Prototype Recommender System

Project Report-CS609

Spring 2021

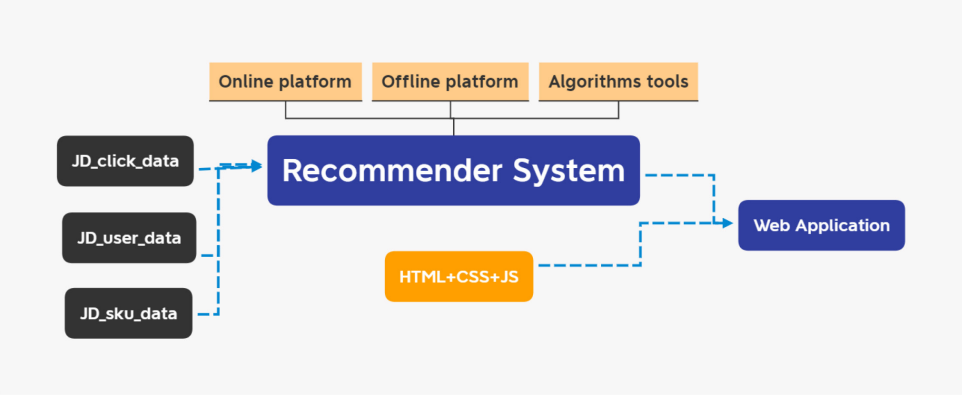
Team member(s): Wei Yang 10448858

Introduction

This project focus on the scenario that a startup company with available thirty days data on their database, and the company want to improve their customer satisfaction by deploying a new recommendation system in their website. This main part of this project is the implementation of the algorithms on recommender system, including item-based collaborative filtering, user-based collaborative filtering, demographic filtering, content-based recommendation. The goal of this project is to make a comparison between several algorithms and make further exploration on the algorithm fusion.

System architecture

There are three main parts that consisted by this recommender system: online platform, offline platform and algorithms tools. Algorithms tools is the existed python packages for recommendations. The output of offline platform is stored into mongo database. The online part will take advantage of the pre-calculated offline data and make recommendations. The effectiveness of this system is only evaluated offline. The data source will be the user data, item data and the interaction data on the left hand-side of the following graph.



