What is a Recommender System?

A Recommender system is a software application that helps to find find items that are useful to a user. It is an increasingly common artifact in today's online world. Some common recommender systems are Netflix, Amazon, Imdb etc.

How is a Recommender system different from a search engine?

Traditionally, search engines have focussed more on information retrieval techniques to fulfill user needs. A recommender system on the other hand is more concerned about information filtering techniques whose main aim is to remove redundant or unwanted data before presenting it to the user. However, with the development of these two systems, these two communities are coming back together and now both recommender systems and search engines use similar technologies. For example, a search engine like Google doesn't just display the searched results but also orders them based on user's preferences. Similarly, a recommender system also employs search to better understand the user's needs.

Why do we need a Recommender System ?

This question can be answered from two different perspectives. From the point of view of the seller, it serves as a great sales and marketing tool. While it definitely helps sellers to increase the number of items sold, it also serves as a platform to introduce a range of diverse products to the user which is otherwise not possible under the traditional setup\cite{her}. Also recommender systems are a great tool to elicit user buying preferences that help to launch directed advertising and selling campaigns that are more targeted towards the individual user. Overall, recommender systems increase user satisfaction and trust by providing him a more comfortable and personalised shopping experience.

As the number of items available to the user increase everyday, it becomes harder for him to know about every one of them and make an informed decision while buying them. This is where recommender systems come in for the user. By reducing the product space for the user according to his tastes, they serve to substantially improve his buying capacity \cite{her}. Also, for complex and difficult to navigate search spaces, they serve as a great discovery tool. Suppose a user wishes to purchase a DLSR camera. But he is unaware of the intricacies of its features or the details of its specifications. A recommender system can help solve this problem by telling him the most popular camera, or the top quality one. It can show him what the users think about the product and thus help him arrive at a final buying decision.Some users use the recommender as a platform to express their tastes and preferences to the broader community. They also tend to influence other users and affect their choices.

A recommender system must maintain a delicate balance in trying to balance both the seller and the buyer's needs. While it must be useful and beneficial to the user, it must also not forget its ultimate goal of selling more products and increasing the seller's revenue.

How did it all begin?

A key pioneer in this field was John Riedl - A professor in University of Minnesota who redefined our thinking about these systems. His main contribution was the publication of a phenomenal paper on titled “Grouplens: an open architecture for collaborative filtering of netnews”\cite{group}. Grouplens is a recommender system that was designed to recommend news items to the users on the Usenet. The concepts presented in the paper served as a starting point for research in this field and led to the development of the modern recommenders as we know today. This key paper has more than 4000 citations as of today.

Basic components of a recommender system

While every recommender is unique and has its own internal structure, on a broad level they contain three main components. The first important component of a recommender is its raw data. Real world data is very messy with a lot of incomplete or incorrect information. It is therefore necessary to do some pre-processing on the data before it is fed to the recommender. The next main component is the logic of the recommender. It is where the main algorithms that determine how the data is manipulated to derive recommendations are stored. Acting as an intermediary between these two components, sits the controller or the business layer.

For the data preprocessing stage, distance measures such as jaccard distance can be used. We can also use dimensionality reduction techniques such as PCA to reduce the data to a manageable level \cite{book}. For the analysis part, various classification and clustering algorithms are used.

Types Of Recommender Systems

Recommender systems can be broadly classified into five categories. Let us look into them in more detail

1. Content based recommenders

   These recommender are one of the most basic ones. Their philosophy is that if a user viewed a recommendation for an item in the past, it is likely that he will like items similar to the previous one. Thus, if a similar item is found, recommend it to him.
   So the recommendation problem consists of the following steps:

   1) Analyze items previously rated by the user
   2) Build a user profile based on the items rated by him

3) Recommendation process - Basically match the attributes of the user profile against the attributes of a content object.

By observing the past habits of a user, a profile is built up which then serves as a reference point to compare items against. The content based recommenders have three main components - a profile learner, a content analyser(which does the comparison between a new item and the user profile) and a filter(to present the most relevant recommendations to the user).

