CHAPTER 1
INTRODUCTION

1.1 PROBLEM DEFINITION

- This thesis mainly focuses on the evaluation and development of a web based recommendation system.
- Various features are integrated here in order to improve accuracy by solving data sparsity problem in recommendation systems.
- In this thesis, study the positive impact of using content information in a collaborative filtering approach is addressed by developing content boosted collaborative filtering prediction technique.

1.2 PROBLEM DESCRIPTION

- Recommendation Systems are widely accepted in e-commerce application to solve “Information Overloaded” problem by retrieving the information desired by the user based on his/her similar users’ tastes and preferences.
- It is used by E-commerce sites to suggest products to their customers. It enhances E-commerce sales in three ways [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
  - **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. For instance, a site might recommend additional products in the checkout process, based on those products already in the shopping cart.
  - **Loyalty:** In a world where a site’s competitors are only a click or two away, gaining customer’s loyalty is an essential business strategy. Recommender systems improve loyalty by creating a value-added relationship between the site and the customer. The more a customer uses the recommendation system teaching it what they want the more loyal they are to the site.
1. Introduction

- Recommended System Provide us a virtualization of Sales People rather than a Marketing tool. It is used in many e-commerce website like-Table 1:

<table>
<thead>
<tr>
<th>Web-Site</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon.com</td>
<td>Book, CD etc.</td>
</tr>
<tr>
<td>Moviefinder.com</td>
<td>Movie</td>
</tr>
<tr>
<td>Ebay.com</td>
<td>Clothing, Laptop</td>
</tr>
<tr>
<td>Netflix.com</td>
<td>DVD</td>
</tr>
<tr>
<td>CDNow.com</td>
<td>CD</td>
</tr>
</tbody>
</table>

- Some of the use cases where a recommender system is considered to be useful are [2]:
  - Filtering information to provide only worth-consuming information, predicting and distinguishing between desired and undesired content
  - Finding good items among all those present in the system. There exist special cases where users want or need to reduce the amount of not-good-enough items.
  - Experimenting with the system, just for pleasure of doing so, or to test whether the recommender system is able to capture user preferences correctly or not.
  - Social Users use the recommender system to contribute to its improvement or to benefit from the community, even influencing and introducing bias into the system

- Recommender Systems create great impact in e-commerce. Because of all these factors, we defined it as the overlying aim is to begin the work with Recommendation System.
- Today, almost all recommender system applications use a rating data in order to formalize evaluation of a product by a user.
- The output of a Recommender System can be either Prediction or Recommendation. Prediction is a rating value representing the estimated rating of the active user c for item s; where recommendation is a list of N items that is anticipated to be liked by the active user.
- One of the Recommendation technique (CF) approach aims to find similar users in order to recommend items to the active user.
- The recommendation step, similarity calculation depend on the algorithm selection, however, finding similar users by the help of already provided data to common items is the key point in the execution of a generic collaborative algorithm. This is the reason why sparsity of existing data such as sparse user-item matrix degrades the overall performance of the algorithm.
1. Introduction

- When a system has a sparse user-item matrix, number of ratings given to same set of items by any two users will be low and that will cause bad or totally no recommendation. So this is called problem because of data sparsity.
- Data Sparsity is kind of cold start problem.
- As recommendation technique (CF) methods recommend items based on users’ past preferences, new users will need to rate sufficient number of items to enable the system to capture their preferences accurately and thus provides reliable recommendations.
- Similarly, new items also have the same problem. When new items are added to system, they need to be rated by substantial number of users before they could be recommended to users who have similar tastes with the ones rated them. For active web application, every moment a new user can register to the system and a new item can be added to item list. All of these facts increase the dimensions of user item matrix and causes more sparse matrices.
- The aim of this research work is to predict accurate recommendation using hybrid recommendation approach. It is deal with sparse data rating and improves accuracy of system because customers required accurate prediction. Task of this research work is to design hybrid recommendation engine to recommend top N products. Variety of features are added here in order to improve accuracy using Content-boosted Collaborative filtering approach [3].

