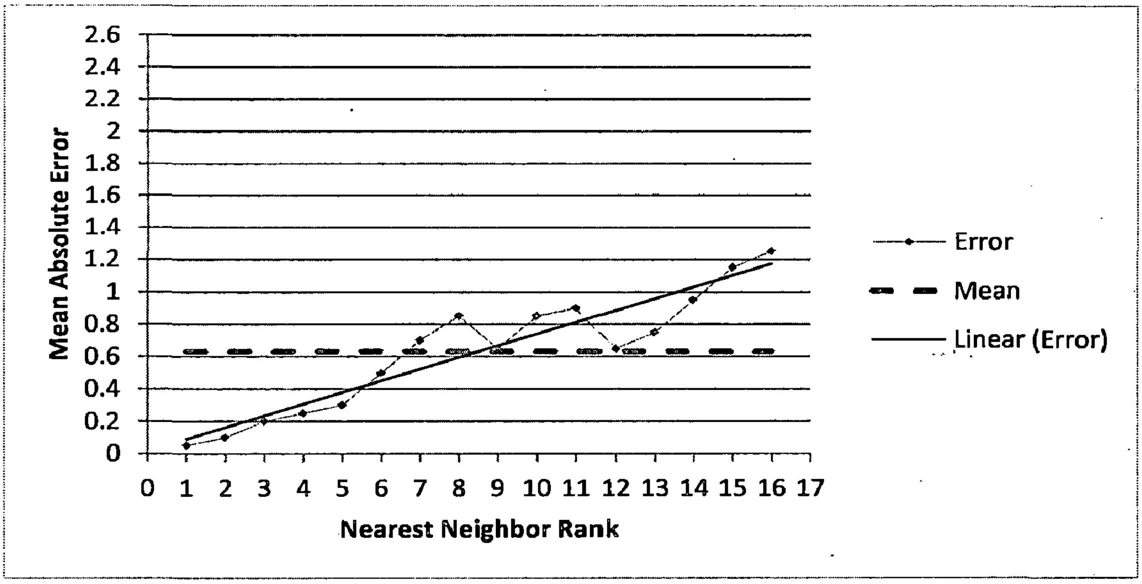


Figure 5.4 – Per User MAE for Multidimensional Approach

This algorithm by far delivers the best results obtained. Here the user error is showing a very regular upward trend as we go from better ranks to worse which means our Top N is very relevant. Also, the mean absolute error has significantly dropped from 1.0375 in Figure 5.3 to 0.63125 in Figure 5.4.

We have now seen how these algorithms perform individually by comparing the error of users getting various ranks in the same algorithm, but in order to prove the proposed algorithm gives better results, we do the following analysis.

5.3.1.5 Average MAE Comparison of all sets

To compare these algorithms we take a moving average of Mean Absolute Error with the size of the set being the number of users encountered i.e. Point with X-Value 10 depicts the Average Mean Absolute Error in ratings when the first ten users' ratings are taken into account.

Here, the X-axis representation has changed. The X-axis here represents the N in Top-N users that were taken to evaluate the user set. The Y-value represents the Mean of MAE taken over all the users. As shown in Figure 5.5, the multidimensional algorithm and page rating algorithm perform better than existing algorithm as the error has been reduced.

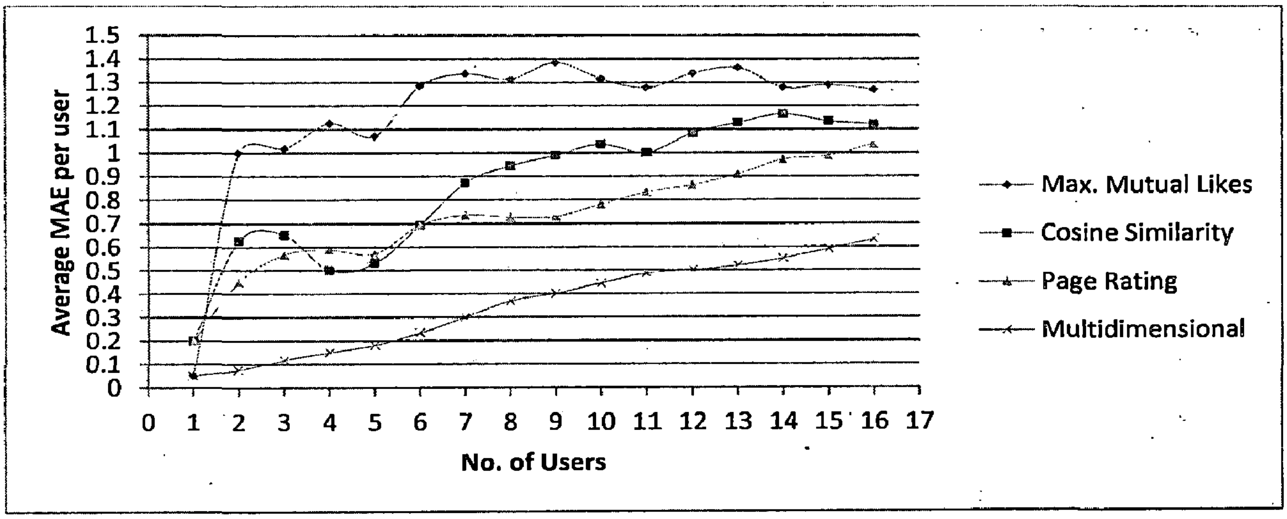


Figure 5.5 – Comparison of Average MAE of Top-N users using 4 algorithms

5.3.2 Prediction Evaluation

In Section 5.3.1, we have compared the set of users obtained from our 4 algorithms and observed the set of users in Page Rating and Multidimensional algorithms gave ratings most close to our user. In this section, we will be comparing the prediction ability of these 4 algorithms. We will be calculating the predicted ratings for a user based on his Top-N users and then compare them with the actual ratings given by the user. The metric is the same i.e. Mean Absolute Error and the following equation (5.2) has been used to predict ratings. To calculate error in predictions the Mean Absolute Error of predicted rating vs. Actual rating is taken. Lower is the error, better is the performance of the algorithm. Figure 5.6 shows how these four algorithms will perform if they are used for generating recommendation with user-based collaborative filtering.

