IA 316: Recommender Systems

Nicolas Scotto Di Perto and Valentin Charvet

Télécom ParisTech

February 28, 2019

Presentation Of The Environment

The objective of this project is to model the behavior of buyers in a simulated retail environment:

- a set of users $\mathcal{U}$
- a set of items $\mathcal{I}$
- at each time step t, a user buys an item among a set s_t

We want to learn a function $\hat{f} : (\mathcal{U}, \mathcal{I}^k) \rightarrow \mathcal{I}$

Initialization Of The Environment

When resetting the environment, we retrieve a history of transactions, each one of them is modeled by

- a user
- a list of available items
- an action, which corresponds to the recommended item
- the reward from this action, it values
    - the price of the item if the user bought the one that was recommended
    - 0 else (the recommendation was incorrect)

Running the environment

We make the recommendations iteratively, at each time step t:

- we receive a state s_t that corresponds to a user and a set of available items
- we recommend the item output by $\hat{f}(s_t)$
- we receive a feedback (the reward) that indicates weather the recommendation was correct
- *optional* retrain the model on-line using the feedback

Data Format

- a state s_t is represented by an array of k substates, each of them contains
  - item id
  - price
  - 2 item-specific variables
  - 1 meta data
- each of these substates is given for one user and with two user-specific variables

Implicit Feedback

- Contrary to the first and second environment, we don't have access to an explicit feedback from the users
- However, we know can extract implicit information from the purchases history:
  - if the reward from an action is non zero, we can tag the bought article as "positive", since the user liked it enough to buy it
  - if the reward is 0, we tag the item as "negative" since the user preferred another one despite the recommendation
- Therefore, we can use an inference model using this pseudo explicit feedback
- the rate of good recommendations in the history is $29.6 \pm 1.3\%$

Cold Start Issue

- During the runs, we frequently encounter users that did not appear in the history dataset (usually a dozen)
- Therefore, we need to use the users metadata to be able to make predictions for those new users
  - For a new user u_{new}, we look for the most similar to him in $\mathcal{U}$ using cosine similarity with users variables
  - *First Method:* we recommend the same item we would have done to the most similar user
  - *Second Method:* we use the embedding of the most similar user to make the prediction for the new one

User Based Recommendation

- For this model, we only retrieve the positive recommendations from the history
- at each time step, we receive $u \in \mathcal{U}$ and $i_1 \ldots i_k \in \mathcal{I}^k$
- we compute the cosine similarity between u and the users in the history that have bought an item from $i_1 \ldots i_k$
- we recommend the most expensive item from those having the highest similarities
- It enables solving the cold start problem easily

Implicit Feedback + User Based

- As said previously, the implicit feedback model fails at recommending items for new users as it embeds unseen values
- at time step t if we encounter a unseen user, we look for the most similar one
- we use the embedding from the similar (already seen) user as input of the neural network

Evaluation Metrics

To assess the performance of our model, we need to define a metric

- A naive way to do this is to measure the rate of correct predictions ie count the number of positive rewards during a run
- However, from a business point of view, we want to maximize the mean reward $\bar{R}$. If $\bar{R} > \frac{1}{30} \times \underset{i \in \mathcal{I}}{max}\{price(i)\}$(reward if we always recommend the most expensive item) then our model performed well

Experimental Results

recommendation	max_price	mean_price	average_reward	good_reco_ratio	average_reward_normalized_max	average_reward_normalized_mean
Null special, en ligne	948.61	71.0337	159.974	0.348	0.169103	2.21164
Fully Implicit Null special, en ligne	975.326	70.9933	190.918	0.336	0.195501	2.67344
Fully Implicit Null binaire, en ligne	938.306	71.5375	190.431	0.334	0.203	2.63528
Fully Implicit Null, en ligne (batch)	967.177	70.9395	178.746	0.326	0.184934	2.54073
Null, hors ligne	988.98	67.9246	106.693	0.326	0.107767	1.5315
Fully Implicit Null, hors ligne	981.438	71.2866	170.767	0.324	0.17439	2.38399
Null binaire, en ligne (batch)	966.931	69.1898	107.825	0.32	0.111365	1.56555
Null binaire, hors ligne	970.499	70.1291	191.767	0.32	0.196501	2.67836
Null, en ligne	987.082	71.7005	157.42	0.312	0.159324	2.19686
Fully Implicit Null binaire, en ligne (batch)	977.08	75.11	184.572	0.304	0.190082	2.45169
Fully Implicit Null binaire, hors ligne	955.954	66.0836	171.664	0.304	0.179991	2.59192
Simple, hors ligne	975.053	70.1065	187.343	0.3	0.192437	2.71928
Null special, hors ligne	927.328	67.907	178.528	0.298	0.194693	2.63961
Fully Implicit Simple, en ligne	964.053	70.315	186.099	0.298	0.193112	2.65159
Null special, en ligne (batch)	978.596	73.9991	184.464	0.296	0.18828	2.41235
Null, en ligne (batch)	987.111	71.7079	179.49	0.296	0.181767	2.44749

Figure: Results with our different metrics for several models