Recommendation Systems.

A brief introduction to recommendation systems, and its applications in social media like Twitter.

Alex Beltran, Enaitz Berrocal, Oihane Cantero.

Contents:

Introduction

As the internet grows and expands each day, countless amounts of data and information are added to it. This, added with the prevalence of online services, have provided internet users access to a really massive amount of information, yet **an average human internet user can only be shown a finite set amount of data/information on a given time**, and furthermore **can only be expected to be interested in a more limited set of that shown information.**

This "What do we show to him" problem is even more concerning when we look at some utilities like advertisements, where knowing and showing the user what they are actually interested in becomes a critical matter, since **no one's gonna buy something they are not remotely interested in.**

Recommendation systems arise as a treatment to this problem. These work as "filters" to **produce suggestions and recommendations** to assist users in many decision-making processes. With the help of these, **users are more likely** to access appropriate products and services such as movies, books, music, food, hotels, and restaurants. Notice the fact that **these systems do not imply the user will be interested in a certain item or service, but they guarantee that there is a bigger chance something the user is interested in will be shown to them.**

Phases of recommendation process

1. ***Information collection phase:***
   The first phase is very important because the recommendation agent can't function accurately until the user profile has been well defined. The agent needs to know as much information as possible from the user to provide good recommendations. There are 3 different ways to access user's information, explicit, implicit and hybrid feedback:

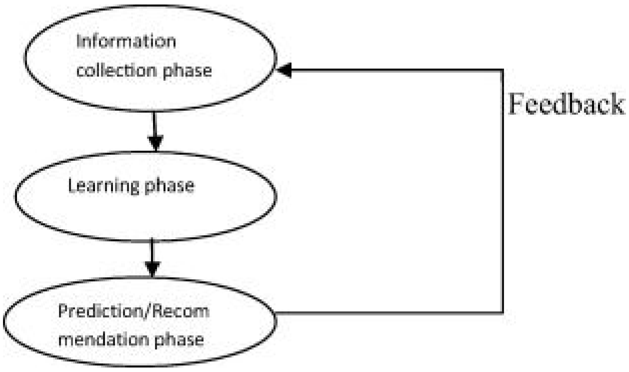
   Explicit feedback: In the explicit feedback, the system will ask the user to rate different items in order to construct and improve the model. The more rating the user makes, the better the system's accuracy will be. The user needs to do more effort (he has to rate items one by one) but in the end, the model will be better and the quality of recommendation will be improved.

   Implicit feedback: Here the system automatically deduces the user's preferences by observing some information: For example the history of purchases, navigation history, how long he spent in different web pages, which links he followed and the content of emails. With this method, the user doesn't need to rate manually items and the system will deduce his likes. Therefore the recommendation's accuracy will be smaller.

Hybrid feedback: This method mixes the previous ones. The user can give his feedback but the system will also learn by observing his information. This way, the user isn't forced to absolutely make a lot of ratings and accuracy is better than in the implicit feedback.

2. ***Learning phase:***
   In this phase, the system will apply a learning algorithm to exploit the information it collected in the previous phase.

3. ***Prediction/recommendation phase:***
   Here, in the last phase, the system will recommend an item to the user, using what it learns.



Challenges of recommender systems.

Yet these recommender systems dont come with their own set of problems, the most substantial ones being:

1. ***Lack of Data and the sparsity problem:***
   The main issue of recommendation systems is that they need immense amounts of data in order to work properly. The more item and user data a recommender system has to work with, the stronger the chances of getting good recommendations.

   The sparsity problem: The sparsity problem is a particular case of Lack of Data. The sparsity problem occurs when transactional or feedback data is sparse and insufficient for identifying neighbors and it is a major issue limiting the quality of recommendations and the applicability of collaborative filtering in general.

2. ***Changing data:***
   Humans work on trends. Something that can be desired by a lot of people in a certain moment in time might have 0 interest in a future moment. This makes past data heavily unreliable, and more so in matters of fashion, where trends heavily mark the user choices. This usually makes the issue of lack of data even bigger, since now we need a lot of data and we need it to be updated with the current state of trends in an area / a city / a region / a country / the world.

3. ***Changes in the users preferences:***
   Continuing with the human behavior examples, how many times did you actually like, for example, an X item that has this complete opposite Y item, only to change preference and like the Y item more after a while. Another example is that one day, a certain person might be searching for an item for himself/herself but another day, that exact same person, might be looking for something for a relative.

4. ***Unpredictable items:***
   Similar yet different to the previous problem, some users might be interested in opposite items at the same time, for example, a hard-core metal fan might also be a fan of Julio iglesias, and a system like spotify needs to actually spot these two totally opposite preferences and take them into account.

Deep learning within recommender systems.

Thanks to the boom of deep learning in the recent years and to the high amount of computational power available nowadays, the introduction of deep neural networks and deep learning on a subject like recommendation systems is a given. Knowing that deep learning is beneficial in analyzing data from multiple sources and discovering hidden features, the researchers have already started to benefit from deep learning techniques in recommender systems. They have utilized deep learning techniques to produce practical solutions to the challenges of recommender systems such as scalability and sparsity.

Advantages of using Deep Learning to produce recommendations are several, like dimensionality reduction, feature extraction from different data sources. They can also be used to create or model user-item preference matrix and side information.

