WrapRec: An Easy Extension of Recommender System Libraries

Babak Loni, Alan Said
Multimedia Computing Group
Delft University of Technology, Netherlands
{b.loni, a.said}@tudelft.nl

ABSTRACT

WrapRec is an easy-to-use Recommender Systems toolkit which allows users to easily implement or wrap recommendation algorithms from other frameworks. The main goals of WrapRec are to provide a flexible I/O, evaluation mechanism and code reusability. WrapRec provides a rich data model which makes it easy to implement algorithms for different recommender system problems, such as context-aware and cross-domain recommendation. The toolkit is written in C# and the source code is publicly available on Github under the GPL license.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

Keywords

Recommender Systems, Open Source Library

1. INTRODUCTION

Personalized recommender systems are becoming very popular in online marketing, social networks and mobile applications. RecSys and machine learning communities have developed several successful libraries for recommender systems such as *MyMediaLite*[1], *Apache Mahout*[2] and *LensKit*[3].

Most of existing libraries require specific data formats, usually as text files, and are not flexible enough to support new data formats. Data processing is an important step in recommender systems which are usually neglected in current RecSys libraries. The existing frameworks usually require the users to provide the data in a specific format, and similarly they output results in a specific format. As an example consider that a user has a data source with a particular format which is not supported by the existing toolkits. A common practice to use this data is to convert to a format which is supported by the toolkit and then run the experiment on the converted data. Now, if the user wants to repeat the experiment for many different scenarios, usually the data preprocessing is done by an external application in a semi-manual way which makes the usage of toolkits difficult and error prone.

WrapRec is a toolkit, with defined routines. It allows users to incorporate any data processing steps easily into the experiment, without requiring to change the underlying algorithms. The toolkit is a wrapper around existing libraries, which allows them to be plugged into the system. The main goal of the toolkit is to provide high-level interfaces for low-level services to make it easier to run RecSys algorithms. WrapRec's solution to this goal is to provide a rich data object model to make data access flexible and safe. This feature makes it easier to implement algorithms which rely on multiple data sources such as context-aware and cross-domain recommendation scenarios. Furthermore, WrapRec provides high level interfaces to potential algorithms that can be wrapped into the toolkit. Similarly an abstraction layer for evaluation of algorithms is defined in the toolkit to implement custom evaluators and re-use evaluation logics for different RecSys algorithms.

2. OVERVIEW OF THE TOOLKIT

The WrapRec toolkit consists of three main components: Data Layer, Recommendation Engine and Evaluation Pipeline. The components are made independent of each other, meaning they can be modified and extended without requiring to change other components. Figure 1 illustrates the high level architecture of the toolkit. Below, we briefly describe each component:

- **Data Layer**: This component provides a common data interface, regardless of the underlying format of the data source. If users want to read a custom data format, they need to implement an interface which reads the data and converts it into the common data objects. This feature makes it possible to run different experiments without requiring to change the underlying algorithms or evaluation mechanism. Furthermore, the toolkit is able to represent the data objects within a context which enables the consumer of data to take advantage of the C# language and issue LINQ (Language Integrated Queries) on data. This feature enables users to write an integrated query, select a custom subset of data and pass it to the other components of the system.

[1] `http://mymedialite.net`
[2] `http://mahout.apache.org`
[3] `http://lenskit.grouplens.org`

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.
ACM 978-1-4503-2668-1/14/10.
http://dx.doi.org/10.1145/2645710.2645717.