1.3 PENDING RESEARCH ISSUES

In this section I have described limitation raise in recommendation system from[4] [5] as below:

(1) Ramp up/Cold Start Problem:

New user: When a new user signs up to a recommendation system, there is only little information about that user. So, it is very difficult for the system to produce realistic recommendations.

New item: This problem is seen when there is a newly added item to the system. In this situation, there is not enough feedback that is provided for that item by users.

Cold start: This problem occurs when a new user or item has just entered the system; it is difficult to find similar ones because there is not enough information. So the recommender system is unable to guess their interests.

(2) Data sparsity:

The data sparsity challenge appears in several situations, specifically, when the cold start problem occurs. Coverage can be defined as the percentage of items that the algorithm could
provide recommendations for. The reduced coverage problem occurs when the number of users’ ratings may be very small compared with the large number of items in the system, and the recommender system may be unable to generate recommendations for them. Neighbor transitivity refers to a problem with sparse databases, in which users with similar tastes may not be identified as such if they have not both rated any of the same items.

(3) Scalability:
When numbers of existing users and items grow tremendously, several recommender techniques will suffer serious scalability problems, with computational resources going beyond practical or acceptable levels. For example, with tens of millions of customers ($M$) and millions of distinct catalogue items ($N$), algorithm with the complexity of $O(n)$ is already too large. As internet contain massive information, it is difficult to recommend item in less amount of time because of scalability issue.

(4) Synonym:
Synonymy refers to the tendency of a number of the same or very similar items to have different names or entries. Most recommender systems are unable to discover this latent association and thus treat these products differently. For example, the seemingly different items “children movie” and “children film” are actual the same item.

(5) Shilling attacks:
Anyone can provide recommendation; people may give tons of positive recommendation for their own material and negative recommendation for their competitor.

(6) Gray sheep:
A user, whose opinion does not consistently agree or disagree with any group people,

(7) Black Sheep:
Opposite group whose idiosyncratic taste makes recommendation nearly impossible?

1.4 MOTIVATION FOR THE RESEARCH WORK

- Recommender systems are changing from novelties used by a few E-commerce sites, to serious business tools that are re-shaping the world of E-commerce.
- Many of the largest commerce Web sites are already using recommender systems to help their customers find products to purchase [1].
1. Introduction

- Types of recommendation techniques are: CF, CB, and Demographic, utility based and Knowledge-based.
- Every recommendation techniques has several advantages and disadvantages so it is better to combine several techniques to improve its performance because of that research work devoted toward hybrid approach.
- CB technique has several issues like start-up problem which required enough information to build a reliable classifier and it is also limited by feature explicitly associated with the object which may have recommended while CF can recommended object without any descriptive data so by combining both we can get better recommended option.
- So in this thesis, we aim to make recommendation for user comfortable by presenting a content boosted collaborative filtering approach for recommender systems

1.5 NEED ANALYSIS

- Recently, Internet has become one of the parts of our lives and its popularity is increasing day by day. This popularity has caused the Internet to contain huge amount of information in a wide range of topics. It creates problem of information overloading where users would not be able to reach the information that they really need.
- E-commerce websites help consumers to discover products of interest from huge data available over the Internet. These websites use recommender systems to generate personalized suggestions to consumers on how to find relevant items from the large number of choices.
- Many systems and approaches make it possible for the users to be guided by the recommendations they provide about new items such as news, web pages, articles, books, music, and movies.
- In the past decade, lot of work has been done on recommender systems and various techniques have been developed.
- Every recommendation technique has several advantages and disadvantages so it is better to combine several techniques to improve its performance because of that this thesis work devoted toward hybrid approach.
- Content based techniques have several issues like start-up problem which required enough information to build a reliable classifier and it is also limited by feature explicitly associated with the object which may have recommended while Collaborative filter can recommended object without any descriptive data so by combining both we can get better recommended option.
For that, main focus is here on content-boosted collaborative filtering which combine content based and collaborative filtering using hybrid approach.

