

## Rectools

Norman Matloff, Pooja Rajkumar  
University of California, Davis

### Abstract

Recommendation engines have a number of different applications. From books to movies, they enable the analysis and prediction of consumer preferences. The prevalence of recommender systems in both the business and computational world has led to clear advances in prediction models over the past years. Current R packages include recosystem and recommenderlab. However, our new package, rectools, currently under development, extends its capabilities in several directions. One of the most important differences is that rectools allows users to incorporate covariates, such as age and gender, to improve predictive ability and better understand consumer behavior. Our software incorporates a number of different methods, such as non-negative matrix factorization, random effects models, and nearest neighbor methods. In addition to our incorporation of covariate capabilities, rectools also integrates several kinds of parallel computation.

## Rectools

### Recommendation Engines

A recommender system is an engine to predict the rating or preference that a user would give an item. Recommender systems are pervasive on e-commerce websites in particular, which traditionally utilize collaborative filtering for their methods. This package includes several interesting features, particularly a use of the latent factor model in predictions.

### Methods

#### The data set

Overall, the package takes in data sets in the following form:

#### findYdots

findYdots allows us to use a latent factor model to predict values in our data set.

Ydots makes use of the following equation:

$$Y_{ij} = \mu + E\{\alpha_i\} + E\{\beta_j\} + \varepsilon_{ij}$$

$$Y_{ij} = \textit{predicted value}$$

$$E\{\alpha_i\} = Y_{i.} - Y_{..}$$

$$E\{\beta_j\} = Y_{.j} - Y_{..}$$

$$\varepsilon_{ij} = \textit{extraneous variables}$$

Suppose we have a user Ali. Ali has seen the following movies:

userID	movieID	rating
13	10	2
2	100	3
..	...	...

Our estimated  $\alpha_i$  here is the tendency for a user to rate a particular item compared to everyone else. Thus, the  $\alpha_{\text{Ali}}$  here would be Ali's tendency to be an "easy" or "hard" rater. For example, if the average rating for movies is a 3 and Ali is generally harsh grader (with primarily 1s and 2s) then Ali's  $\alpha_{\text{Ali}}$  would be negative. Vice versa, Ali's alpha would be positive if Ali is a generally easy grader.

Our estimated  $\beta_j$  is the tendency for the movie to be rated really highly or poorly. For example, a movie that does really well will have generally high ratings, and thus, a generally high  $\beta_j$ . If a movie has generally poor ratings, then the  $\beta_j$  will be low.

There are two predictive methods that we implement in our ydots method:

1. Method of Moments
2. Maximum Likelihood

### Method of Moments

Method of Moments regresses the ratings against covariates such as age, gender, or genre. We then subtract these predictions from the actual value and apply the latent factor model. In order to use method of moments, make sure your data set (called ratingsIn) follows the same format as below:

<b>userID</b>	<b>itemID</b>	<b>rating</b>
8	1	2
99	5	5
..	...	...

### findYdotsMM

```
findYdotsMM <- function(ratingsIn, regressYdots=FALSE, cls=NULL)
```

### regressYdots

If regressYdots is true, apply lm to the estimated latent factors and their product, enabling rating prediction from the resulting linear function of the factors. This is currently implemented as if there are no covariates.

### cls

Para