System architecture

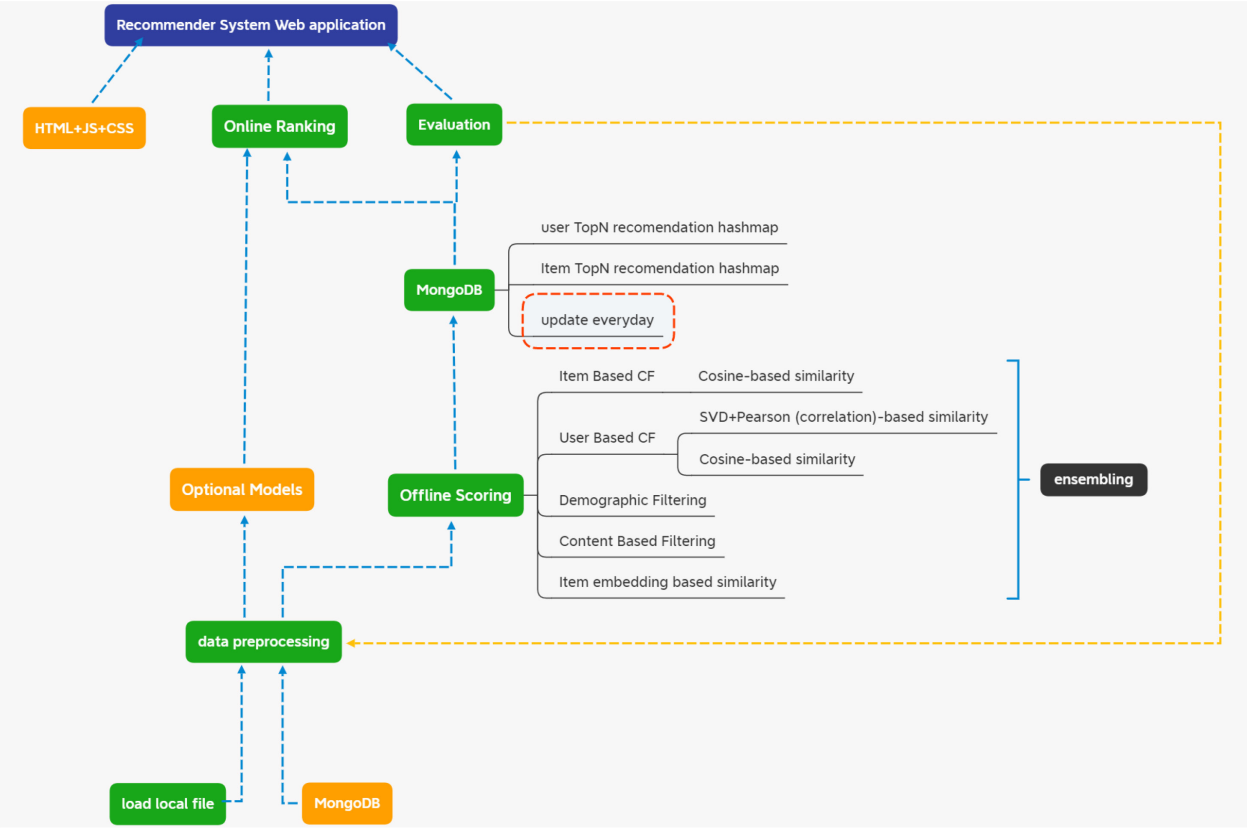
For the online platform, it mainly uses java scripts, which load the stored output from offline part, and make further computation based on these offline pre-calculated mapping. The mapping can also be considered as a model or recommendation list for each user or item.

For the offline platform, there are two main categories: user to items mapping table and item to items mapping table. For the user to items mapping table, the system will need to give a user ID, and the system will return the top 10 items as its recommendation, which is the items that user probably tend to click from

the method aspect. It is similar for the item to items case, the system only need be provided an item ID, and it will return the top 10 items as its recommendation.

Technical solutions

The following graph is the data follows of this system, which also contains technical details.



Architecture detail

This system uses click interaction as primary target to extract information. Joined the tables to get the desired table shown as follows, which contains SKU ID, user ID. Each row is a click by the user. (**Preprocess**) It removed click records with the same user ID and request time since users may click several times on their screen, but the effective records in there are only once. For privacy protection, the SKU ID and user ID are encoded, which make the result of the system not that intuitive.

	sku_ID	user_ID	request_time	channel	type	brand_ID	attribute1	attribute2	user_level	first_order_month	plus	gender	age	marital_status	education	city_level	purchase_power
0	ab5f54528b	026f999613	2018-03-12 01:30:56	app	2	0def762aeb	4.0	100.0	4	2012-06	1	M	16-25	S	4	1	2
1	d10b8d2fba	6093ba9a56	2018-03-12 10:49:57	app	2	9b0d3a5fc6	-	-	2	2017-06	0	F	26-35	S	3	4	3
2	d10b8d2fba	6093ba9a56	2018-03-12 14:22:16	app	2	9b0d3a5fc6	-	-	2	2017-06	0	F	26-35	S	3	4	3
3	aefceb7422	6093ba9a56	2018-03-12 14:25:41	app	2	9b0d3a5fc6	3.0	60.0	2	2017-06	0	F	26-35	S	3	4	3
4	d10b8d2fba	6e8f517d7b	2018-03-12 14:34:52	app	2	9b0d3a5fc6	-	-	1	2016-11	0	F	36-45	M	-1	1	-1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