1.6 OBJECTIVES AND GOAL OF THE RESEARCH WORK

- RS are a key way to automate mass customization for E-commerce sites. Recommender systems made a significant progress over the last decade when numerous content-based, collaborative and hybrid methods were proposed and several “industrial-strength” systems have been developed and also it suffer from several issues.
- This fact has provided incentive for research in hybrid recommender systems that combine techniques for improved performance.
- The objective of this research work is:
  - To apply content based algorithm on sparse user rating matrix.
  - To apply output of content based algorithm as input of collaborative algorithm.
  - To apply Naïve Bayes and Pearson Correlation coefficient to improve the predictive performance of Recommendation system.
1. Introduction

1.7 EXAMPLE OF RECOMMENDER SYSTEM

Fig. 1.1: Amazon recommends products to customers by customizing CF systems [36]

Fig. 1.2: Recommendation of book in Amazon website [37]
1. Introduction

1.8 RECOMMENDER SYSTEMS FUNCTION

RSs, software tools and techniques which provide users, suggestions for the concerned items a user wish to utilize, has been defined in the previous section. Further, we refine the RS definition describing maximum possible roles that can be played through illustration. Initially, we have to distinguish the roles played on behalf of service provider and the role of RS. For example, A travel intermediary (Expedia.com) introduces the travel recommender system or a destination management organization (Visitfinland.com) to grow the turnover (Expedia) in reference to boost the sales of hotel rooms and visitors to the destination. [6] While, the initial interest of the user to have access to two systems is to find the appropriate hotel and interesting details like events or attractions at the time of visiting a destination. In fact, there are ample reasons behind exploiting this technology by the service provider.

- **Boost the number of items for selling:** It may be one of the most essential functions for the Commercial RS as it can be useful for the additional growth compare to the growth in its absence. It is possible due to the recommendations of the items based on user’s needs and wants. Seemingly, the user will identify it after several recommended tries. It has also been seen that non-commercial applications have such goals without any additional cost for the user for the selection of an item. For example, a content network aims to raise the number of news items read on its site. By and large, it can be said that the fundamental aim of introducing a RS is to advance the conversion rate, i.e. to show the comparison of user for both the states: getting information with and without recommendation.

- **Sell more distinct items:** A further advantage of RS is that it is more effective in selecting items which might be possible in absence of recommendation. For example, in a movie RS such as Netflix, the service provider wish to rent all the dvds, present in the catalogue, not only which are the most popular ones. This can be easy with a RS since the service provider cannot afford the risk of advertising movies that are not likely to suit a particular user’s taste. As unpopular movies to the right users are suggested as well as advertised through RS.

- **Increase the user’s contentment:** A properly designed RS can also improve the experience of the user with the site or the application when the recommendations are found interesting, relevant and, with a properly designed human-computer interaction, the user will also have pleasure using the system. The combination of effective, i.e., accurate, recommendations and a usable interface will result into the increment of the user’s
subjective evaluation of the system. This will reflect into the increased system usage and the likelihood that the recommendations will be accepted.

- **Increase user’s reliability:** The website should identify the repeat user and the user should be treated as a valuable visitor. This is a very common feature of a RS since many RSS compute recommendations, leveraging the information received from the user in the earlier interactions, e.g., the ratings of items given by the user. Subsequently, the longer the user interacts with the site, the more refined the user model becomes, i.e., the system representation of the user’s preferences, and the more the recommender output can be effectively customized to match the user’s preferences.

- **Enhanced understanding of user’s needs:** Another essential feature of a RS, which can be leveraged to many other applications, is the depiction of the user’s likings, either collected explicitly or predicted by the system. The service provider may then decide to reuse this knowledge for lots of other goals such as convalescing the management of the item’s stock or production. For example, in the travel domain, destination management organizations can plan to advertise a selected region to new customer sectors or advertise a particular type of promotional message resulted by analysing the data collected by the RS (transactions of the users).