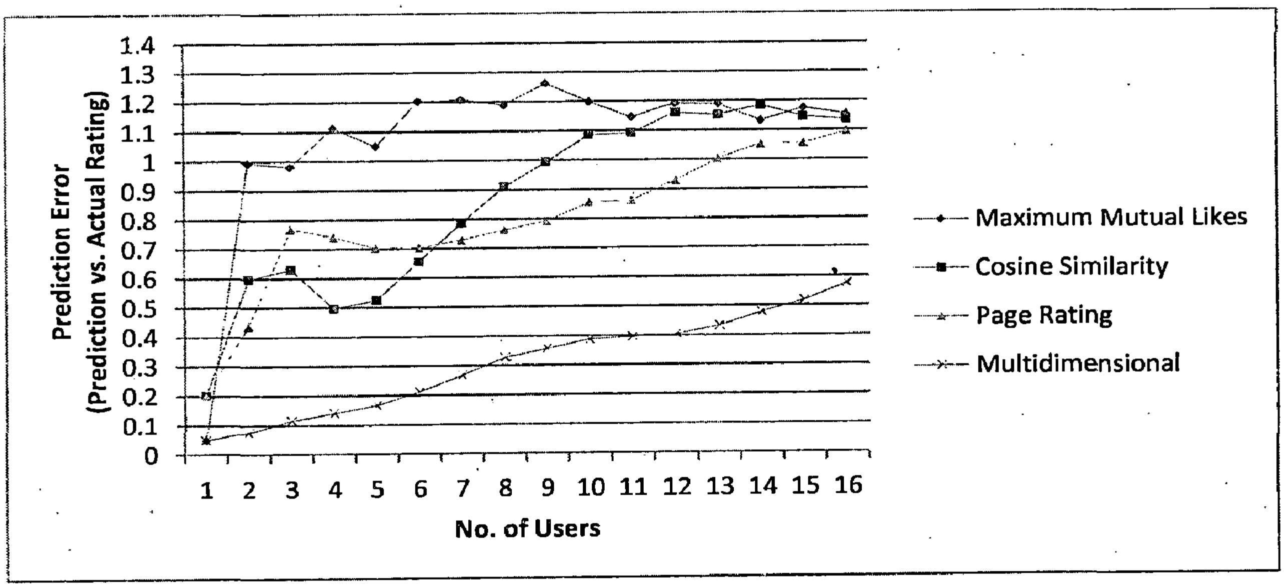


Figure 5.6 – Comparison of prediction performance of 4 algorithms

CHAPTER 6
CONCLUSION AND FUTURE WORK

Social Network Analysis has been pounded upon by many researchers and with the boom of social networking websites as Facebook and Twitter, the data that was otherwise hard to mine explicitly is now available in seconds. Generating recommendations is also the need of the hour as now people look up to recommender systems to virtually tell them what to do. In this pretext, there is a dire need to make this process scalable and efficient.

In our work, we discussed two new algorithms as to how we can use social context to improve the process of pre-filtering thereby making context aware recommendation feasible, scalable and effective. We compared the performance of our technique to other techniques on a small scale and showed how our system was giving much more similar, relevant users than the existing work.

On the other hand, we proposed a hierarchy for context and we proposed the notions of short-term and long-term context. We developed the hierarchy in such a way that modeling short-term and long-term context would be easy. A context aware recommendation system added to a context aware system that takes temporal information into account will be, we say, a much valuable addition. We also saw how user feedback has to play an essential role in enabling context awareness by eventually taking user ratings to evaluate our system.

What we did not address was as to how short-term context will be used to perform contextual post-filtering. We give some suggestions for the same in the following section.

6.1 Suggestions for Future Work

1) Contextual Post-Filtering

With the availability of sensor today and with the advent of mobile computing, a user is essentially connected all the time and trend of performing computations on the fly makes the problem of mining user's activities a very trivial one. The information readily available from low level sensors includes that of location, lighting, temperature etc. and the high level information that can be deduced from this includes activity the user is performing, his current mood etc. There has been a lot of good work on how to obtain and model sensor information.

The major reason why we are discussing this is because sensor information primarily implies the information of user's current situation. And we propose that for post-filtering purposes, we need to use the information of user's current situation i.e. we need to filter out recommendation that are of or negligible use to him currently. e.g. - one very good example of this can be – say a user U goes to new city as a tourist and he wants recommendations for the sites he wants to visit. The pre-filtering + 2D recommendation process will generate recommendations about the places he is most likely to like and visit (note we have taken into account social context already), what can be done now is that these recommended location can be filtered based on his current location i.e. the sight which is closest to his current location will get a higher rating as it will be feasible for him to visit that location first.

The problem here remains is how this short-term context will be modeled. There has been a lot of work on how to use Ontology tools for modeling context, the most commonly used being OWL. Hierarchy suggested in Figure 3.1 can be used and an ontology can be built over it. Figure 6.1 partially shows how a context ontology can be built.

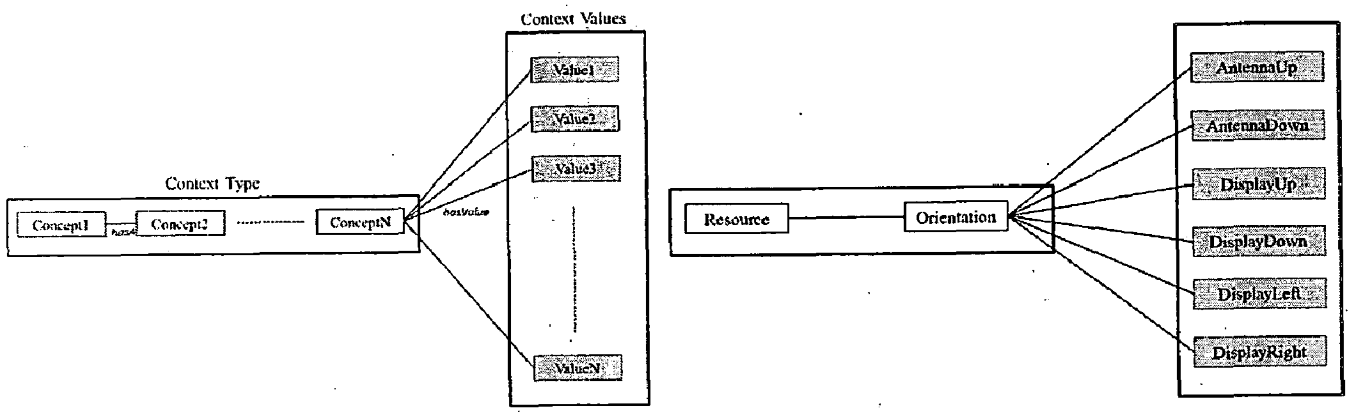


Figure 6.1 – Building Context Ontology (a) A model for creating vocabularies (b) Partial resource Context Ontology Example.

2) Context Awareness Provisioning as a service

Another thing we propose is bit offshore from the issue of context-aware recommendations and is more related to the core of context aware systems themselves. We observed that most of the work that has been done in the field of context aware computing tightly couple the context representation and modeling with the base functionality of the service being provided. We feel that context aware provision needs to be isolated from the base functionality of the service being provided and should be made available as service in itself.

For this purpose, something like an API as shown in Figure 6.2 can be built with standard procedure calls, which is accessible via an interface. The procedure calls take certain parameters about a user and return his context variables, both short-term and long-term.