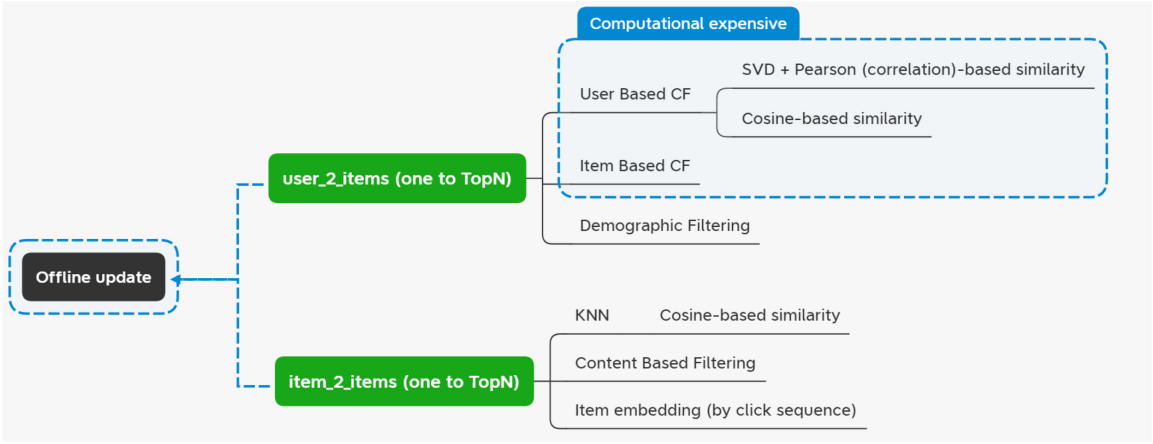
For collaborative filtering, the offline platform of this system retrieved accumulated click counts for each user by SKU ID to get a user item matrix first, which is kind of panel data for user ID and SKU ID. For simulation the true environment, consider that it only has 15 days data, and the system will update day by day with the time moving forward until the end of this dataset. The user item matrix is accumulated from beginning since users probably have more inclination to buy the item if they click more times of that item. With this matrix, this system implemented the user based collaborative filtering and item based collaborative filtering with the Pearson and cosine similarity measurements.

For content-based recommendation, the algorithm get the grouping table for item attributes first, then it finds the 10 highest click frequency items in each group, and finally join the table back with item attributes. In other words, the algorithm group items based on their attributes first, and find what is the most popular ten items in each group. If a user clicks an item within one group, the system will just recommend the most popular items in that group, on the recent 7 days. Obviously, the more attributes the items had, the more accuracy this method will be.

For demographic based filtering, it is almost the same as previous one. The only difference is that we need to grouping table by user attributes first.

For item embedding, the algorithm first generate user click sequence, and treat these SKU ID sequence as vocabulary in NLP concept, then get the embedding for each SKU ID by training a skip-gram model with genism package in python.

The ranking part of this system is computed based on the scores on the scoring part, rather than ranking neural networks. The summary is shown as graph as follows.



Offline platform

Related software: MongoDB, Node.Js, VScode, python and python packages (numpy, pandas, sklearn, genism)

Experiments

The dataset only contains 30 successive days records, which contains 31868 unique SKU ID, 457298 unique user ID and 20214515 click records. The interaction data is about 1 GB, but when loading into memory and computing the correlations, the RAM will be consumed really fast. To speed up experiment, this project only randomly choosing about 10% users and items as target first. As show on the technical solutions part, the SKU table contains two attributes, which are encoded. The user table contains 9 attributes, which are listed as follows.

1	user_ID	user_level	first_order_month	plus	gender	age	marital_status	education	city_level	purchase_power
2	000089d6	1	2017-08	0	F	26-35	S	3	4	3
3	0000babd	1	2018-03	0	U	U	U	-1	-1	-1
4	0000bc01	3	2016-06	0	F	>=56	M	3	2	3
5	0000d0e5	3	2014-06	0	M	26-35	M	3	2	2
6	0000dce4	3	2012-08	1	U	U	U	-1	-1	-1

User table attributes

For environment, this experiment implemented on windows system, which has 32 GB RAM and AMD 2700X as CPU.

