Implemented Methods

Rating Prediction

- averages: global, user, item
- linear baseline method by Koren and Bell
- frequency-weighted Slope One
- k-nearest neighbor (kNN):
  - based on user or item similarities, with different similarity measures
  - collaborative or attribute-/content-based
- (biased) matrix factorization; factor-wise/SGD training; optimized for RMSE or MAE

Item Prediction

- random
- most popular item
- linear content-based model optimized for BPR (BPR-Linear)
- support-vector machine using item attributes
- k-nearest neighbor (kNN):
  - based on user or item similarities
  - collaborative or attribute-/content-based
- weighted regularized matrix factorization (WR-MF)
- matrix factorization optimized for Bayesian Personalized Ranking (BPR-MF)

Download

Get the latest release of MyMediaLite here:

`http://ismll.de/mymedialite`

Contact

We are always happy about feedback (suggestions, bug reports, patches, etc.). To contact us, send an e-mail to

`mymedialite@ismll.de`

Follow us on Twitter: `@mymedialite`

Acknowledgements

MyMediaLite was developed by Zeno Gantner, Steffen Rendle, and Christoph Freudenthaler at University of Hildesheim.

This work was partly funded by the European Commission FP7 project MyMedia (Dynamic Personalization of Multimedia under the grant agreement no. 215006.

SEVENTH FRAMEWORK PROGRAMME

MyMediaLite – Recommender System Algorithm Library

Universität Hildesheim

Machine Learning Lab

MyMediaLite is a lightweight, multi-purpose library of recommender system algorithms. It addresses the two most common scenarios in collaborative filtering: **rating prediction** (e.g. on a scale of 1 to 5 stars) and **item prediction from implicit feedback** (e.g. from clicks or purchase actions).

`http://ismll.de/mymedialite`

MyMediaLite's Key Features

Choice:

- dozens of different recommendation methods,
- methods can use collaborative, content, or relational data.

Accessibility:

- Includes evaluation routines for rating and item prediction;
- command line tools that read a simple text-based input format;
- usable from C#, Python, Ruby, F#, etc.,
- complete documentation of the library and its tools.

Compactness: Core library is less than 200 KB "big".

Portability: Written in C#, for the .NET platform; runs on every architecture where Mono works: Linux, Windows, Mac OS X.

Parallel processing on multi-core/multi-processor systems.

Serialization: save and reload recommender models.

Real-time incremental updates for most recommenders.

Freedom: free/open source software (GPL).

Target Groups

Researchers

- Don't waste your time implementing methods if you actually want to study other aspects of recommender systems!
- Use the MyMediaLite recommenders as baseline methods in benchmarks.
- Use MyMediaLite's infrastructure as an easy starting point to implement your own methods.

Developers

- Add recommender system technologies to your software or website.

Educators and Students

- Demonstrate/see how recommender system methods are implemented.
- Use MyMediaLite as a basis for you school projects.

Recommendation Tasks Addressed

Rating Prediction

Popularized by systems like MovieLens, Netflix, or Jester, this is the most-discussed collaborative filtering task in the recommender systems literature. Given a set of ratings, e.g. on a scale from 1 to 5, the goal is predict unknown ratings.

	Alice	*Ben*	*Christine*
The Usual Suspects	5		4
American Beauty	3	4	3
The Godfather		??	1
Road Trip	2		

Implicit Feedback Item Recommendation

Getting ratings from users requires explicit actions from their side. Much more data is available in the form of implicit feedback, e.g. whether a user has viewed or purchased a product in an online shop. Very often this information is positive-only, i.e. we know users like the products they buy, but we cannot easily assume that they do not like everything they have not (yet) bought.