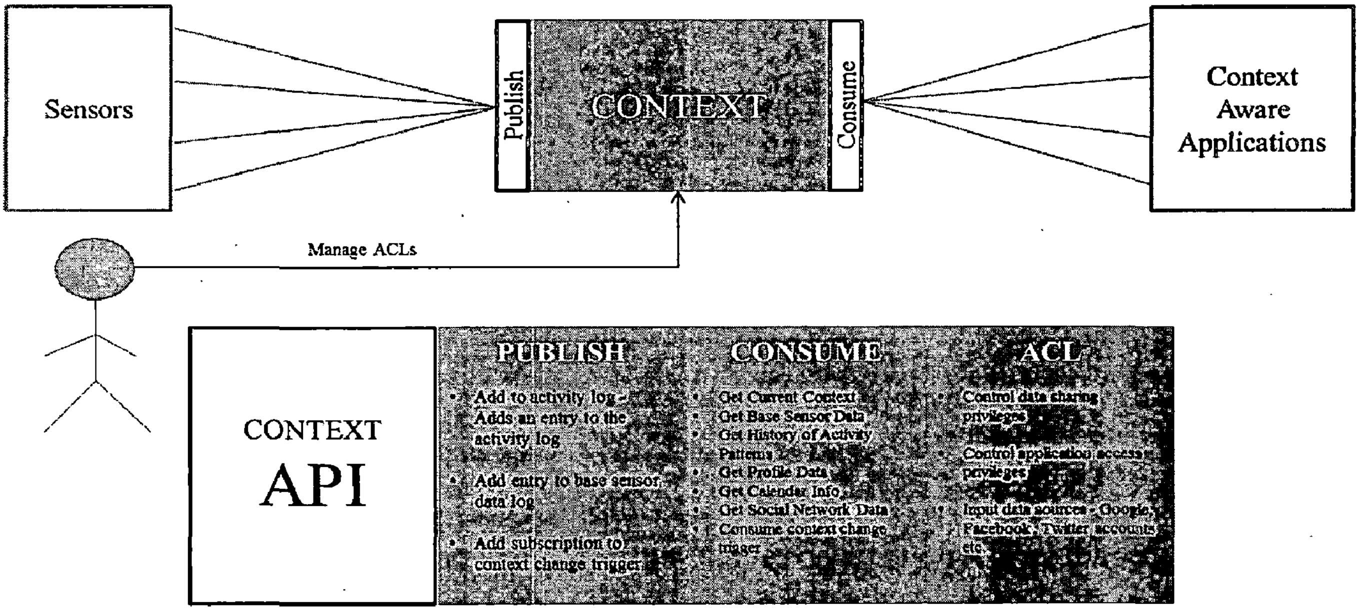


Figure 6.2 – A partial publish/consume Context API

3) Other Suggestions

Some other miscellaneous, nonetheless important suggestions are:

1) Our system has been evaluated on short-scale, it can be evaluated for larger systems to see how much the performance improves.
2) Item-based collaborative filtering can be implemented as the next step on the set of users obtained to generate final recommendations for the user.
3) Activity Analysis can be exploited as another measure of obtaining long-term context for contextual pre-filtering along with social context (as discussed in this work).

REFERENCES

[1]. Bill Schilit, Norman Adams, and Roy Want. "Context-aware computing applications," Proc. IEEE Workshop on Mobile Computing Systems and Applications, IEEE Computer Society Press, Dec. 1994, pp. 85-90.

[2]. Guanling Chen, David Kotz, "A Survey of Context-Aware Mobile Computing Research", Technical Report, Dartmouth College, Hanover, NH, USA,2000

[3]. Dey, A. K., "Providing Architectural Support for Building Context-Aware Applications", PhD. Thesis, Georgia Institute of Technology, 2000.

[4]. A. K. Dey and G.D. Abowd, "The Context Toolkit: Aiding the Development of Context-Aware Applications," Proc. SIGCHI conference on Human factors in computing systems,May 1999, pp. 434-441

[5]. Gediminas Adomaviciu, Alexander Tuzhilin, "Context-aware recommender systems.", Recommender Systems Handbook, Part 1, 2011, pp. 217-253

[6]. Anthony Jameson, "Modelling both the context and the user", Personal and Ubiquitous Computing , Jan. 2001,vol. 5,no. 1 pp. 29–33

[7]. Anusuriya Devaraju, Simon Hoh, Michael Hartley, "A context gathering framework for context-aware mobile solutions" , Proc. 4th international conference on mobile technology, applications, ACM, New York, NY, USA, 2007, pp. 39-46.

[8]. R. Want, A. Hopper, V. Falcao and J. Gibbons, "The Active Badge Location System", ACM Transactions on Information Systems, Jan. 1992, vol. 10, no. 1, pp. 91-102.

[9]. B. N. Schilit, D. M. Hilbert, and J. Trevor, 'Context-Aware Communication'. IEEE Wireless Communication, 2002, pp. 46-54.

[10]. K. Chaverst, K. Mitchell, and N. Davies, "Design of an Object Model for a Context Sensitive Tourist GUIDE," Computers and Graphics, 1998, vol. 23, no. 6, pp. 883-891.

[11]. G. D. Abowd, et al., "Cyberguide: a mobile Context-Aware Tour Guide," ACM Wireless Networks,Oct. 1997, vol 3, no. 5, pp. 421-433.

[12]. M. Roman, et al., 'Gaia: A Middleware to Enable Active Spaces', IEEE Pervasive Computing, 2002, pp. 74-83.

[13]. M. Riaz, S. L. Kiani, S. Lee, S. Han and Y. Lee, 'Service Delivery in Context Aware Environments: Lookup and Access Control Issues', Proc. RTSCA'05, 2005.

[14]. H. Chen, T. Finin and A. Joshi, 'An Intelligent Broker for Context-Aware Systems', Proc. UbiComp 2003, 2003, pp. 183-184.

[15]. M. Samulowitz, F. Michahelles and C. Linnhoff-Popien, 'CAPEUS: An Architecture for Context-Aware Selection and Execution of Services', in New developments in distributed applications and interoperable systems, 2001, pp. 23 – 39.

[16]. U. Mahmud, N. Iltaf, A. Rehman, and F. Kamran, 'Context-Aware Paradigm for a Pervasive Computing Environment (CAPP),' Proc. WWW/Internet 2007, Villa Real, Portugal, Oct 2007,vol. 1, pp. 337-346.

[17]. T. Gu, H. K. Pung and D. Q. Zhang, 'A Middleware for Building Context-Aware Mobile Services',Proc. IEEE Vehicular Technology Conference, LA, USA. 2004, pp. 2656-2660.

[18]. P. Fahy, and S. Clarke, 'CASS A Middleware for Mobile Context-Aware Applications', Proc. MobiSys 2004, Boston, USA. 2004.

[19]. Y. K. Wang, 'Context Awareness and Adaptation in Mobile Learning', Proc. 2nd IEEE International Workshop on Wireless and Mobile technologies in Education (WMTE'04), 2004.

[20]. S. Deugd, R. Carroll, K. E. Kelly, B. Millett and J. Ricker, 'SODA: Service-Oriented Device Architecture', IEEE journal of Pervasive Computing, 2006, pp. 94-97.

[21]. P. Korpipaa, J. Mantyjarvi, J. Kela, H. Keranen and E. Malm, 'Managing Context Information in Mobile Devices', IEEE Pervasive Computing Journal, 2003, pp.42 - 51.

[22]. T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger and J. Altmann, 'Context-Awareness on Mobile Devices – The Hydrogen Approach", Proc. 36th Annual Hawaii International Conference on System Sciences, 2002, pp. 292-302.

[23]. Greg Linden, Brent Smith, Jeremy York, "Amazon.com Recommendations: item-to-item collaborative filtering", Internet Computing, IEEE, Jan 2003, vol. 7, no. 1, pp. 76-80.

[24]. Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl, "Item-based collaborative filtering recommendation algorithms ", Proc. 10th international conference on World Wide Web (WWW '01). ACM, New York, NY, USA, 2001, pp. 285-295.

[25]. J. Ben Schafer, Joseph Konstan, John Riedi, "Recommender systems in e-commerce", Proc. 1st ACM conference on Electronic commerce (EC '99). ACM, New York, NY, USA, 1999, pp.158-166.

[26]. Xiaoyuan Su, Taghi M. Khoshgoftaar, "A survey of collaborative filtering techniques", Advances in Artificial Intelligence 2009, 2009, pp. 4:2 – 4:2.

[27]. A. Ansari, S. Essegaier, R. Kohli, "Internet recommendation systems," Journal of Marketing Research, 2000, vol. 37, no. 3, pp. 363–375.

[28]. Reyn Nakamoto, Shinsuke Nakajima, Jun Miyazaki, Shunsuke Uemura, "Tag-Based Contextual Collaborative Filtering", IAENG International Journal of Computer Science, Nov. 2007, vol. 34, no. 2.