By observing the user's previous buying habits, a matrix is built up in the system for that user. The columns of this matrix represent the different items that the user has looked at. The rows represent the various features of the product that the user was interested in. This sort of a representation is called an - item matrix and is used to store the user's profile information. Once we have the item matrix for the user, then the task of recommendation can be solved by classical methods such as :

1) Keyword matching - TF-IDF : Term frequency–inverse document frequency is a popular measure that is used to infer how important a word is in a document. Using this, the most important factor for a user can be inferred and recommendations given accordingly.

2) Automatic feature extraction : There are also systems that automatically infer important features that are necessary for the comparison.

3) Ontology based approaches (Context based matching) : One can also use semantically related information to perform the comparison thus increasing its efficiency. SiteIF is one such system that uses the ontology to determine both english and spanish words simultaneously \cite{site}.

Some of its advantages/Disadvantages are:

Advantages
1) Very personalized
2) Transparency

Disadvantages
1) Cold start problem
2) Limited content discovery
3) Over-specialization

2. Collaborative Filtering based Recommenders

These type of recommenders are more social in nature and take into account the opinions of other users as well when recommending an item to a user.
They are mainly of two types : Memory based Recommenders and Model based Recommenders. There are two types of memory based recommenders : User-User and Item-Item.

User-User

They operate the following way :

- Identify items rated by target user
- Identify the most similar users using Neighborhood methods
- Take those user's ratings and predict target user's ratings

Item-Item

Their method of operation is :

- Identify users who have rated target item i
- Identify which other items were rated by those users
- Compute neighbours based on the similarity between these items
- Predict ratings for the target item

So while Item-Item approach uses the columns of the Item matrix to compute similarity between users, the User-User uses the rows of an item matrix to group similar users.

Model Based collaborative filtering
These class of recommenders take the item matrix for a set of users and instead of directly calculating recommendations from them, build a model using machine learning techniques to learn user preferences. This stage is usually performed offline prior to the recommendation process.

Some common techniques that are used in the building of an offline model include clustering methods (neighbour joining), association rule mining(that are used in data mining to discover information between data using various measures), and the Bayes net approach (belief network approach).

Some of the limitations of collaborative based approaches are :

1) Requires data about users and items.
Limited support for new users (cold start problem)
Also, users do not rate many items

2) Popularity bias - Inherent in the system due to reliance on other user's preferences

3) Unique taste users. Difficult to find unique items.

3. Knowledge Based Recommender Systems

For a good recommendation, not just users and items but the context is important too. Consider a travel recommendation system. If it recommends a beach to a traveler in winter, it is of no use. But traditional recommendation techniques do not have any special provisions for these types of contextual information. Knowledge based or context aware systems try to fill this gap by inferring the need for a recommendation at a particular instant.

It is not always easy to infer the exact meaning of context as it is highly subjective. A paper published in this area lists various scenarios that can be considered as contextual information to a recommender. They include - "life stages of a customer" ie. his age. A recommendation that is suitable to a teenager may not apply to a man of 40 years. Thus age is an important differentiating factor. "Intent to purchase" or spending power is also another important factor. It is worthwhile to spend more time and energy in developing recommendations for a serious user rather than worrying about a user who is "just browsing". Weather can also be a factor that needs to be taken into account while generating real world recommendations. The information as to whether a user is single or has a companion is also contextual information that can be built into a recommendation system to fine tune its recommendations.

So now, instead of concentration only on users and items, we concentrate on the associated context too. For a movie viewing system, this would be,

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

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

Also consider contextual info such as,

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

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

While these systems generate real time predictions based on the current situation, they can get extremely complex and require lots of data processing and storage space. Also, how to obtain the contextual information is not very clear. These recommenders consist of a relatively new and unexplored field.

4. Hybrid Recommender Systems

These recommender systems consist of a combination of two or more recommendation techniques described above. There are three ways in which multiple methods are combined in order to take advantage of their strengths and alleviate their drawbacks.