For evaluation, A/B test is obviously not available in this dataset on this period since there is no influenced user-item interaction generated by this system. Therefore, this project only evaluated on daily scale based on accuracy. For item to items method, given previous click SKU ID, the system will predict next click of SKU ID. For user to items method, given user ID, the system will predict next click of SKU ID. And the following table make the summary and comparison between different algorithms.

Rec_num is the parameter that decides how many SKU that will recommend to the user. Basically, the more recommended number of items, the higher accuracy the algorithm will have. Because it will cover more possible item that user probably click in next time.

Comparison result on 2018-03-15:

1		suc_ratio_on_2018-03-15	suc_ratio_on_2018-03-15	suc_ratio_on_2018-03-15	suc_ratio_on_2018-03-15	suc_ratio_on_2018-03-15	suc_ratio_on_2018-03-15
2	rec_num = 1	0.134	0.752	0.018	0.506	0.476	0.001
3	rec_num = 2	0.172	0.777	0.041	0.605	0.635	0.010
4	rec_num = 3	0.193	0.797	0.066	0.661	0.701	0.016
5	rec_num = 4	0.203	0.800	0.079	0.676	0.757	0.016
6	rec_num = 5	0.208	0.811	0.090	0.692	0.802	0.016
7	rec_num = 6	0.217	0.817	0.110	0.714	0.820	0.019
8	rec_num = 7	0.221	0.824	0.119	0.732	0.843	0.019
9	rec_num = 8	0.223	0.830	0.131	0.737	0.858	0.020
10	rec_num = 9	0.225	0.837	0.133	0.749	0.876	0.020
11	total_click_samples	2010	2010	2010	2010	2010	2010
12	method	user based recommendation && user to item	item based recommendation && item to item	item based recommendation && user to item	contents based recommendation && item to item	demographic based recommendation && user to item	w2v based recommendation && item to item

Comparison result on 2018-03-16:

		suc_ratio_on_2018-03-16	suc_ratio_on_2018-03-16	suc_ratio_on_2018-03-16	suc_ratio_on_2018-03-16	suc_ratio_on_2018-03-16	suc_ratio_on_2018-03-16
2	rec_num = 1	0.133	0.772	0.025	0.517	0.465	0.001
3	rec_num = 2	0.161	0.788	0.054	0.594	0.619	0.007
4	rec_num = 3	0.173	0.806	0.066	0.666	0.718	0.010
5	rec_num = 4	0.176	0.808	0.074	0.690	0.786	0.010
6	rec_num = 5	0.178	0.813	0.083	0.708	0.813	0.010
7	rec_num = 6	0.182	0.815	0.093	0.716	0.847	0.011
8	rec_num = 7	0.184	0.828	0.094	0.722	0.875	0.013
9	rec_num = 8	0.186	0.831	0.096	0.746	0.891	0.013
10	rec_num = 9	0.187	0.841	0.100	0.748	0.901	0.013
11	total_click_samples	1725	1725	1725	1725	1725	1725
12	method	user based recommendation && user to item	item based recommendation && item to item	item based recommendation && user to item	contents based recommendation && item to item	demographic based recommendation && user to item	w2v based recommendation && item to item

Comparison result on 2018-03-17:

		suc_ratio_on_2018-03-17	suc_ratio_on_2018-03-17	suc_ratio_on_2018-03-17	suc_ratio_on_2018-03-17	suc_ratio_on_2018-03-17	suc_ratio_on_2018-03-17
2	rec_num = 1	0.142	0.782	0.027	0.526	0.506	0.003
3	rec_num = 2	0.178	0.797	0.039	0.619	0.641	0.006
4	rec_num = 3	0.186	0.809	0.046	0.661	0.730	0.009
5	rec_num = 4	0.191	0.810	0.053	0.690	0.775	0.009
6	rec_num = 5	0.201	0.819	0.071	0.693	0.821	0.009
7	rec_num = 6	0.203	0.827	0.089	0.722	0.847	0.011
8	rec_num = 7	0.205	0.834	0.106	0.727	0.869	0.011
9	rec_num = 8	0.206	0.835	0.109	0.732	0.893	0.011
10	rec_num = 9	0.211	0.847	0.112	0.742	0.910	0.011
11	total_click_samples	1598	1598	1598	1598	1598	1598
12	method	user based recommendation && user to item	item based recommendation && item to item	item based recommendation && user to item	contents based recommendation && item to item	demographic based recommendation && user to item	w2v based recommendation && item to item