[29]. Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, Alexander Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach", ACM Trans. on Information Systems , Jan 2005, vol. 23,no. 1 pp. 103-145.

[30]. Gediminas Adomavicius, Alexander Tuzhilin "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", IEEE Trans. on Knowledge and Data Eng., Jun 2005,vol. 17, no. 6, pp. 734-749.

[31]. Van Setten, Mark, Pokraev, Stanislav,Koolwaaij, Johan, "Context-Aware Recommendations in the Mobile Tourist Application COMPASS", Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science, vol. 3137, pp. 515-548.

[32]. Dhoha Almazro, Ghadeer Shahatah, Lamia Albdulkarim, Mona Kherees, Romy Martinez, William Nzoukou, "A Survey Paper on Recommender Systems", CoRR, 2010.

[33]. Zhiwen Yu, Xingshe Zhou, Daqing Zhang, Chung-Yau Chin, Xiaohang Wang, Ji Men. "Supporting Context-Aware Media Recommendations for Smart Phones", IEEE Pervasive Computing, Jul. 2006, vol. 5, no. 3, pp. 68-75.

[34]. Social Genome | @Walmart Labs - http://www.walmartlabs.com/social-genome/

[35]. Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu,Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, Andrew T. Campbell, "Sensing Meets Mobile Social Networks: The Design,Implementation and Evaluation of the CenceMe Application", Proc. 6th ACM conference on Embedded network sensor systems,Nov. 2008, pp. 337-350

[36]. Hans W. Gellersen,Albercht Schmidt, Michael Beigl, "Multi-sensor context-awareness in mobile devices and smart artifacts.", Mobile Networks and Applications, Oct. 2002, vol. 7, no. 5, pp. 341-351.

[37]. Discover Facebook Pages - https://www.facebook.com/pages/browser.php

[38]. PHP SDK Overview - Facebook Developers – https://developers.facebook.com/docs/reference/php/

[39]. Javascript SDK Overview - Facebook Developers - https://developers.facebook.com/docs/reference/javascript/

[40]. Graph API - https://developers.facebook.com/docs/reference/api/

[41]. Van Eck Nees Jan, Waltman Ludo, "Appropriate Similarity Measures for Author Co-Citation Analysis", Journal of the American Society for Information Science and Technology, 2008, vol.59 no.10, pp. 1653–1661.

PUBLICATIONS

I

Status: Accepted & In Press (to be published by IEEE Computer Society Press)

Title: Categorizing Context and Using Short Term Contextual Information to obtain Long Term Context.

Conference: The 11th IEEE International Conference on Ubiquitous Computing and Communications – Liverpool, UK

II

Status: Communicated

Title: A Social Context based Pre-filtering Algorithm for Context Aware Recommender Systems

Conference: 6th ACM RecSys Workshop on Context Aware Recommender Systems – Dublin, Ireland.

APPENDIX A
EXPLICIT DATA COLLECTED

Explicit Data Collection - Ratings Maximum Mutual Likes - Top 16 Users					
rank	itemname	rating	rank.	itemname.	rating.
1	Android	9	9	Android	8
1	Sachin Tendulkar	10	9	Sachin Tendulkar	5
1	New York	10	9	New York	4
1	The Beatles	8	9	The Beatles	8
1	Google	9	9	Google	9
1	Sholay	10	9	Sholay	6
1	Jeffrey Archer	6	9	Jeffrey Archer	4
1	Dan Brown	7	9	Dan Brown	3
1	Goldman Sachs	5	9	Goldman Sachs	8
1	Engineering	5	9	Engineering	8
1	Barrack Obama	8	9	Barrack Obama	8
1	Mashable	8	9	Mashable	8
1	Incredible India	9	9	Incredible India	9
1	Facebook	7	9	Facebook	7
1	Forrest Gump	10	9	Forrest Gump	10
1	Harry Potter	9	9	Harry Potter	7
1	Anna Hazare	7	9	Anna Hazare	5
1	Adolf Hitler	5	9	Adolf Hitler	8
1	AR Rehman	6	9	AR Rehman	9
1	Adventure Sports	10	9	Adventure Sports	10
2	Android	9	10	Android	7
2	Sachin Tendulkar	5	10	Sachin Tendulkar	10
2	New York	4	10	New York	10
2	The Beatles	8	10	The Beatles	8
2	Google	9	10	Google	9
2	Sholay	1	10	Sholay	10
2	Jeffrey Archer	4	10	Jeffrey Archer	6
2	Dan Brown	7	10	Dan Brown	7
2	Goldman Sachs	10	10	Goldman Sachs	9
2	Engineering	2	10	Engineering	5
2	Barrack Obama	8	10	Barrack Obama	8
2	Mashable	8	10	Mashable	8
2	Incredible India	9	10	Incredible India	5
2	Facebook	7	10	Facebook	7
2	Forrest Gump	10	10	Forrest Gump	8
2	Harry Potter	9	10	Harry Potter	9
2	Anna Hazare	7	10	Anna Hazare	9
2	Adolf Hitler	8	10	Adolf Hitler	4
2	AR Rehman	9	10	AR Rehman	6
2	Adventure Sports	8	10	Adventure Sports	10
3	Android	9	11	Android	9
3	Sachin Tendulkar	9	11	Sachin Tendulkar	9
3	New York	7	11	New York	7
3	The Beatles	8	11	The Beatles	8
3	Google	9	11	Google	9
3	Sholay	10	11	Sholay	10

3	Jeffrey Archer	6	11	Jeffrey Archer	6
3	Dan Brown	7	11	Dan Brown	7
3	Goldman Sachs	5	11	Goldman Sachs	5
3	Engineering	6	11	Engineering	6
3	Barrack Obama	9	11	Barrack Obama	9
3	Mashable	10	11	Mashable	5
3	Incredible India	9	11	Incredible India	9
3	Facebook	7	11	Facebook	7
3	Forrest Gump	10	11	Forrest Gump	10
3	Harry Potter	9	11	Harry Potter	9
3	Anna Hazare	7	11	Anna Hazare	7
3	Adolf Hitler	10	11	Adolf Hitler	10
3	AR Rehman	6	11	AR Rehman	6
3	Adventure Sports	3	11	Adventure Sports	7
4	Android	6	12	Android	7
4	Sachin Tendulkar	10	12	Sachin Tendulkar	5
4	New York	4	12	New York	8
4	The Beatles	8	12	The Beatles	8
4	Google	9	12	Google	9
4	Sholay	10	12	Sholay	3
4	Jeffrey Archer	4	12	Jeffrey Archer	4
4	Dan Brown	7	12	Dan Brown	7
4	Goldman Sachs	10	12	Goldman Sachs	10
4	Engineering	10	12	Engineering	6
4	Barrack Obama	8	12	Barrack Obama	4
4	Mashable	8	12	Mashable	3
4	Incredible India	9	12	Incredible India	9
4	Facebook	7	12	Facebook	7
4	Forrest Gump	10	12	Forrest Gump	10
4	Harry Potter	9	12	Harry Potter	9
4	Anna Hazare	7	12	Anna Hazare	7
4	Adolf Hitler	8	12	Adolf Hitler	8
4	AR Rehman	9	12	AR Rehman	9
4	Adventure Sports	9	12	Adventure Sports	10
5	Android	7	13	Android	7
5	Sachin Tendulkar	10	13	Sachin Tendulkar	10
5	New York	7	13	New York	9
5	The Beatles	8	13	The Beatles	9
5	Google	9	13	Google	9
5	Sholay	10	13	Sholay	7
5	Jeffrey Archer	6	13	Jeffrey Archer	4
5	Dan Brown	7	13	Dan Brown	7
5	Goldman Sachs	9	13	Goldman Sachs	8
5	Engineering	5	13	Engineering	6
5	Barrack Obama	8	13	Barrack Obama	4
5	Mashable	8	13	Mashable	10
5	Incredible India	5	13	Incredible India	9
5	Facebook	7	13	Facebook	6
5	Forrest Gump	8	13	Forrest Gump	10
5	Harry Potter	9	13	Harry Potter	9
5	Anna Hazare	9	13	Anna Hazare	7
5	Adolf Hitler	4	13	Adolf Hitler	8
5	AR Rehman	6	13	AR Rehman	3
5	Adventure Sports	10	13	Adventure Sports	4
6	Android	10	14	Android	9
6	Sachin Tendulkar	5	14	Sachin Tendulkar	10
6	New York	4	14	New York	9

