IT-562
Recommendation Systems and Engines

Team- 404

Article Recommendation System

An overview of the slides

Slide 4

Perhaps the most important components of any recommender system are the users. It becomes extremely essential to analyze these users. Slide 4 highlights some of the characteristics of a standard user of an article recommender system. The user has a definite taste, which does not vary on a regular basis: if someone is interested in Machine Learning, it takes time for his/her taste to change. This enables the user groups to be broadly divided into definite categories. Another important aspect is the fact that the number of actions (likes/comments/bookmark etc) per day are very few.

Slide 5

To suit the above requirements, our model should be able group similar types of items on the basis of their content, and be able to group people on the basis of their affinities. The parameters of the model should not be very volatile, but the model should work well with very few actions. Our model embodies all the above requirements, and also ensures two key components - individuality and serendipity.

Slide 6

The 6th slide explains the data. The data is collected from CI&T;s internal communication platform DeskDrop and consists articles about things like Machine Learning, Photography etc.

Slide 8

A bird eye view of the model - which is different for regular users and new users. For the new users, a model comprising of clustering and finding the most popular article on each cluster is applied. For the regular users, a hybrid model which comprises of collaborative filtering (user - user) and content based filtering is used. This model also seeks inspiration from RL. Both of these models are explained in the forthcoming slides.

Slide 9

This slide highlights the rating system adopted. With weights given to actions by the user. We did not have database for comments. Hence we were unable to decide whether the comment is positive or not. A simple assumption is taken here: If the User has liked and commented, then that interaction is rewarded otherwise we have not assigned anything to that comment.
Also an interesting thing to note about our rating system is the weights attached to actions start getting old, their respective weights start decreasing (by a factor of 8% for every month) this virtually means that an action which is more than a year old is not considered for recommendations. This is similar to the Discounted Rewards concept used in Q-learning type of Reinforcement Learning which ensures continuous and evolving learning.
The Final Rating is evaluated from the multiplication of individual weighted ratings.

Slide 10

The slides explains the hybrid model:

- 60% of the recommendations are an intersection of results from collaborative and content based: Articles which are similar to the articles liked by the user and also which are liked by people similar to him/her.
- 20% of the articles which are similar to the user preferences but do not come up in a list of the likings of items which are liked by people similar to the user. This ensures individuality. This ensures that something like photography which is liked by the user comes up even if the user belongs to a group which likes Machine Learning.
- 20% of the articles which come up in a list of the likings of items which are liked by people similar to the user but are not similar to the items liked by the user. This ensures serendipity and novelty and reduces risk factor when it comes to the same: The user is shown articles which he is not expecting, but there is a high chance high he may like them because it has been liked by his user group.

Slide 11-13

The slides explains the content based part: It is based on TF-IDF which reflects how important every word is in a given document. The weights of each of these words is multiplied with the ratings in the user vector which later helps in finding the similarity between the user preferences. This process is interactive. Every action is reflected in the user vector and this affects the finding of the similarity between the contents of the user vector and the articles. The similarity between these two is found using cosine similarity.

Slide 14

The matrix factorization is done via Singular Value Decomposition, and the missing values are found using the gradient descent algorithm, the parameters are specified.

Slide 15-16:

This slide explains how our Model responds to New Users and deals with the Cold Start Problem faced in Collaborative filtering. The system checks if the user is registered or not. If the User is new, the system assigns the user an unique ID.

The initial Recommendations shown are the most Popular Articles from each Genre. We do not have predefined genres for articles in the Data-set. Thus, we use K-means Clustering Algorithm to generate 10 cluster based on similar features. The algorithm randomly generates 10 clusters from the Data-points at the beginnings and then used Euclidean Distance to assign clusters to the Data-points. The algorithm re-calculates the clusters by calculating the centroid and thus the final clusters are achieved by convergence.

We have Pre-Processed the Articles before generating the clusters.

This popularity plus genre based recommendation is only for 5 interactions to tackle the Cold-Start Problem. After 5-interactions, our model recommends using the Hybrid Model and User Interactions.

The Snippet shows the Console application of the new user interaction.

Slide 17:

Since our Data-set had timestamps for the User Interactions and we have modelled our system to consider time-weighted ratings, we have split the data for training and testing using a reference date. The date was chosen such that the Training and Testing Data are in the Ratio 4:1.

Slides 18-19:

Recall at k is the proportion of relevant items found in the top-k recommendations.

We have used Recall-5 and Recall-10 method for testing the Accuracy of our model.

We have Compared our Hybrid Interactive Model (which uses Collaborative filtering and Content-based Filtering along with User Interactions and Time-based Weights) to pure Popularity based, Collaborative-based and Content-based approaches. The accuracy results are 52% for Recall-10 and 40% for Recall-5 which is an improvement from the normal approaches.

Slide 20:

This slide shows the past interactions of an already registered user and the recommendations generated by our Model based on that.

Slide 21:

To conclude, the model achieves a lot of features: It is based on user behaviour (described in the 4th and 5th slides). The recommendations are based on timestamp (ignores actions which are older than a year). It also solves the problem of ‘groupism’ that CF brings along with it and at the same time achieves serendipity and novelty. Another problem which this code effectively handles is that of cold start (as explained in the 15th slide). Thus we see, that this model tries to solve a lot of problems which various models bring along with them and integrates all these models to find interesting recommendation results.