A Distributed and Accountable Approach to Offline Recommender Systems Evaluation

Diego Monti[a], Giuseppe Rizzo[b] and Maurizio Morisio[a]

[a]Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy
[b]Istituto Superiore Mario Boella (ISMB), Via Pier Carlo Boggio 61, 10138 Turin, Italy

Introduction

Different authors have empirically demonstrated that offline evaluation protocols in the context of recommender systems have several weaknesses [1]. Nevertheless, offline experiments are extremely important for comparing a large number of candidate algorithms without sustaining the costs of an online evaluation involving too many human subjects [2].

We propose a way of overcoming the problem of comparing offline evaluation results obtained from different recommendation algorithms in heterogeneous settings. To this end, we implemented an open source evaluation framework for top-k prediction methods, called RecLab, that is capable of interacting with several recommenders using RESTful APIs.

Thanks to this approach, it is possible to compare, in a controlled environment, different algorithms and techniques without necessary disclosing their implementation details. The results of each experiment, along with the respective configuration parameters, are publicly available to support accountability.

The code is freely available in a GitHub repository at `https://github.com/D2KLab/reclab`.

Interaction Protocol

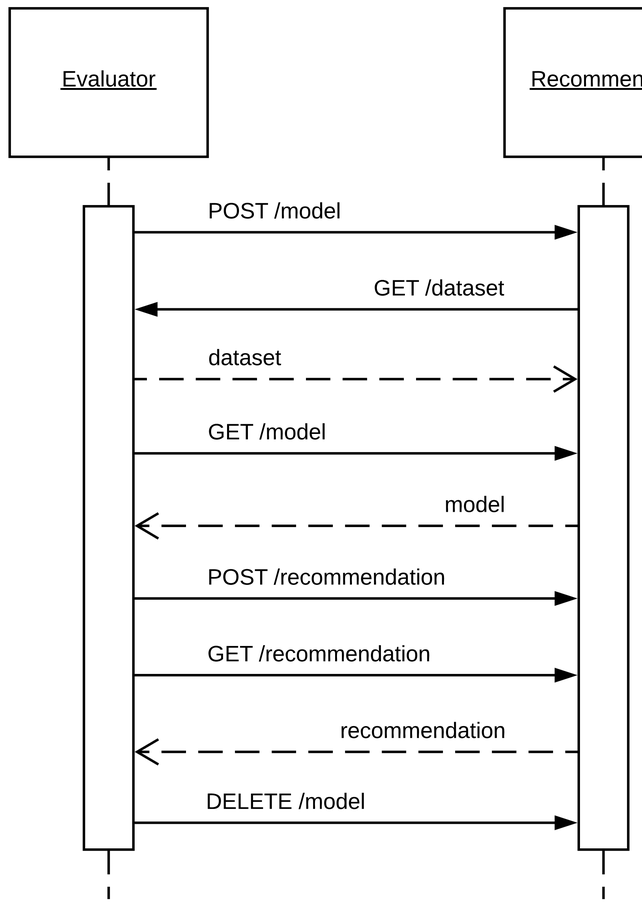

Figure 1: Simplified UML sequence diagram

Evaluation Framework

The experimenter needs to specify the following parameters before starting an evaluation:

- the initial rating dataset;
- the technique used to split the dataset;
- the size of the training and the test set;
- the length k of the lists of recommended items;
- the threshold between negative and positive ratings;
- the list of recommenders to be evaluated.

For demonstrative purposes, we included in RecLab a set of recommender systems that follow the interaction protocol illustrated in Figure 1. However, anyone is encouraged to implement other techniques for the purpose of evaluating them with this framework. Further recommenders can be added by simply inserting their URIs in a configuration file present in our repository. All available recommenders are then displayed to the experimenter.

Experimental Results

In Table 1 we report the results of a first experiment where we selected the MovieLens 1M dataset [3] and we chose a random splitting protocol. For the other parameters, we used the default values of the framework.

In a second experiment, we only changed the splitting protocol to the timestamp-based one, while we retained all other parameters unmodified. The results are reported in Table 2.

Algorithm	Coverage	Precision	Recall	NDCG	Novelty	Diversity	Serendipity
Random	1.000000	0.005152	0.002526	0.005069	13.37526	0.966485	0.005003
Most Popular	0.017920	0.145146	0.084294	0.158512	8.580345	0.600524	0.071869
Item KNN	0.473527	0.212028	0.137608	0.224146	10.55504	0.788987	0.196637
User KNN	0.141732	0.263337	0.189679	0.295034	9.052157	0.657436	0.205550
BPRMF	0.326907	0.225464	0.148515	0.247625	9.473122	0.717023	0.176972
WRMF	0.120554	0.258400	0.169925	0.287808	9.138422	0.667673	0.210835

Table 1: Evaluation results with the MovieLens 1M dataset and a random splitting

Algorithm	Coverage	Precision	Recall	NDCG	Novelty	Diversity	Serendipity
Random	0.993719	0.017555	0.002910	0.017394	13.41860	0.963699	0.016938
Most Popular	0.037411	0.257487	0.053148	0.273653	8.546251	0.528352	0.066517
Item KNN	0.344894	0.231296	0.056491	0.244158	9.759138	0.670575	0.095962
User KNN	0.117422	0.275042	0.062094	0.293720	8.842892	0.567265	0.114470
BPRMF	0.220918	0.265339	0.062845	0.282460	9.083545	0.611382	0.112675
WRMF	0.136537	0.276164	0.065600	0.297178	8.941997	0.587175	0.121929

Table 2: Evaluation results with the MovieLens 1M dataset and the timestamp splitting

Metrics

In order to better analyze the recommender systems under evaluation from different perspectives, we decided to include in RecLab a comprehensive set of seven different metrics (Table 1 and 2). In fact, it is not possible to accurately evaluate in an offline experiment a set of recommenders by only relying on a single indicator [4]. We selected not only traditional metrics like coverage and precision but also less widespread ones like novelty, diversity, and serendipity.

Live Demonstration

It is possible to design and run an experiment at `http://datascience.ismb.it/reclab/`.

Conclusion and Future Work

We have introduced RecLab, an open source framework for evaluating top-k recommender systems in a distributed setting. The main aim of this work is to support the accountability and the reproducibility of the results of the experiments by permanently storing and publicly displaying their configuration parameters and numerical outcomes.

As future works, we plan to integrate more rating datasets and other recommendation techniques. We also envision the possibility of enhancing the interaction protocol in order to let the experimenter specify the configuration parameters of each recommender.

References

[1] Alan Said and Alejandro Bellogín.
Comparative recommender system evaluation.
In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 129–136. ACM Press, 2014.

[2] Asela Gunawardana and Guy Shani.
Evaluating recommender systems.
In *Recommender Systems Handbook*, chapter 8, pages 265–308. Springer US, 2 edition, 2015.

[3] F. Maxwell Harper and Joseph A. Konstan.
The MovieLens datasets: History and context.
ACM Transactions on Interactive Intelligent Systems, 5(4):1–19, 2015.

[4] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl.
Evaluating collaborative filtering recommender systems.
ACM Transactions on Information Systems, 22(1):5–53, 2004.