6	The Beatles	8	14	The Beatles	8
6	Google	9	14	Google	9
6	Sholay	6	14	Sholay	9
6	Jeffrey Archer	4	14	Jeffrey Archer	6
6	Dan Brown	3	14	Dan Brown	7
6	Goldman Sachs	10	14	Goldman Sachs	5
6	Engineering	10	14	Engineering	6
6	Barrack Obama	8	14	Barrack Obama	8
6	Mashable	8	14	Mashable	8
6	Incredible India	9	14	Incredible India	9
6	Facebook	7	14	Facebook	7
6	Forrest Gump	10	14	Forrest Gump	9
6	Harry Potter	3	14	Harry Potter	9
6	Anna Hazare	5	14	Anna Hazare	7
6	Adolf Hitler	8	14	Adolf Hitler	4
6	AR Rehman	9	14	AR Rehman	6
6	Adventure Sports	10	14	Adventure Sports	10
7	Android	7	15	Android	6
7	Sachin Tendulkar	10	15	Sachin Tendulkar	10
7	New York	9	15	New York	4
7	The Beatles	9	15	The Beatles	8
7	Google	9	15	Google	9
7	Sholay	7	15	Sholay	10
7	Jeffrey Archer	4	15	Jeffrey Archer	4
7	Dan Brown	7	15	Dan Brown	7
7	Goldman Sachs	8	15	Goldman Sachs	10
7	Engineering	6	15	Engineering	10
7	Barrack Obama	4	15	Barrack Obama	8
7	Mashable	10	15	Mashable	8
7	Incredible India	9	15	Incredible India	9
7	Facebook	6	15	Facebook	7
7	Forrest Gump	10	15	Forrest Gump	10
7	Harry Potter	9	15	Harry Potter	9
7	Anna Hazare	7	15	Anna Hazare	7
7	Adolf Hitler	8	15	Adolf Hitler	8
7	AR Rehman	3	15	AR Rehman	9
7	Adventure Sports	4	15	Adventure Sports	9
8	Android	7	16	Android	7
8	Sachin Tendulkar	10	16	Sachin Tendulkar	10
8	New York	9	16	New York	10
8	The Beatles	9	16	The Beatles	8
8	Google	9	16	Google	9
8	Sholay	9	16	Sholay	10
8	Jeffrey Archer	4	16	Jeffrey Archer	6
8	Dan Brown	7	16	Dan Brown	7
8	Goldman Sachs	8	16	Goldman Sachs	9
8	Engineering	6	16	Engineering	8
8	Barrack Obama	8	16	Barrack Obama	8
8	Mashable	9	16	Mashable	8
8	Incredible India	9	16	Incredible India	3
8	Facebook	6	16	Facebook	7
8	Forrest Gump	7	16	Forrest Gump	8
8	Harry Potter	9	16	Harry Potter	9
8	Anna Hazare	7	16	Anna Hazare	9
8	Adolf Hitler	8	16	Adolf Hitler	4
8	AR Rehman	5	16	AR Rehman	6
8	Adventure Sports	8	16	Incredible India	10

Explicit Data Collection - Ratings Page Popularity Rating - Top - 16 Users					
rank	**itemname**	**rating**	**rank.**	**itemname.**	**rating.**
1	Android	9	9	Android	7
1	Sachin Tendulkar	10	9	Sachin Tendulkar	10
1	New York	9	9	New York	4
1	The Beatles	8	9	The Beatles	8
1	Google	9	9	Google	9
1	Sholay	9	9	Sholay	10
1	Jeffrey Archer	6	9	Jeffrey Archer	4
1	Dan Brown	7	9	Dan Brown	7
1	Goldman Sachs	5	9	Goldman Sachs	10
1	Engineering	6	9	Engineering	10
1	Barrack Obama	8	9	Barrack Obama	8
1	Mashable	8	9	Mashable	8
1	Incredible India	9	9	Incredible India	9
1	Facebook	7	9	Facebook	7
1	Forrest Gump	9	9	Forrest Gump	10
1	Harry Potter	9	9	Harry Potter	9
1	Anna Hazare	7	9	Anna Hazare	7
1	Adolf Hitler	4	9	Adolf Hitler	8
1	AR Rehman	6	9	AR Rehman	9
1	Adventure Sports	10	9	Adventure Sports	10
2	Android	9	10	Android	6
2	Sachin Tendulkar	9	10	Sachin Tendulkar	10
2	New York	7	10	New York	4
2	The Beatles	8	10	The Beatles	8
2	Google	9	10	Google	9
2	Sholay	10	10	Sholay	10
2	Jeffrey Archer	6	10	Jeffrey Archer	4
2	Dan Brown	7	10	Dan Brown	7
2	Goldman Sachs	5	10	Goldman Sachs	10
2	Engineering	6	10	Engineering	10
2	Barrack Obama	9	10	Barrack Obama	8
2	Mashable	10	10	Mashable	8
2	Incredible India	9	10	Incredible India	9
2	Facebook	7	10	Facebook	7
2	Forrest Gump	10	10	Forrest Gump	10
2	Harry Potter	9	10	Harry Potter	9
2	Anna Hazare	7	10	Anna Hazare	7
2	Adolf Hitler	10	10	Adolf Hitler	8
2	AR Rehman	6	10	AR Rehman	9
2	Adventure Sports	3	10	Adventure Sports	9
3	Android	7	11	Android	9
3	Sachin Tendulkar	10	11	Sachin Tendulkar	9
3	New York	10	11	New York	7
3	The Beatles	8	11	The Beatles	8
3	Google	9	11	Google	9
3	Sholay	10	11	Sholay	10
3	Jeffrey Archer	6	11	Jeffrey Archer	6
3	Dan Brown	7	11	Dan Brown	7
3	Goldman Sachs	9	11	Goldman Sachs	5
3	Engineering	5	11	Engineering	6
3	Barrack Obama	8	11	Barrack Obama	9
3	Mashable	8	11	Mashable	8
3	Incredible India	5	11	Incredible India	9

