Recommenders Benchmark Framework

Aviram Dayan
Ben-Gurion University
Beer-Sheva 80145
Israel
avdayan@cs.bgu.ac.il

Guy Katz
Ben-Gurion University
Beer-Sheva 80145
Israel
rakreshet@gmail.com

Naseem Biasdi
Ben-Gurion University
Beer-Sheva 80145
Israel
naseem@cs.bgu.ac.il

Lior Rokach
Ben-Gurion University
Beer-Sheva 80145
Israel
liorrk@bgu.ac.il

Bracha Shapira
Ben-Gurion University
Beer-Sheva 80145
Israel
bshapira@bgu.ac.il

Roland Schwaiger
Deutsche Telekom AG
Berlin 10781
Germany
roland.schwaiger@telekom.de

Aykan Aydin
Deutsche Telekom AG
Berlin 64295
Germany
aykan.aydin@telekom.de

Radmila Fishel
Ben-Gurion University
Beer-Sheva 80145
Israel
fishelr@bgu.ac.il

ABSTRACT

In this demo we present a recommender benchmark framework that serves as an infrastructure for comparing and examining the performance and feasibility of different recommender algorithms on various datasets with a variety of measures. The extendable infrastructure aims to provide easy plugging of novel recommendation-algorithms, datasets and compare their performance using visual tools and metrics with other algorithms in the benchmark. It also aims at generating a WEKA-type workbench [1] for the recommender systems field to enable usage and application of common recommender systems (RS) algorithms for research and practice. The demo movie is available at : **http://www.youtube.com/watch?v=fsDITf6s0WY**

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *Information filtering, Selection process*.

General Terms

Algorithms, Measurement, Performance, Experimentation.

Keywords

Benchmark, recommendation engines, datasets, evaluators, metrics, dataset characteristics.

1. INTRODUCTION

Providing quality recommendations for various domains and datasets is challenging many academic studies and commercial applications that aim at suggesting the user the most suited items for her needs, context and preferences. An application designer who wishes to add a recommendation engine to an application should choose from a large variety of available algorithms or examine the feasibility and performance of a newly developed algorithm for the application at hand.

2. SYSTEM DESCRIPTION

Our benchmark may serve two goals: 1. a decision support tool for selection the best suitable recommender engine for a certain problem and data. The framework provides a recommendation-system administrators the tools for evaluating multiple recommenders with different datasets, metrics, and benchmark-policies to assist with her decision; 2. An infrastructure to perform research related to RS. The benchmark currently includes several RS techniques, numerous datasets and evaluation metrics, so that algorithms can be evaluated and compared with any suit able datasets and metrics. We also enable easy plug-in of new algorithms. At the moment the benchmark includes for example: collaborative filtering SVD, Ensemble of CF, Community-based recommender and cross-domain algorithms. As a test–case for the feasibility and ease of plugging new algorithms and datasets, a class of graduate students from BGU University implemented and plugged 10 new algorithms and numerous datasets and performed evaluation comparing the algorithms, among the algorithms that they plugged in are clustering-based recommender, ontology-based, transfer learning for cross-domain and others.

The framework maintains the following features:

1. Support of a variety of RS techniques (e.g., CF, Social, etc.)
2. Support of a variety of evaluation measures
3. Support of off-line and on-line evaluations protocols
4. Including built-in statistical procedures and reporting tools
5. Easy to plug new algorithms, datasets and measures using API
6. User-friendly

When using the benchmark (Figure 1), the user can choose the datasets; the algorithms; evaluators and metrics (Figure 2). Then, the user can explore the benchmark results using several views (Figure 3) to deduce conclusions. He can also obtain analysis of the datasets and manipulate its features (e.g., sparsity).

