MyMediaLite Manual

Zeno Gantner, Steffen Rendle

September 26, 2010

Contents

Chapter 1

Recommender Engines

1.1 Rating Predictors

1.1.1 Global average

The global average engine is one of the simplest engines available. Its prediction is based on the average of all observed ratings (so for all items and all users). This is a baseline method, useful mainly when other algorithms lack the required amount of data to be sufficiently accurate: the item average engine for example cannot be used on items that have not yet been rated by anyone.

This engine does not support online updates. However, full training is very fast.

Configuration There is no configuration necessary.

Implementation `MyMediaLite.rating_predictor.GlobalAverage`

1.1.2 Item average

Similarly to the global average engine, the item average engine is based on the average of the observed ratings, but for a specific item. In other words, this returns the average of all ratings of an item made by all users.

This engine is less suitable for items that have received no or very few ratings.

This engine does not support online updates. However, full training is very fast.

Configuration There is no configuration necessary.

Implementation `MyMediaLite.rating_predictor.ItemAverage`

1.1.3 User average

The prediction of the user average engine is based on the average of the observed ratings for a specific user. In other words, this returns the average of all ratings made by a specific user.

This engine is less suitable for users that have rated no or very few items.

This engine does not support online updates. However, full training is very fast.

Configuration There is no configuration necessary.

Implementation `MyMediaLite.rating_predictor.UserAverage`

1.1.4 Matrix factorization

Matrix factorization is a method for approximating the true unobserved ratings matrix R by TODO, where the u-th row w_u of W contains the k features that describe the u-th user (e.g. how much a user likes action movies) and the i-th row h_i of H contains k corresponding features for the i-th item (e.g. how much this movie is an action movie). This means that the predicted rating of an item for a specific user is the dot product of the user's features and the item's features. The feature matrices can be seen as latent variables – i.e. they are unobserved values. This means that no features are given in advance, instead the best "features" are automatically learned from the feedback data. For learning one usually tries to find the best values of the matrices on the observed feedback. For preventing overfitting, the learning method should use some kind of regularization. Using a higher feature dimension can improve the quality of the prediction but makes the computations more resource intensive.

This engine is less suitable for new users (that have rated no or very few items) and new items (that have received no or very few ratings).

This engine supports online updates.

Configuration

num_features	The number of feature dimensions of the factorization.
learn_rate	Learn rate for training.
regularization	Regularization constant to prevent overfitting.
init_f_mean	The feature values are initialized by drawing from the normal distribution N(init_f_mean, init_f_stdev).
init_f_stdev	The feature values are initialized by drawing from the normal distribution N(init_f_mean, init_f_stdev).
num_iter	Number of iterations over the training data.

Data requirements This algorithm expects the 1-st attribute of the Rating relationship to be the user ID, the 2-nd to be the item ID and the 3-rd to be the rating value. No other attributes are used.

Implementation `MyMediaLite.rating_predictor.MatrixFactorization`
The implementation uses a gradient descent method for minimizing the regularized error.

1.2 Item Recommenders

The following engines recommend items based on implicit feedback. E.g. a user has viewed/ purchased an item. These engines can be used to recommend a ranked list of items.

1.2.1 Most Popular

The items that are most popular over all users are recommended.
This engine supports online updates. Full training is very fast.

Configuration

Implementation `MyMediaLite.item_recommender.MostPopular`

1.2.2 K-Nearest Neighbor (Item-Based)

This engine ranks items based on the feedback history of the user and similar items. The similarity of a pair of items is measured by the cosine similarity. Then for prediction, the similarity of the current item to items in the feedback history of the specific user is summed up. According to these personalized scores the items are ranked.
This engine does not support online updates. Entity and relation updates will simply be ignored.

Configuration TODO

Implementation `MyMediaLite.item_recommender.kNN`

1.2.3 User-Based K-Nearest Neighbor

This engine ranks items based on their feedback history. The similarity of two users is measured by the cosine similarity. Then, for prediction, the similarity of the user to users in the feedback history of the specific item is summed up. According to these personalized scores the items are ranked.
This engine does not support online updates. Entity and relation updates will simply be ignored.

Configuration TODO

Implementation `MyMediaLite.item_recommender.UserkNN`

1.2.4 Singular Value Decomposition (SVD)

A singular value decomposition on the observed feedback matrix. It is assumed that all unobserved data are negative values.

Configuration

num_features	The number of dimensions of the factorization.
init_f_mean	The feature values are initialized by drawing from the normal distribution N(init_f_mean, init_f_stdev).
init_f_stdev	The feature values are initialized by drawing from the normal distribution N(init_f_mean, init_f_stdev).
num_iter	Number of iterations over the training data.

Implementation `MyMediaLite.item_recommender.SVD`

1.2.5 Weighted Regularized Matrix Factorization (WRMF)

Weighted matrix factorization method proposed by

- Hu et al. and Pan et al.: Y. Hu, Y. Koren, and C. Volinsky: Collaborative filtering for implicit feedback datasets. In IEEE International Conference on Data Mining (ICDM 2008), pages 263--272, 2008.

- R. Pan, Y.Zhou, B. Cao, N. N. Liu, R. M. Lukose, M. Scholz, and Q. Yang: One-class collaborative filtering. In IEEE International Conference on Data Mining (ICDM 2008), pages 502--511, 2008.

We use the fast computation method proposed by Hu et al. and we allow a global weight to penalize observed/unobserved values.

This engine does not support online updates. Entity and relation updates will simply be ignored.

Configuration

num_features	The number of dimensions of the factorization.
regularization	Regularization constant to prevent overfitting.
c_pos	The weight/confidence that is put on positive observations.
init_f_mean	The feature values are initialized by drawing from the normal distribution N(init_f_mean, init_f_stdev).
init_f_stdev	The feature values are initialized by drawing from the normal distribution N(init_f_mean, init_f_stdev).
num_iter	Number of iterations over the training data.

Implementation `MyMediaLite.item_recommender.WRMF`

1.2.6 Matrix Factorization optimized for Bayesian Personalized Ranking (BPR-MF)

Matrix factorization method proposed by

- Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, Lars Schmidt-Thieme (2009): *BPR: Bayesian Personalized Ranking from Implicit Feedback,* in Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009), Montreal, Canada.

This engine supports online updates.

Configuration

num_features	The number of dimensions of the factorization.
learn_rate	Learn rate for training.
reg_u	Regularization constant for the user factors.
reg_i	Regularization constant for the factors of positive items.
reg_j	Regularization constant for the factors of negative items.
init_f_mean	The feature values are initialized by drawing from the normal distribution N(init_f_mean, init_f_stdev).
init_f_stdev	The feature values are initialized by drawing from the normal distribution N(init_f_mean, init_f_stdev).
num_iter	Number of iterations over the training data.

Implementation `MyMediaLite.item_recommender.BPR_MF`

The implementation uses a gradient descent method for optimizing the BPR-Opt criterion.