Comparison result on 2018-03-18:

		suc_ratio_on_2018-03-18	suc_ratio_on_2018-03-18	suc_ratio_on_2018-03-18	suc_ratio_on_2018-03-18	suc_ratio_on_2018-03-18	suc_ratio_on_2018-03-18
2	rec_num = 1	0.119	0.790	0.014	0.595	0.508	0.001
3	rec_num = 2	0.147	0.795	0.027	0.682	0.627	0.002
4	rec_num = 3	0.157	0.804	0.047	0.720	0.713	0.003
5	rec_num = 4	0.170	0.806	0.065	0.748	0.761	0.003
6	rec_num = 5	0.172	0.812	0.083	0.757	0.790	0.003
7	rec_num = 6	0.176	0.817	0.087	0.762	0.815	0.004
8	rec_num = 7	0.178	0.820	0.099	0.767	0.850	0.004
9	rec_num = 8	0.180	0.826	0.103	0.778	0.870	0.005
10	rec_num = 9	0.181	0.832	0.107	0.785	0.882	0.005
11	total_click_samples	1316	1316	1316	1316	1316	1316
12	method	user based recommendation && user to item	item based recommendation && item to item	item based recommendation && user to item	contents based recommendation && item to item	demographic based recommendation && user to item	w2v based recommendation && item to item

Comparison result on 2018-03-19:

		suc_ratio_on_2018-03-19	suc_ratio_on_2018-03-19	suc_ratio_on_2018-03-19	suc_ratio_on_2018-03-19	suc_ratio_on_2018-03-19	suc_ratio_on_2018-03-19
2	rec_num = 1	0.116	0.775	0.013	0.539	0.436	0.004
3	rec_num = 2	0.144	0.794	0.023	0.621	0.644	0.007
4	rec_num = 3	0.148	0.802	0.028	0.675	0.719	0.008
5	rec_num = 4	0.152	0.806	0.041	0.702	0.788	0.010
6	rec_num = 5	0.153	0.809	0.046	0.708	0.821	0.012
7	rec_num = 6	0.156	0.824	0.049	0.725	0.856	0.014
8	rec_num = 7	0.157	0.828	0.057	0.727	0.875	0.015
9	rec_num = 8	0.157	0.830	0.064	0.738	0.885	0.015
10	rec_num = 9	0.158	0.832	0.068	0.754	0.893	0.018
11	total_click_samples	1060	1060	1060	1060	1060	1060
12	method	user based recommendation && user to item	item based recommendation && item to item	item based recommendation && user to item	contents based recommendation && item to item	demographic based recommendation && user to item	w2v based recommendation && item to item

Conclusion

From comparison on several days, the item-based recommendation (which directly calculate the cosine similarity between different items based on user item matrix) and the demographic based recommendation are more effective than other methods in there.

The online model fusion part should be only written in java scripts form, which is not implemented in this period. Also, the evaluation method such as A/B test for this dynamic online fusion method is needed.

In the future, this project will be tested on different dataset that without encoding the SKU and user ID, which may have more intuitive recommendation result. Furthermore, with more computational resource, this system should be run on the large scale with the prepared parameters in functions. On ranking model aspect, to further improve the accuracy, this project probably extends more neural network models. Of course, the speed of the response will need to be considered.

Except for the original dataset, the project code and related files are keep on updating on github: https://github.com/InscribeDeeper/RS_code .

Code execution window: