Michele Damian

Evulpo ML Recommender System

Dataset

The dataset consists of ~327740 rows, enough for some machine learning application with no risk of undercutting the data. Although, many of the data is useless for a recommender system and some is missing, subject_id in primis which is essential, but I think that be derived in some way. Moreover this some of the distinguish data in specific columns is redundant in a recommender perspective since there is no need to have 4 distinguish “description” words which state that the activity has been completed, these can be distinguishable from other properties of the entry. The most useful data that can be used in this dataset are description (simplified to be binary 1 if completed and 0 if not), subject type, causer_id (only if user), properties (partially, will be discussed later) and updated_at.

State-of-the-art Recommender System

I begin with describing the two approaches broadly used also by big-tech companies like Netflix, Google, Amazon, etc, and how they would look using the given dataset.

Collaborative Filtering

The main idea behind this method is that having two subjects A and B with similar taste and/or which have completed the same exercises, probably A and B have similar taste on other new products. This approach then focuses on all the data but especially the similarities between users. To obtain precise suggestions we can use two different methods: memory based and model based (ML).

In both cases we need to factorise the data in order to have a matrix n*m with the number of users and the number of activities present in the platform. The each user will then create a binary string with 1 if they completed an activity and 0 if not.

	activity1	activity2	activity3	activity4	activity5
user1	1	0	1	0	1
user2	0	1	1	1	0
user3	0	0	1	1	0
user4	1	1	1	1	1

For the memory based method, the approach is very basic, based on similarity between users’ activity. To extract the similarity between two users’ activity one can use plenty of different methods, Cosine Similarity, Hamming Distance, and to more complex like Pearson’s Correlation, etc. Then based on the most similar user, an activity which has not been done already is picked. This is a rater simplistic approach and I do not think that suffices to deliver high quality results.

Much more interesting instead is the model based method, the approach would be very similar to a optimisation problem, where the input is the activity of the specific user and the model is trained over the whole dataset. The model would then find the users which have the most similar activity to the input and suggest some activity which the input user still have not completed but the similar users did.
In this way the dataset is simplified in order to achieve the best results and extreme efficiency. The model used to predict the suggestions can be as simple an k-nearest neighbours, but other

methods can be used based on performance. (a benchmarking can be done using K-fold cross-validation to understand which model is the best).

The advantage of collaborative filtering is the simplicity of implementation and the high level of coverage that it gives. The take-away of this approach is that it is not friendly to new activities which very few or no users have completed. Moreover if a user has completed the most activities among the similar, there may be no new activity to suggest. In this case a solution can be to add a small factor of randomisation in order to have the possibility to suggest each and every activity to each and every user. Follows an image taken from the paper *Acilar, Ayse Merve and Ahmet Arslan. “A collaborative filtering method based on artificial immune network.” Expert Syst. Appl. 36 (2009): 8324-8332.* which describes how collaborative works in high level. This paper also presents an interesting solution to the collaborative filtering problems aiNet, but the concept remains the same also without the reduced database step.