•Weighted
scores/votes of several recommendation techniques combined together to produce a single recommendation

•Switching
system switches between recommendation techniques depending on the current situation

•Mixed
recommendations from several different recommenders presented at the same time

5. Critiquing Recommender Systems

So far in all the recommender systems, the user is presented with a recommendation that is deemed to be the best fit for him according to his tastes and preferences. What happens when the user is not interested in those recommendations or if he thinks they are bad? Should the recommender take his feedback and try to improve its suggestions or should it start all over again? This is called “Narrowing problem”. Critiquing recommenders address this problem by taking a conversational approach to recommendation. In the initial stage, a suggestion is presented to the user. If he does not like it, feedback is taken and the recommendations are refined to better suit his taste.

By doing so, user feedback becomes an integral part of the recommender system instead of being an optional mechanism. The user feedback is collected by means of a critique - which is a feature of the product that a user can tweak according to his liking. Since critiques for a product are not always independent and affecting one may change others as well, a bunch of unit critiques can be grouped together to form a compound critique. In a real world scenario, both unit and compound critiques are presented to a user.

Critiquing recommenders too have their own pitfalls. They have a very limited product space discovery. They tend to get too narrow and focussed on the user and therefore may miss out on other products. Also, they assume that the user knows exactly what he wants which might not always be the case. They tend to get long winded. While this is good for some situations, it may not suit all situations.

Evaluation Of Recommender Systems

Recommender systems can be evaluated in many ways - based on the properties of recommender systems and under experimental settings.

Based on Properties of Recommender Systems

1) Accuracy - Accuracy is defined as the ratio of the number of successful recommendations versus the total number of recommendations. this metric is not always suitable to measure the effectiveness of a recommender. If a recommender recommends very similar movies to a user, even though it scores high on accuracy, it is not a good recommendation as there is no uniqueness in them

2) Serendipity - To solve the problem mentioned in the previous step, we define a new metric that measures the uniqueness or novelty of a recommendation. The aim is to present to the user as diverse a collection of items as possible while at the same time, not compromising a lot on accuracy. This can be done by penalizing recommendations that correspond to blockbusters etc.

3) Trust - For a successful recommendation, the user has to be able to trust the system. This is an important factor to take into account. Various methods have been devised to measure user trust and improve it in order to make recommendations more valuable. They include the development of trust network based recommendation systems, social recommendation systems that leverage the user's social contacts to add weight to the recommendations and explanation based recommenders. It has been found that when a system explains the meaning of its recommendation to the user, it increases his trust toward the recommender. While this is simple for some recommenders, it becomes very difficult to answer this question in more complex recommenders like knowledge based or hybrid recommenders.

4) User interactiveness - The ultimate users of a recommender system are people and therefore it is imperative that the system is easy to use and intuitive to the user. One must know the group of users that are being targeted by the recommender- novice or experienced and tweak recommendations based on that. We have to keep in mind that the user's needs are constantly evolving and must take them into account. The principles of HCI(Human Computer Interaction) are used to model these user designs.

5) Other factors - Some other factors that must be considered are: Confidentiality, Privacy, Maintainability and Item space coverage.

Under Experimental Settings

There are 2 types of experiments carried out for recommender systems - offline and online

1) Offline Experiments

At this stage, the main algorithm of the recommender is tested under different settings by changing various parameters. There are various issues to consider here.

Advantages
- Low cost.
- Test a number of different approaches

Problems
- Simulate user behavior. How?
  1)Take previous data (Isn't it outdated?)
  2)Algorithm to generate user data (how close is it to the real world?)

Main Use
- Test the core functionality of the recommender

2) Online Experiments

For online testing, A/B style testing is followed where two different versions of the same system ar rolled out to the users. After a certain time, these two versions are tested and the better one is released as the final version.

The offline testing of a recommendation takes place for a couple of days after which it is released for online testing. The online testing process takes anywhere between a couple of weeks to months. In conclusion, testing recommenders is a complex process. As such, there is no single metric or method that suits all. Even the properties of measurements are contradictory to each and involve finding a balance between them based on the ideas of the recommender.