We depicted above some important motivations as to why e-service providers introduce RSs. But users may also want a RS, if their tasks or goals are supported effectively. Her locker et al. [6], in a paper that has become a classical reference in this field, defines eleven popular tasks that a RS can assist in implementing. Some may be considered as the core tasks that are normally associated with a RS, i.e., to offer useful suggestions of items to a user. Others might be considered as more “opportunistic” ways to exploit a RS. As a matter of fact, this task differentiation is very similar to the functions of a search engine. Its primary function is to locate documents that are relevant to the user’s information need, but it can also be used to check the importance of a Web page (looking at the position of the page in the result list of a query) or to uncover the various usages of a word in a gathering of documents.

- **Find Some Good Items:** Recommend to a user some items as a ranked list along with predictions of how much the user would like them (e.g., on a one- to five star scale). This is the main recommendation task that many commercial systems address.
1. Introduction

- **Find and recommend all the relevant items:** Suggest all the items that can comply with the needs of some users. In such cases, it is not adequate to just find some good items. This is especially true when the number of items is relatively small or when the RS is mission-critical, such as in medical or financial applications. In these situations, in addition to the benefit derived from carefully examining all the possibilities, additional explanations that the RS generates can benefit the user from the RS ranking of these items or from.

- **Annotation in context:** Shown an existing context, e.g., a list of items, accentuate some of them depending on the user’s long-term likings. For instance, a TV recommender system might annotate which TV shows displayed in the electronic program guide (EPG) is significance watching.

- **Recommend a sequence:** The idea of RS is to suggest a sequence of items that are gratifying as a whole instead of concentrating on the generation of a single recommendation; typical examples include recommending a TV series; a book on RSs after having recommended a book on data mining; or a compilation of musical tracks [6].

- **Recommend a package:** Recommend a group of items that can serve well together. For example, a travel plan may be combinations of various attractions, destinations, and accommodation services that are located in a delimited area. From the point of view of the user, these various alternatives are be measured and targeted as a single travel destination [6].

- **Just browsing:** The user browse the catalogue without any intention of purchasing an item, may be for the future reference. Here, the role of the recommender is to assist the user to browse the items that are more likely to fall within the scope of the user’s likings for that specific browsing session. This is a task that has been also supported by adaptive hypermedia techniques [6].

- **Find reliable recommender:** Some users do not put their 100% faith in the recommendations by recommender systems as they play with them to see how good they are in making recommendations. Hence, some system may also offer specific functions to let the users test their behaviour in addition to those just required for acquiring recommendations.

- **Improve the profile:** It focuses to the capability of the user to supply (input) information to the recommender system about what the user likes and dislikes. This is a fundamental
task that is strictly necessary to provide personalized recommendations. If the system has no specific knowledge about the active user, it can only provide him with the same recommendations that would be delivered to an “average” user.

- **Communicate with self:** Some users may not be careful about the recommendations at all. Rather, what it is essential to them is that they should be allowed to contribute with their ratings and express their opinions and beliefs. The user’s satisfaction for that activity can still function as leverage for holding the user tightly to the application.

- **Facilitate others:** Some users are happy to share information, e.g., their evaluation of items (ratings), because they believe that the community can get benefits from their sharing. This could be a major motivation for entering information into a recommender system that is not used routinely. For instance, with a car RS, a user, who has already bought a new car, is aware that the rating entered in the system is more likely to be useful for other users rather than for the future purchase of the car.

- **Influence others:** In Web-based RSs, there are users whose main goal is too explicitly influence other users into purchasing particular products. In fact, there are also some malicious users that may use the system just to promote or penalize certain items.

As these a variety of points describe, the role of a RS within an information system can be quite diverse. This diversity calls for the exploitation of a range of different knowledge sources and techniques.

### 1.9 CHALLENGES

Below mentioned additional challenging topics are briefly note that can be considered important for the development of the research on RSs.