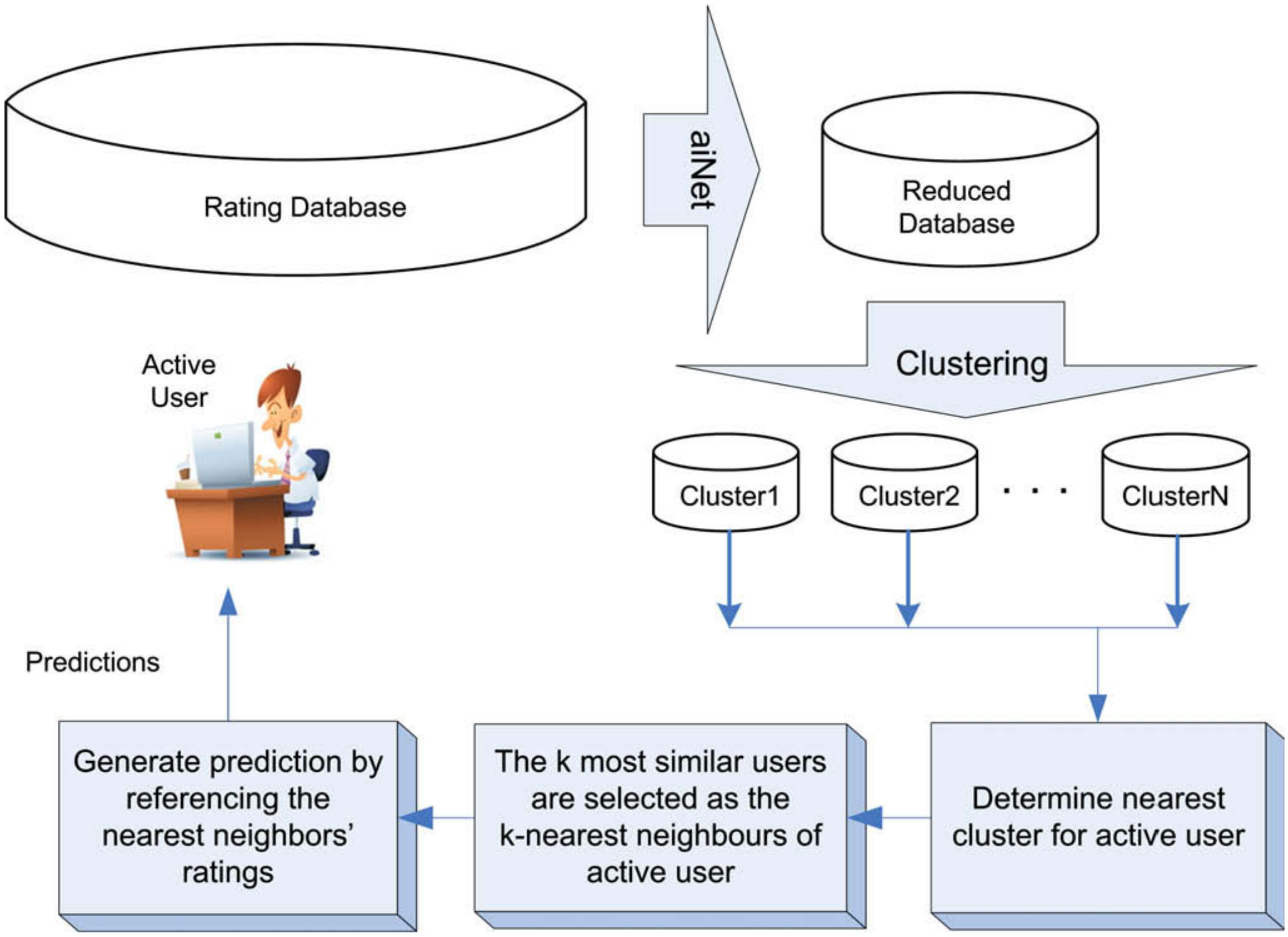

Rating Database
aiNet
Reduced Database
Clustering
Active User
Cluster1
Cluster2
. . .
ClusterN
Predictions
Generate prediction by referencing the nearest neighbors’ ratings
The k most similar users are selected as the k-nearest neighbours of active user
Determine nearest cluster for active user

Fig. 1. Block diagram of the proposed work.

Content based filtering

On the other hand, content based filtering focuses on the items similarity (in this case activities) rather than similarities between users. A suggestion would then be made by analysing the past activity of a user and understand which categories or properties are more common among them, and the suggestion would be a set of activities which share similar properties.

Solely based on the dataset given, the activities have very few properties, if not no properties excluding the unique id. In fact, the video activity only have the lesson_id and the other only present a “type” or “success rate” but nothing more to work with. With these properties, the similarity method would give very weak results, for just a small subset of the activities, while it would be impossibile to train the whole dataset.

Conclusions

As discussed above, the best approach with the given data would be to only use collaborative filtering with machine learning - model based methods, Again this can be done by manipulating the data in a very straightforward way and feed it to a model of choice (kNN for example).

Dataset Improvements

The general best approach to recommenders is utilising both the Collaborative and the content based filtering in a hybrid fashion. The base for collaborative filtering is already present, but some improvements can always be done. Firstly, this approach would greatly benefit from user similarities, this meaning that a user profile can be created (always respecting privacy). With properties like age, gender, geographical region, profession, the similarity between users can be extended and different models evaluating different similarities and suggestions based on activity or user properties can collaborate to achieve higher quality results.
Another addition (already present) can be to create time-series of events/activities completed and create a model exploiting the order in which activities are registered in the database. This would refine the suggestions and the whole system, would benefit from it.

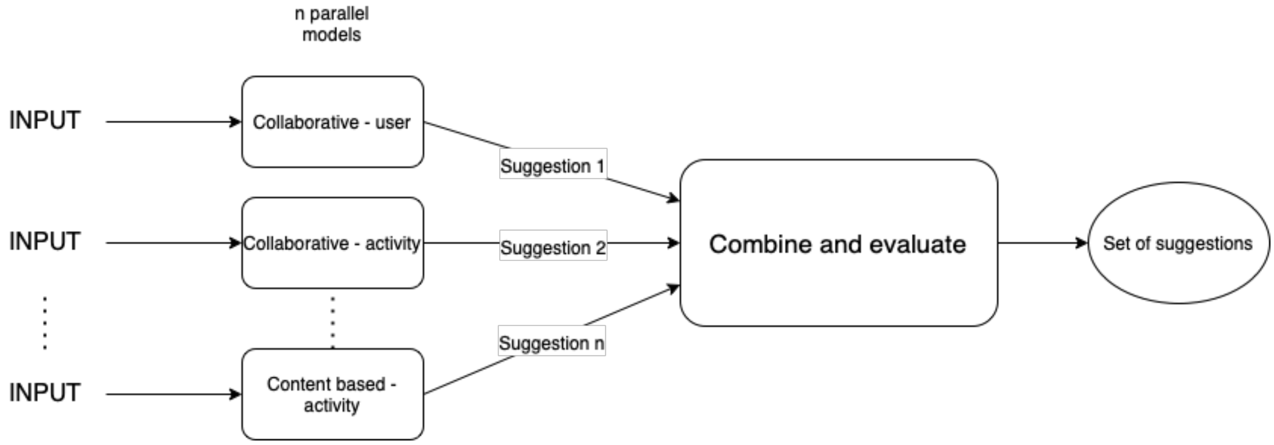
Talking about the content based filtering, with the given data it is not possibile to implement it. In general to be able to use this approach, the activities must have a set of properties that distinguish them. If these properties are present, a basic implementation of this method can be integrated by training a model similar to the collaborative filtering one which suggests an activity based on similarity of properties of past completed activities. In this case the matrix would be an n*m with the number of properties and the number of activities completed by the user. Here an example

	Maths	**Language**	**multiple choice**	**video**	**open question**	**Score**
lesson1	1	0	1	0	1	3
lesson2	0	1	0	1	0	4
lesson3	0	0	1	1	0	1
lesson4	0	0	1	0	1	5

This approach would greatly benefit from a user-rating feature. This rating can be both very basic, for example asking the user to rate a completed activity with a score between 0 and 5, but it could also be open text or a combination of both. In the case of the user directly rating the activity, the content based method would simply take these scores and use them as weights in the evaluation of the similarity model as described above.
In the case that open text is present in a rating of an activity we would need to use more AI (Natural Language Processing NLP) techniques to extract significant data from it. Specifically I think that sentiment analysis would be the key to understand how positive or negative a review is. To perform sentiment analysis there are plenty of studies and approached which each have pros and cons. With my experience I can say that lexicon-based approaches are very efficient and accurate when dealing with short text which is not too articulated, but suffers when used to extract sentiment from a complex unstructured text. In which case a machine learning approach is much more accurate and outputs better quality results, at the expense of complexity in training and running the model. By extracting then a polarity score (amount of sentiment contained in some text) the content based approach can benefit in the same way as for the user-scores, but this time with a much better accuracy.

Hybrid

The ideal scenario would be a hybrid implementation which consists of a parallel computation inside a sequential model. The parallel part would be formed by the multiple models evaluating different suggestions using the collaborative filtering alternatives and the content based alternatives. The output of all these parallel suggestions would then be fed to a sequential layer which would evaluate the predictions and output one or more recommended activity. In this way all the advantages of the different methods would be combined and used to create the best (but not the most accurate!!) prediction. By doing this then each activity would have the possibility to be recommended to each and every user. Follows a very high level graphical representation of the solution.


n parallel models
INPUT
Collaborative - user
INPUT
Collaborative - activity
INPUT
Content based - activity
Suggestion 1
Suggestion 2
Suggestion n
Combine and evaluate
Set of suggestions