User-Item preference matrix: *For a given [1,2,3...n] list of users and a [1,2,3...m] list of items / features, a user-item preference matrix is defined as a matrix of n rows and m columns* ***where the value of a given a ∈ n, b ∈ m, [a,b] cell represents the interest of an a user in the b item.*** *(the interest usually represented in a natural number between 1 and 10 or 1 and 100, where the higher the number the higher the interest of the user).*

Items

Users	1	2	...	i	...	m
1	5	3		1	2	
2		2				4
:			5			
u	3	4		2	1	
:					4	
n			3	2		

*Side information: Side information is usually defined as **information that doesn't quite fit into the general observation framework, but is additional information to have "on the side"** that can be useful in modeling the recommendation system.*

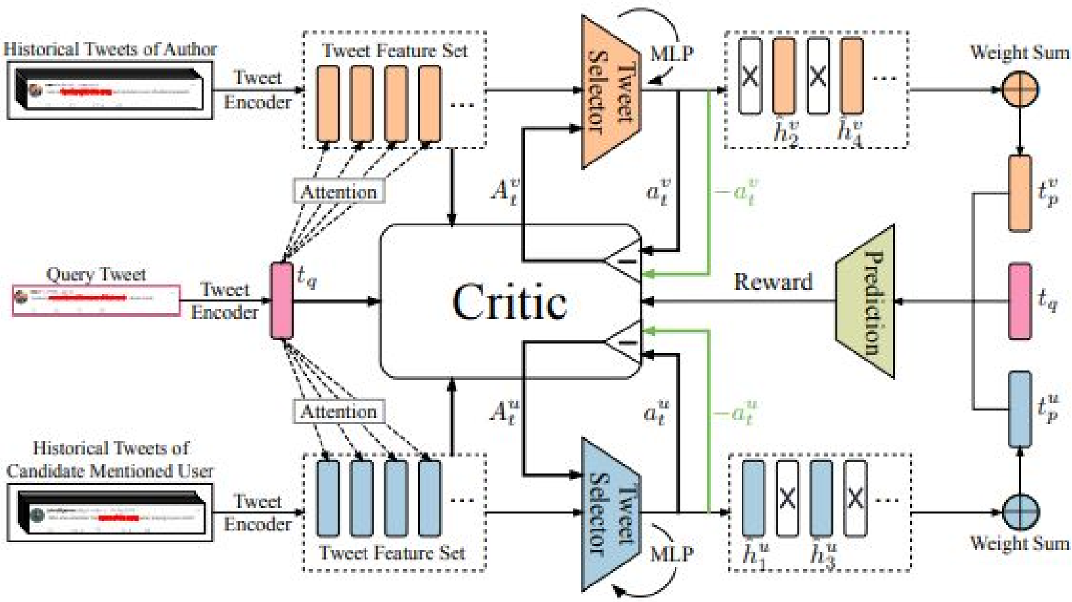
Deep learning techniques **are not specialized onto a unique recommendation method**; they are utilized in all kinds of recommendation methodologies with different purposes. In content-based filtering, these techniques are mostly used **for extracting features to generate content-based user/item profiles** (The user/item matrix defined before) from data sources. However, in CF, they are often utilized as a model-based approach to **extracting factors** on the user-item matrix (The inverse method, instead of modelling the matrix, making conclusions of such a matrix). In hybrid recommender systems, **deep learning methods are utilized for extracting features from auxiliary information and integrating them into the recommendation process** (Using side information to reinforce the model and making better predictions).

Recommendation Systems in Twitter

In Twitter the "@" symbol can be used to mention users whom the user wants to alert regarding the message.An automatic suggestion to the user of a small list of candidate names can improve communication efficiency.Previously, they used recents tweets or randomly select historical tweets to make an inference about his preferred list of names.
However, because they are too many historical tweets by users and a wide variety of content types, the use of several tweets cannot guarantee the desired results. So, it needs to extract many of the relevant indicator tweets from the historical tweets of the user and candidate mentioned users.

First tweet encoder is used to represent tweets. Second, a policy gradient is used to select relevant indicator tweets. Both the user and candidate-mentioned-user tweet selectors are policy gradient agents, whitch take the query tweet and historical tweets as inputs and determine which historical tweets should be selected.The selectors are trained by following the different gradients estimated by the critic. FInally, they merge the query-tweet representation and the features selected by agents, and use a fully connected softmax layer for prediction.

A simple but effective method is used to represent the query tweet, which consists of a bag of words where each word is one vector. A randomly initialized word embedding matrix is used to store the representations of all words. Each word is embedded in a continuous space.



At each time step t, the advantage A_t^e of selector e is given by comparing the current global reward to the reward received when that agent's action is replaced with an opposite action $-a_t^e$

Conclusion

Recommendation systems open new opportunities of retrieving personalized information on the Internet. It also helps to alleviate the problem of information overload.

The widespread use of recommendation systems and their advantages on commodity / user based services makes a really important area of current and future research. With their applications on social media still being explored, they have become a staple on services like Netflix, Spotify, Amazon or even Google ads. The introduction of deep learning and multi agent systems, although increasing both the complexity and computational power needed of these recommendation systems, opens a new universe in possibilities and there are a large number of new developing techniques and emerging models each year. With this work we hope to have given the reader the basic insights of recommender systems, a brief explanation of the uses of deep learning recommender systems and their modern applications on social media like twitter.

References

I. Recommendation systems: Principles, methods and evaluation.
II. A review on deep learning for recommender systems: challenges and remedies
III. Fairness in Recommendation Ranking through Pairwise Comparisons
IV. Recommending what video to watch next: a multitask ranking system
V. Introduction to recommendation systems and How to design Recommendation system,that resembling the Amazon
VI. Introduction to Recommender Systems in 2019
VII. Mention Recommendation in Twitter with Cooperative Multi-Agent Reinforcement Learning
VIII. Deep Learning Based Recommender System: A Survey and New Perspectives.