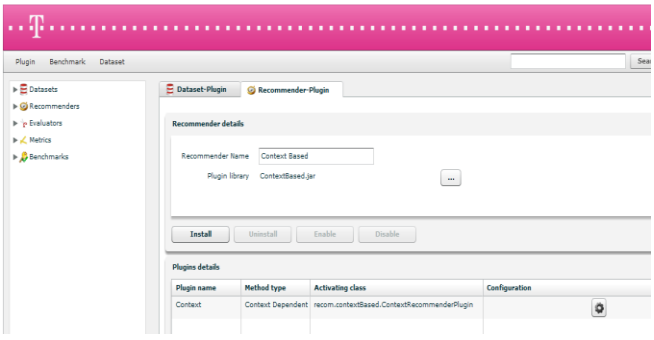

Figure 1. Introducing a recommender/dataset to the framework

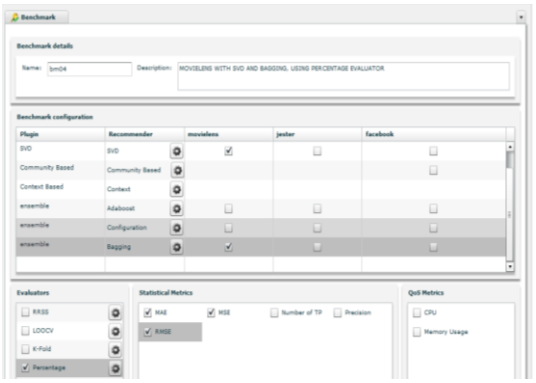

Figure 2. Benchmark configuration

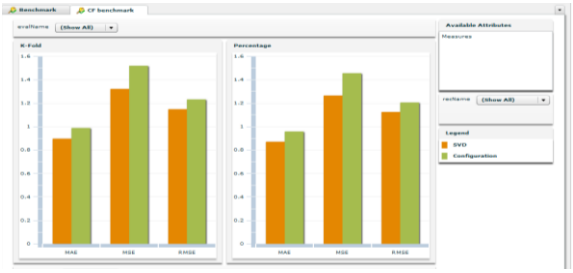

Figure 2. Example of Benchmark results

3. SYSTEM ARCHITECTURE

The benchmark framework architecture (Figure 4) follows the multi-tier paradigm. To allow maximum flexibility, recommendation algorithms and dataset are implemented as plug-ins to the framework. These plug-ins can be added and removed to and from the framework in runtime which allows development and integration with no dependency of the framework itself. Another key feature is the complete decoupling of the recommendation algorithms plug-ins and the dataset plug-ins implementation. An algorithm can use the dataset as input as long as the data contained in this dataset is appropriate to the algorithm (for example, community-based recommendation algorithm needs community data).

The framework contains four core components:

1. The recommendations framework – contains the APIs' for recommender algorithms and data-sets.
2. The benchmark framework – contains the APIs' for benchmarking as well as the notions of Evaluators and Metrics.
3. The plug-in manager – contains the APIs' for wrapping recommenders and data-sets as plug-ins.
4. Recommenders and data-sets plug-ins –are developed separately from the framework and are following the plug-in APIs'. They are plugged-in to the system using the plug-in manager.

Recommender and dataset plug-ins are completely decoupled. The benchmark framework allows using a recommender with a dataset according to the used recommendation method and the content type of the data in the dataset. For example: collaborative filtering recommenders can be benchmarked only with datasets that contain rating data.

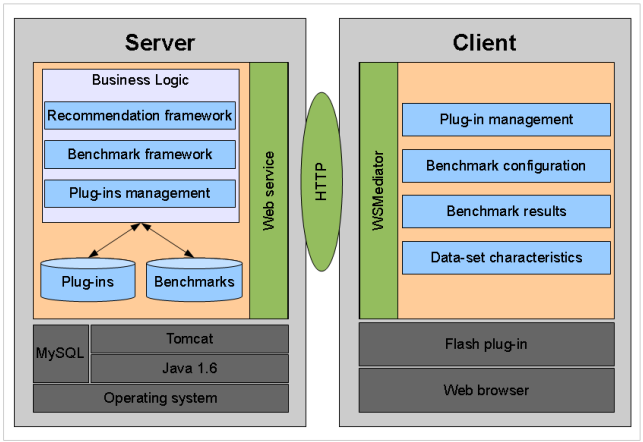

Figure 4: System Architecture

4. RELATED WORK

Two related framework should be mentioned, the WEKA [1] framework for machine learning is a well established workbench for data mining tasks such as clustering and classification. Although many recommender systems algorithms are based on machine learning methods, WEKA is not suitable for evaluating RS algorithms as it does not support basic RS algorithms such as CF and not the basic data models such as rating matrices. Mahout[2] is a machine learning library that includes collaborative filtering and SVD algorithms for recommendations. It is a library of scalable modules and is currently limited to the above mentioned algorithms. Its library modules are to be used when implementing recommender systems for production. Our benchmark is aimed at a much earlier stage of selecting a suitable recommender system for the problem at hand comparing candidate methods rather than the actual construction of the system.

5. REFERENCES

[1] WEKA workbench -**http://www.cs.waikato.ac.nz/ml/weka/**

[2] Sean Owen, Robin Anil, Ted Dunning, and Ellen Friedman. Mahout in Action. Manning Publications, 2010. published under Manning Early Access Program.

DEMO DESCRIPTION

The demo enables evaluation and comparison of recommender algorithms on various datasets. It enables easy plug-in of new algorithms and new datasets, configuration of the datasets features and the condition of the evaluation, and presents graphical results of the evaluation.

Link to demo movie:

http://www.youtube.com/watch?v=fsDITf6s0WY

Hardware requirements

For a small scale demonstration any standard machine will do. For a full-blown demonstration: 2 x Intel Xeon 4C Processor, 24GB RAM and at least 1TB of storage.

Software requirements

Java 1.6.x, MySQL 5.1, Tomcat 6.0.2x.

Network Access requirements

Port 8080 should be open for incoming HTTP connections. Any other port can be used for that matter.

The presenter Aviram Dayan is part of a team of researcher and scientific programmers at Ben-Gurion University that is working on various projects related to recommender systems together with researchers from Deutsche Telekom Laboratories (TLabs) at Berlin.