3	Facebook	7	11	Facebook	7
3	Forrest Gump	8	11	Forrest Gump	10
3	Harry Potter	9	11	Harry Potter	9
3	Anna Hazare	9	11	Anna Hazare	7
3	Adolf Hitler	4	11	Adolf Hitler	10
3	AR Rehman	6	11	AR Rehman	6
3	Adventure Sports	10	11	Adventure Sports	9
4	Android	9	12	Android	7
4	Sachin Tendulkar	10	12	Sachin Tendulkar	5
4	New York	10	12	New York	8
4	The Beatles	8	12	The Beatles	8
4	Google	9	12	Google	9
4	Sholay	10	12	Sholay	3
4	Jeffrey Archer	6	12	Jeffrey Archer	4
4	Dan Brown	7	12	Dan Brown	7
4	Goldman Sachs	5	12	Goldman Sachs	10
4	Engineering	5	12	Engineering	6
4	Barrack Obama	8	12	Barrack Obama	4
4	Mashable	8	12	Mashable	3
4	Incredible India	9	12	Incredible India	9
4	Facebook	7	12	Facebook	7
4	Forrest Gump	10	12	Forrest Gump	10
4	Harry Potter	9	12	Harry Potter	9
4	Anna Hazare	7	12	Anna Hazare	7
4	Adolf Hitler	5	12	Adolf Hitler	8
4	AR Rehman	6	12	AR Rehman	9
4	Adventure Sports	10	12	Adventure Sports	10
5	Android	8	13	Android	7
5	Sachin Tendulkar	7	13	Sachin Tendulkar	10
5	New York	10	13	New York	9
5	The Beatles	8	13	The Beatles	9
5	Google	9	13	Google	9
5	Sholay	10	13	Sholay	7
5	Jeffrey Archer	6	13	Jeffrey Archer	4
5	Dan Brown	7	13	Dan Brown	7
5	Goldman Sachs	5	13	Goldman Sachs	8
5	Engineering	8	13	Engineering	6
5	Barrack Obama	8	13	Barrack Obama	4
5	Mashable	8	13	Mashable	10
5	Incredible India	9	13	Incredible India	9
5	Facebook	7	13	Facebook	6
5	Forrest Gump	7	13	Forrest Gump	10
5	Harry Potter	9	13	Harry Potter	9
5	Anna Hazare	7	13	Anna Hazare	7
5	Adolf Hitler	4	13	Adolf Hitler	8
5	AR Rehman	6	13	AR Rehman	3
5	Adventure Sports	7	13	Adventure Sports	4
6	Android	7	14	Android	7
6	Sachin Tendulkar	9	14	Sachin Tendulkar	5
6	New York	4	14	New York	8
6	The Beatles	8	14	The Beatles	8
6	Google	9	14	Google	9
6	Sholay	10	14	Sholay	10
6	Jeffrey Archer	4	14	Jeffrey Archer	4
6	Dan Brown	7	14	Dan Brown	7
6	Goldman Sachs	9	14	Goldman Sachs	10

6	Engineering	9	14	Engineering	6
6	Barrack Obama	4	14	Barrack Obama	4
6	Mashable	8	14	Mashable	3
6	Incredible India	9	14	Incredible India	9
6	Facebook	7	14	Facebook	7
6	Forrest Gump	10	14	Forrest Gump	10
6	Harry Potter	9	14	Harry Potter	9
6	Anna Hazare	7	14	Anna Hazare	7
6	Adolf Hitler	8	14	Adolf Hitler	8
6	AR Rehman	9	14	AR Rehman	9
6	Adventure Sports	10	14	Adventure Sports	10
7	Android	9	15	Android	7
7	Sachin Tendulkar	5	15	Sachin Tendulkar	10
7	New York	4	15	New York	10
7	The Beatles	8	15	The Beatles	8
7	Google	9	15	Google	9
7	Sholay	1	15	Sholay	10
7	Jeffrey Archer	4	15	Jeffrey Archer	6
7	Dan Brown	7	15	Dan Brown	7
7	Goldman Sachs	10	15	Goldman Sachs	9
7	Engineering	2	15	Engineering	5
7	Barrack Obama	8	15	Barrack Obama	8
7	Mashable	8	15	Mashable	8
7	Incredible India	9	15	Incredible India	5
7	Facebook	7	15	Facebook	7
7	Forrest Gump	10	15	Forrest Gump	8
7	Harry Potter	9	15	Harry Potter	9
7	Anna Hazare	7	15	Anna Hazare	9
7	Adolf Hitler	8	15	Adolf Hitler	4
7	AR Rehman	9	15	AR Rehman	6
7	Adventure Sports	8	15	Adventure Sports	10
8	Android	6	16	Android	7
8	Sachin Tendulkar	10	16	Sachin Tendulkar	10
8	New York	4	16	New York	10
8	The Beatles	8	16	The Beatles	8
8	Google	9	16	Google	9
8	Sholay	10	16	Sholay	10
8	Jeffrey Archer	4	16	Jeffrey Archer	6
8	Dan Brown	7	16	Dan Brown	7
8	Goldman Sachs	10	16	Goldman Sachs	9
8	Engineering	10	16	Engineering	8
8	Barrack Obama	8	16	Barrack Obama	8
8	Mashable	8	16	Mashable	8
8	Incredible India	9	16	Incredible India	3
8	Facebook	7	16	Facebook	7
8	Forrest Gump	10	16	Forrest Gump	8
8	Harry Potter	9	16	Harry Potter	9
8	Anna Hazare	7	16	Anna Hazare	9
8	Adolf Hitler	8	16	Adolf Hitler	4
8	AR Rehman	9	16	AR Rehman	6
8	Adventure Sports	9	16	Incredible India	10

Explicit Data Collection - Ratings Page Popularity Rating : Top - 16 Users					
rank	itemname	rating	rank.	itemname.	rating.
1	Android	9	9	Android	9
1	Sachin Tendulkar	10	9	Sachin Tendulkar	9
1	New York	9	9	New York	7
1	The Beatles	8	9	The Beatles	8
1	Google	9	9	Google	9
1	Sholay	9	9	Sholay	10
1	Jeffrey Archer	6	9	Jeffrey Archer	6
1	Dan Brown	7	9	Dan Brown	7
1	Goldman Sachs	5	9	Goldman Sachs	5
1	Engineering	6	9	Engineering	6
1	Barrack Obama	8	9	Barrack Obama	9
1	Mashable	8	9	Mashable	10
1	Incredible India	9	9	Incredible India	9
1	Facebook	7	9	Facebook	7
1	Forrest Gump	9	9	Forrest Gump	10
1	Harry Potter	9	9	Harry Potter	9
1	Anna Hazare	7	9	Anna Hazare	7
1	Adolf Hitler	4	9	Adolf Hitler	10
1	AR Rehman	6	9	AR Rehman	6
1	Adventure Sports	10	9	Adventure Sports	9
2	Android	7	10	Android	9
2	Sachin Tendulkar	10	10	Sachin Tendulkar	7
2	New York	10	10	New York	5
2	The Beatles	8	10	The Beatles	10
2	Google	9	10	Google	9
2	Sholay	10	10	Sholay	9
2	Jeffrey Archer	6	10	Jeffrey Archer	6
2	Dan Brown	7	10	Dan Brown	7
2	Goldman Sachs	9	10	Goldman Sachs	10
2	Engineering	5	10	Engineering	10
2	Barrack Obama	8	10	Barrack Obama	8
2	Mashable	8	10	Mashable	8
2	Incredible India	5	10	Incredible India	9
2	Facebook	7	10	Facebook	10
2	Forrest Gump	8	10	Forrest Gump	10
2	Harry Potter	9	10	Harry Potter	9
2	Anna Hazare	9	10	Anna Hazare	7
2	Adolf Hitler	4	10	Adolf Hitler	4
2	AR Rehman	6	10	AR Rehman	6
2	Adventure Sports	10	10	Adventure Sports	9
3	Android	9	11	Android	9
3	Sachin Tendulkar	7	11	Sachin Tendulkar	7
3	New York	9	11	New York	5
3	The Beatles	8	11	The Beatles	10
3	Google	9	11	Google	7
3	Sholay	9	11	Sholay	9
3	Jeffrey Archer	6	11	Jeffrey Archer	6
3	Dan Brown	7	11	Dan Brown	7
3	Goldman Sachs	10	11	Goldman Sachs	9
3	Engineering	10	11	Engineering	3
3	Barrack Obama	8	11	Barrack Obama	8
3	Mashable	8	11	Mashable	8
3	Incredible India	9	11	Incredible India	9
3	Facebook	7	11	Facebook	6