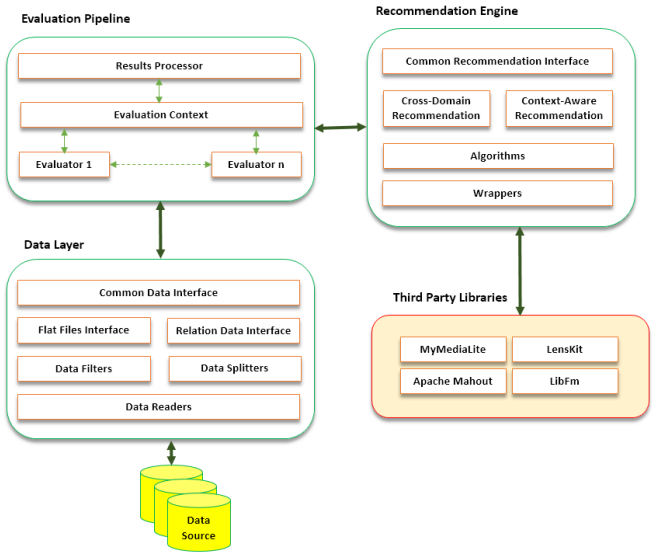

Figure 1: The overall architecture of WrapRec.

- **Recommendation Engine**: This component is the core of the toolkit. It provides common interfaces to RecSys algorithms. The focus of this component is not to implement the RecSys algorithms, but to provide a common interface to different algorithms that are implemented in various libraries. If users want to use an algorithm from existing libraries, they need to wrap the functionality of the third party libraries into the toolkit. Currently, the toolkit provides wrapper classes for MyMediaLite [1] (written in C#) and LibFM [2] (written in C++). This component also provides interfaces to implement Cross-Domain and Context-Aware recommendation algorithms.

- **Evaluation Pipeline**: WrapRec provides evaluation services based on the pipeline design pattern[4]. This enables the evaluator objects to share their results on a context object and thus improve the performance and re-usability of code. Pipeline-based evaluation allows users to re-use the evaluation logic for different RecSys algorithms since the evaluators only talk with high level data objects which are independent of the algorithms.

3. HOW TO USE THE TOOLKIT

The current version of WrapRec is implemented as a .Net library which can be added to a .Net project and be called through its public APIs. The WrapRec toolkit can be easily installed in a .Net project by installing its package through the *Nuget*[5] package management system, or by adding the library file directly to the references of the project. The source code of the toolkit is publicly available under GPL license on Github[6].

A major goal of WrapRec is to be easy to use. Based on the architecture of the system, to run a RecSys experiment users need to define three main objects: a data interface, an algorithm interface and an evaluation pipeline. More advance scenarios can be defined by extending each of the mentioned three objects. Listing 1 lists a simple RecSys experiment which trains a model on the Movielens 1M dataset[7] using MyMediaLite's rating prediction algorithms. The model is then tested using two evaluation metrics on 30% of the data.

Listing 1: Sample three step code to run a recommender system experiment

```
// step 1: dataset
var data = new Dataset<ItemRating>(new
    MovieLensReader("data.dat"), 0.7);

// step 2: recommender
var recommender = new MediaLiteRatingPredictor(new
    MatrixFactorization());

// step3: evaluation
var ep = new EvaluationPipeline(new
    EvalutationContext(recommender, data));

ep.Evaluators.Add(new RMSE());
ep.Evaluators.Add(new MAE());

// run the experiments
ep.Run();
```

Each of the three steps in the above example can be extended without modifying other steps. For example, to read a dataset with a different format, a different dataset reader needs to be defined in step 1. More advanced scenarios and samples can be found in the online documentation of the toolkit.

4. OUTLOOK

In this demo paper, we briefly introduced an open source RecSys toolkit that can be used for various purposes. The toolkit is currently available on Github and is in active development. The current version of the toolkit is available as a Nuget library and can be easily installed in any .Net project. In the future, we plan to further integrate this project with MyMediaLite and other existing libraries to take the advantages of the existing frameworks more.

Acknowledgment

This research is supported by funding from two European Commission's 7th Framework Program projects under grant agreements no. 610594 (CrowdRec) and no. 601166 (PHENICX).

5. REFERENCES

[1] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMediaLite: A free recommender system library. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*, 2011.

[2] S. Rendle. Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol.*, 3(3), May 2012.

[4] msdn.microsoft.com/en-us/library/ff963548.aspx
[5] http://www.nuget.org/
[6] https://github.com/babakx/WrapRec
[7] http://grouplens.org/datasets/movielens/