- **Scalability of the algorithms with large and real-world datasets.** As the research on fundamental techniques proceeds and matures, it becomes clear that a core issue for RSs is to resolve how to embed the core recommendation techniques in real operational systems and how to deal with massive and dynamic sets of data produced by the communications of users with items (ratings, preferences, reviews, etc.). A solution that works fine when tested off-line on relatively small data sets may become inefficient or even totally inapplicable on very large datasets. New approaches and large-scale evaluation studies are needed. [6].
• Proactive recommender systems, i.e., recommenders that opt to provide recommendation even if not explicitly requested [6]. So far a “pull” model has been found following by the largest majority of the recommender systems [6]; where the user originates the request for a recommendation. In the present emerging scenario today, where computers are ubiquitously accessible and users are always connected, it is quite natural to imagine that a RS can detect implicit requests. It, therefore, is required to predict not only what to recommend, but also when and how to “push” its recommendations. In this way, the RS can function as proactive without being perceived as disturbing.

• Privacy preserving recommender systems [6]. RSs utilize user data to produce personalized recommendations. In the endeavour to build increasingly better recommendations, they collect maximum possible user data. It will certainly have an adverse impact on the privacy of the users and the users may start feeling that the system knows too much about their personalised preferences. Therefore, there is a need to design solutions that will parsimoniously and sensibly access user data. At the same time, these solutions will ensure that knowledge about the users cannot be freely accessed by malicious users.

• Diversity of the items recommended to a target user [6]. In a recommendation list, it is more likely that the user will find a suitable item if there is a certain degree of diversification among the included items. There is often no value in having perfect recommendations for a restricted type of product, if the user has not expressed a narrow set of preferences. There are many situations, especially in the early stage of a recommendation process, in which the users want to explore new and diversified directions. In such cases, the user is using the recommenders a knowledge discovery tool. The research on this topic is still in an early stage, and there is a requirement to characterize the nature of this “diversity”, i.e., whether we are seeking diversity among different recommendation sessions or within a session, and how to merge with the diversity goal with the accuracy of the recommendation.

• Integration of long-term and short-term user preferences in the process of creating a recommendation list [6]. Recommender systems may be divided in two classes: those that build a long-term profile, generated by summative all the user transaction data collected by the system (e.g., collaborative filtering) and those that are more focused on capturing the ephemeral preferences of the user, e.g., as in case-based approaches. Obviously both viewpoints are important and either the precise user task or the availability of items may come under consideration in resolving the preference integration problem. In fact, new dimension of research is needed to build hybrid models that can correctly decide to drift or
not toward the contingent user’s likings when there is enough evidence to suggest that the user’s short-term likings are departing from the long-term ones.

- Generic user models and cross domain recommender systems can mediate user data through different systems and application domains [6]. With the help of generic user model techniques, a single RS can produce recommendations about a variety of items. This is normally impossible for a general RS which can combine more techniques in a hybrid approach, but cannot easily benefit from user preferences collected in one domain to generate recommendations in a different one.

- Distributed recommender systems that operate in open networks [6]. The computational model of the largest majority of RSs sticks to a typical client-server architecture, where the user-client requests recommendations to the server-recommender which replies with the suggestions. This is apparently a severe limitation and suffers from all the classical problems of centralized systems. The emerging scenario of grid or cloud computing can be an excellent opportunity to implement more robust and flexible computational models for RSs.

- Recommender that optimize a sequence of recommendations [6]. We already described that conversational RSs have emerged in the attempt to improve the quality of recommendations supplied by the systems based on a simpler approach: a one-time request/response. Conversational RSs can be further improved by implementing learning capabilities that can optimize not only the items that are recommended but also how the interaction establishes between the user and the system that must unfold in all possible situations.

- Recommenders designed to operate in mobile devices and usage contexts [6]. Mobile computing is emerging as one of the most natural platforms for personal computing. Many recommendation requests are likely to be made when the user is on the move, i.e., at shops or hotels in a visited city. This demands “mobilizing” the user interface and to design computational solutions that can efficiently use the still limited resources (computational power and screen size) of the mobile devices.