3	Forrest Gump	10	11	Forrest Gump	9
3	Harry Potter	9	11	Harry Potter	9
3	Anna Hazare	7	11	Anna Hazare	7
3	Adolf Hitler	4	11	Adolf Hitler	4
3	AR Rehman	6	11	AR Rehman	6
3	Adventure Sports	9	11	Adventure Sports	4
4	Android	8	12	Android	9
4	Sachin Tendulkar	7	12	Sachin Tendulkar	10
4	New York	10	12	New York	10
4	The Beatles	8	12	The Beatles	8
4	Google	9	12	Google	9
4	Sholay	10	12	Sholay	10
4	Jeffrey Archer	6	12	Jeffrey Archer	10
4	Dan Brown	7	12	Dan Brown	7
4	Goldman Sachs	5	12	Goldman Sachs	10
4	Engineering	8	12	Engineering	10
4	Barrack Obama	8	12	Barrack Obama	8
4	Mashable	8	12	Mashable	8
4	Incredible India	9	12	Incredible India	9
4	Facebook	7	12	Facebook	7
4	Forrest Gump	7	12	Forrest Gump	10
4	Harry Potter	9	12	Harry Potter	9
4	Anna Hazare	7	12	Anna Hazare	7
4	Adolf Hitler	4	12	Adolf Hitler	10
4	AR Rehman	6	12	AR Rehman	10
4	Adventure Sports	7	12	Adventure Sports	10
5	Android	8	13	Android	6
5	Sachin Tendulkar	10	13	Sachin Tendulkar	10
5	New York	5	13	New York	4
5	The Beatles	8	13	The Beatles	8
5	Google	9	13	Google	9
5	Sholay	10	13	Sholay	10
5	Jeffrey Archer	6	13	Jeffrey Archer	4
5	Dan Brown	7	13	Dan Brown	7
5	Goldman Sachs	5	13	Goldman Sachs	10
5	Engineering	5	13	Engineering	10
5	Barrack Obama	8	13	Barrack Obama	8
5	Mashable	8	13	Mashable	8
5	Incredible India	9	13	Incredible India	9
5	Facebook	7	13	Facebook	7
5	Forrest Gump	10	13	Forrest Gump	10
5	Harry Potter	9	13	Harry Potter	9
5	Anna Hazare	7	13	Anna Hazare	7
5	Adolf Hitler	4	13	Adolf Hitler	8
5	AR Rehman	6	13	AR Rehman	9
5	Adventure Sports	6	13	Adventure Sports	9
6	Android	10	14	Android	3
6	Sachin Tendulkar	6	14	Sachin Tendulkar	10
6	New York	7	14	New York	4
6	The Beatles	10	14	The Beatles	8
6	Google	5	14	Google	9
6	Sholay	10	14	Sholay	10
6	Jeffrey Archer	6	14	Jeffrey Archer	4
6	Dan Brown	9	14	Dan Brown	7
6	Goldman Sachs	5	14	Goldman Sachs	10
6	Engineering	5	14	Engineering	10
6	Barrack Obama	8	14	Barrack Obama	8

6	Mashable	10	14	Mashable	8
6	Incredible India	9	14	Incredible India	9
6	Facebook	10	14	Facebook	7
6	Forrest Gump	10	14	Forrest Gump	10
6	Harry Potter	9	14	Harry Potter	9
6	Anna Hazare	7	14	Anna Hazare	7
6	Adolf Hitler	4	14	Adolf Hitler	8
6	AR Rehman	6	14	AR Rehman	9
6	Adventure Sports	5	14	Adventure Sports	5
7	Android	9	15	Android	10
7	Sachin Tendulkar	10	15	Sachin Tendulkar	9
7	New York	10	15	New York	4
7	The Beatles	8	15	The Beatles	8
7	Google	9	15	Google	9
7	Sholay	10	15	Sholay	10
7	Jeffrey Archer	6	15	Jeffrey Archer	4
7	Dan Brown	7	15	Dan Brown	7
7	Goldman Sachs	10	15	Goldman Sachs	5
7	Engineering	10	15	Engineering	7
7	Barrack Obama	8	15	Barrack Obama	8
7	Mashable	8	15	Mashable	8
7	Incredible India	9	15	Incredible India	9
7	Facebook	7	15	Facebook	7
7	Forrest Gump	10	15	Forrest Gump	10
7	Harry Potter	9	15	Harry Potter	6
7	Anna Hazare	7	15	Anna Hazare	7
7	Adolf Hitler	10	15	Adolf Hitler	8
7	AR Rehman	10	15	AR Rehman	9
7	Adventure Sports	10	15	Adventure Sports	8
8	Android	9	16	Android	5
8	Sachin Tendulkar	9	16	Sachin Tendulkar	10
8	New York	7	16	New York	4
8	The Beatles	8	16	The Beatles	8
8	Google	9	16	Google	9
8	Sholay	10	16	Sholay	10
8	Jeffrey Archer	6	16	Jeffrey Archer	4
8	Dan Brown	7	16	Dan Brown	7
8	Goldman Sachs	5	16	Goldman Sachs	10
8	Engineering	6	16	Engineering	10
8	Barrack Obama	9	16	Barrack Obama	8
8	Mashable	8	16	Mashable	8
8	Incredible India	9	16	Incredible India	9
8	Facebook	7	16	Facebook	7
8	Forrest Gump	10	16	Forrest Gump	10
8	Harry Potter	9	16	Harry Potter	9
8	Anna Hazare	7	16	Anna Hazare	7
8	Adolf Hitler	10	16	Adolf Hitler	8
8	AR Rehman	6	16	AR Rehman	9
8	Adventure Sports	9	16	Incredible India	3

Explicit Data Collection - Ratings Multidimensional Approach: Top - 16 Users					
rank	itemname	rating	rank	itemname	rating
1	Android	9	9	Android	8
1	Sachin Tendulkar	10	9	Sachin Tendulkar	7
1	New York	10	9	New York	10
1	The Beatles	8	9	The Beatles	8
1	Google	9	9	Google	9
1	Sholay	10	9	Sholay	10
1	Jeffrey Archer	6	9	Jeffrey Archer	6
1	Dan Brown	7	9	Dan Brown	7
1	Goldman Sachs	5	9	Goldman Sachs	5
1	Engineering	5	9	Engineering	8
1	Barrack Obama	8	9	Barrack Obama	8
1	Mashable	8	9	Mashable	8
1	Incredible India	9	9	Incredible India	9
1	Facebook	7	9	Facebook	7
1	Forrest Gump	10	9	Forrest Gump	7
1	Harry Potter	9	9	Harry Potter	9
1	Anna Hazare	7	9	Anna Hazare	7
1	Adolf Hitler	5	9	Adolf Hitler	4
1	AR Rehman	6	9	AR Rehman	6
1	Adventure Sports	10	9	Adventure Sports	7
2	Android	9	10	Android	9
2	Sachin Tendulkar	10	10	Sachin Tendulkar	7
2	New York	10	10	New York	8
2	The Beatles	8	10	The Beatles	8
2	Google	9	10	Google	10
2	Sholay	10	10	Sholay	5
2	Jeffrey Archer	6	10	Jeffrey Archer	6
2	Dan Brown	7	10	Dan Brown	7
2	Goldman Sachs	6	10	Goldman Sachs	5
2	Engineering	5	10	Engineering	5
2	Barrack Obama	8	10	Barrack Obama	8
2	Mashable	8	10	Mashable	10
2	Incredible India	9	10	Incredible India	10
2	Facebook	7	10	Facebook	10
2	Forrest Gump	10	10	Forrest Gump	10
2	Harry Potter	9	10	Harry Potter	9
2	Anna Hazare	7	10	Anna Hazare	7
2	Adolf Hitler	4	10	Adolf Hitler	4
2	AR Rehman	6	10	AR Rehman	6
2	Adventure Sports	9	10	Adventure Sports	10
3	Android	9	11	Android	7
3	Sachin Tendulkar	10	11	Sachin Tendulkar	9
3	New York	9	11	New York	10
3	The Beatles	8	11	The Beatles	8
3	Google	9	11	Google	5
3	Sholay	9	11	Sholay	5
3	Jeffrey Archer	6	11	Jeffrey Archer	6
3	Dan Brown	7	11	Dan Brown	7
3	Goldman Sachs	5	11	Goldman Sachs	5
3	Engineering	6	11	Engineering	5
3	Barrack Obama	8	11	Barrack Obama	8
3	Mashable	8	11	Mashable	4
3	Incredible India	9	11	Incredible India	9
3	Facebook	7	11	Facebook	7

3	Forrest Gump	9	11	Forrest Gump	10
3	Harry Potter	9	11	Harry Potter	9
3	Anna Hazare	7	11	Anna Hazare	7
3	Adolf Hitler	4	11	Adolf Hitler	4
3	AR Rehman	6	11	AR Rehman	6
3	Adventure Sports	10	11	Adventure Sports	8
4	Android	9	12	Android	9
4	Sachin Tendulkar	10	12	Sachin Tendulkar	9
4	New York	10	12	New York	7
4	The Beatles	8	12	The Beatles	8
4	Google	9	12	Google	9
4	Sholay	10	12	Sholay	10
4	Jeffrey Archer	6	12	Jeffrey Archer	6
4	Dan Brown	7	12	Dan Brown	7
4	Goldman Sachs	5	12	Goldman Sachs	5
4	Engineering	5	12	Engineering	6
4	Barrack Obama	6	12	Barrack Obama	9
4	Mashable	8	12	Mashable	8
4	Incredible India	9	12	Incredible India	9
4	Facebook	7	12	Facebook	7
4	Forrest Gump	10	12	Forrest Gump	10
4	Harry Potter	9	12	Harry Potter	9
4	Anna Hazare	8	12	Anna Hazare	7
4	Adolf Hitler	4	12	Adolf Hitler	10
4	AR Rehman	6	12	AR Rehman	6
4	Adventure Sports	8	12	Adventure Sports	9
5	Android	9	13	Android	9
5	Sachin Tendulkar	10	13	Sachin Tendulkar	9
5	New York	9	13	New York	7
5	The Beatles	8	13	The Beatles	8
5	Google	9	13	Google	9
5	Sholay	9	13	Sholay	10
5	Jeffrey Archer	6	13	Jeffrey Archer	6
5	Dan Brown	7	13	Dan Brown	7
5	Goldman Sachs	5	13	Goldman Sachs	5
5	Engineering	5	13	Engineering	6
5	Barrack Obama	8	13	Barrack Obama	9
5	Mashable	9	13	Mashable	10
5	Incredible India	9	13	Incredible India	9
5	Facebook	7	13	Facebook	7
5	Forrest Gump	10	13	Forrest Gump	10
5	Harry Potter	9	13	Harry Potter	9
5	Anna Hazare	7	13	Anna Hazare	7
5	Adolf Hitler	4	13	Adolf Hitler	10
5	AR Rehman	6	13	AR Rehman	6
5	Adventure Sports	7	13	Adventure Sports	9
6	Android	8	14	Android	9
6	Sachin Tendulkar	10	14	Sachin Tendulkar	9
6	New York	5	14	New York	7
6	The Beatles	8	14	The Beatles	8
6	Google	9	14	Google	9
6	Sholay	10	14	Sholay	10
6	Jeffrey Archer	6	14	Jeffrey Archer	6
6	Dan Brown	7	14	Dan Brown	7
6	Goldman Sachs	5	14	Goldman Sachs	10
6	Engineering	5	14	Engineering	6
6	Barrack Obama	8	14	Barrack Obama	6

6	Mashable	8	14	Mashable	10
6	Incredible India	9	14	Incredible India	9
6	Facebook	7	14	Facebook	7
6	Forrest Gump	10	14	Forrest Gump	10
6	Harry Potter	9	14	Harry Potter	9
6	Anna Hazare	7	14	Anna Hazare	7
6	Adolf Hitler	4	14	Adolf Hitler	9
6	AR Rehman	6	14	AR Rehman	6
6	Adventure Sports	6	14	Adventure Sports	10
7	Android	7	15	Android	9
7	Sachin Tendulkar	10	15	Sachin Tendulkar	7
7	New York	10	15	New York	7
7	The Beatles	8	15	The Beatles	10
7	Google	9	15	Google	9
7	Sholay	10	15	Sholay	9
7	Jeffrey Archer	6	15	Jeffrey Archer	6
7	Dan Brown	7	15	Dan Brown	7
7	Goldman Sachs	9	15	Goldman Sachs	10
7	Engineering	5	15	Engineering	10
7	Barrack Obama	8	15	Barrack Obama	8
7	Mashable	8	15	Mashable	8
7	Incredible India	5	15	Incredible India	9
7	Facebook	7	15	Facebook	10
7	Forrest Gump	8	15	Forrest Gump	10
7	Harry Potter	9	15	Harry Potter	9
7	Anna Hazare	9	15	Anna Hazare	7
7	Adolf Hitler	4	15	Adolf Hitler	4
7	AR Rehman	6	15	AR.Rehman	6
7	Adventure Sports	10	15	Adventure Sports	9
8	Android	9	16	Android	9
8	Sachin Tendulkar	10	16	Sachin Tendulkar	9
8	New York	5	16	New York	10
8	The Beatles	8	16	The Beatles	7
8	Google	9	16	Google	9
8	Sholay	10	16	Sholay	7
8	Jeffrey Archer	6	16	Jeffrey Archer	6
8	Dan Brown	7	16	Dan Brown	7
8	Goldman Sachs	5	16	Goldman Sachs	10
8	Engineering	8	16	Engineering	10
8	Barrack Obama	8	16	Barrack Obama	8
8	Mashable	8	16	Mashable	8
8	Incredible India	4	16	Incredible India	9
8	Facebook	7	16	Facebook	10
8	Forrest Gump	10	16	Forrest Gump	10
8	Harry Potter	9	16	Harry Potter	9
8	Anna Hazare	7	16	Anna Hazare	7
8	Adolf Hitler	4	16	Adolf Hitler	10
8	AR Rehman	10	16	AR Rehman	6
8	Adventure Sports	10	16	Incredible India	9