recommenderlab\_example.R

11000306

Thu May 19 14:21:33 2016

library("recommenderlab")

## Loading required package: Matrix

## Loading required package: registry

## Loading required package: arules

##   
## Attaching package: 'arules'

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

## Loading required package: proxy

##   
## Attaching package: 'proxy'

## The following object is masked from 'package:Matrix':  
##   
## as.matrix

## The following objects are masked from 'package:stats':  
##   
## as.dist, dist

## The following object is masked from 'package:base':  
##   
## as.matrix

m <- matrix(sample(c(as.numeric(0:5), NA), 50, replace=TRUE, prob=c(rep(.4/6,6),.6)), ncol=10, dimnames=list(user=paste("u", 1:5, sep=''),item=paste("i", 1:10, sep=''))  
)  
m

## item  
## user i1 i2 i3 i4 i5 i6 i7 i8 i9 i10  
## u1 NA NA NA 4 3 NA NA NA NA NA  
## u2 NA 1 NA 1 NA NA NA NA NA 4  
## u3 NA NA 3 NA NA NA 5 5 1 1  
## u4 2 4 NA 1 NA 4 1 NA 1 0  
## u5 NA NA 0 NA NA NA 1 NA NA 3

r <- as(m, "realRatingMatrix")  
r

## 5 x 10 rating matrix of class 'realRatingMatrix' with 20 ratings.

#as(r,"dgCMatrix")  
  
identical(as(r, "matrix"),m)

## [1] TRUE

as(r, "list")

## $u1  
## i4 i5   
## 4 3   
##   
## $u2  
## i2 i4 i10   
## 1 1 4   
##   
## $u3  
## i3 i7 i8 i9 i10   
## 3 5 5 1 1   
##   
## $u4  
## i1 i2 i4 i6 i7 i9 i10   
## 2 4 1 4 1 1 0   
##   
## $u5  
## i3 i7 i10   
## 0 1 3

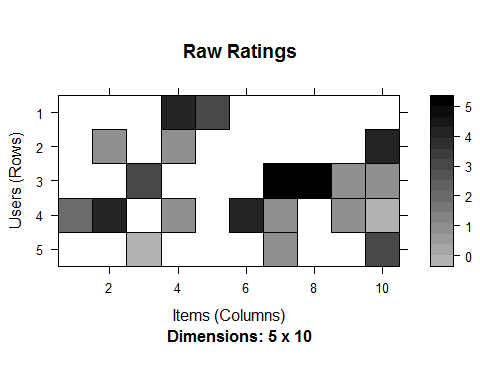
head(as(r, "data.frame"))

## user item rating  
## 6 u1 i4 4  
## 9 u1 i5 3  
## 2 u2 i2 1  
## 7 u2 i4 1  
## 17 u2 i10 4  
## 4 u3 i3 3

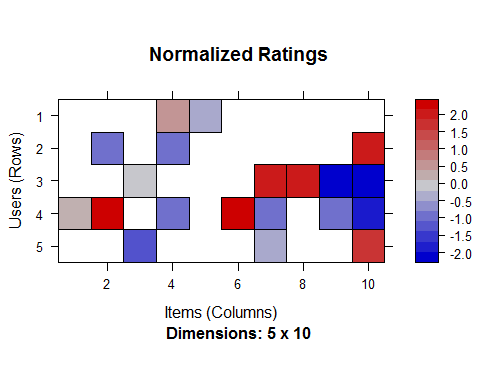
r\_m <- normalize(r)  
r\_m

## 5 x 10 rating matrix of class 'realRatingMatrix' with 20 ratings.  
## Normalized using center on rows.

image(r, main = "Raw Ratings")



image(r\_m, main = "Normalized Ratings")



r\_b <- binarize(r, minRating=1)  
r\_b

## 5 x 10 rating matrix of class 'binaryRatingMatrix' with 18 ratings.

as(r\_b, "matrix")

## i1 i2 i3 i4 i5 i6 i7 i8 i9 i10  
## u1 FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE  
## u2 FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE  
## u3 FALSE FALSE TRUE FALSE FALSE FALSE TRUE TRUE TRUE TRUE  
## u4 TRUE TRUE FALSE TRUE FALSE TRUE TRUE FALSE TRUE FALSE  
## u5 FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE

#################  
  
data("Jester5k")  
Jester5k

## 5000 x 100 rating matrix of class 'realRatingMatrix' with 362106 ratings.

r <- sample(Jester5k, 1000)  
r

## 1000 x 100 rating matrix of class 'realRatingMatrix' with 71081 ratings.

#first user ratings  
rowCounts(r[1,])

## u5707   
## 62

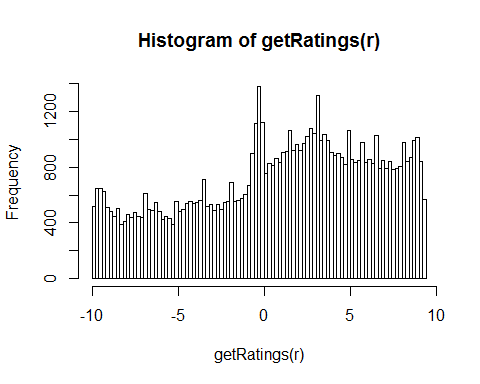
as(r[1,], "list")

## $u5707  
## j5 j6 j7 j8 j10 j11 j12 j13 j14 j15 j16 j17   
## -6.84 0.58 -6.80 -5.73 -9.08 -7.67 -6.89 6.07 4.13 5.49 6.02 5.63   
## j18 j19 j20 j21 j22 j23 j24 j25 j26 j27 j28 j29   
## -7.67 5.78 6.65 6.55 3.06 -7.62 -6.36 6.99 7.48 3.69 -0.29 -8.25   
## j30 j31 j32 j34 j35 j36 j38 j39 j40 j41 j42 j43   
## -6.17 -6.36 -7.33 -5.00 2.14 -5.05 -5.92 7.14 -7.72 6.70 8.83 -6.70   
## j45 j46 j47 j48 j49 j50 j51 j52 j53 j54 j55 j56   
## 5.78 3.30 2.86 -5.49 -4.42 4.90 -6.26 -6.31 6.21 -6.41 -7.04 1.89   
## j57 j59 j61 j62 j63 j64 j65 j66 j67 j68 j69 j70   
## -6.94 -4.17 1.60 -0.29 -8.01 -6.36 1.84 1.65 1.31 0.00 6.07 3.93   
## j93 j99   
## 3.16 7.33

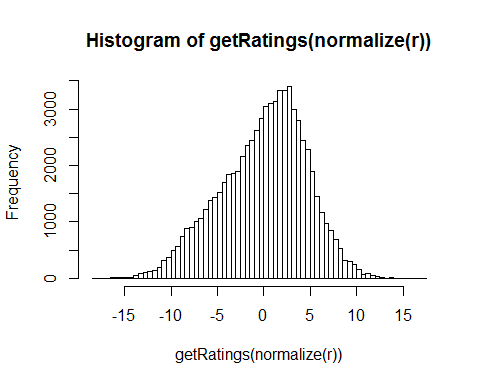
rowMeans(r[1,])

## u5707   
## -0.6514516

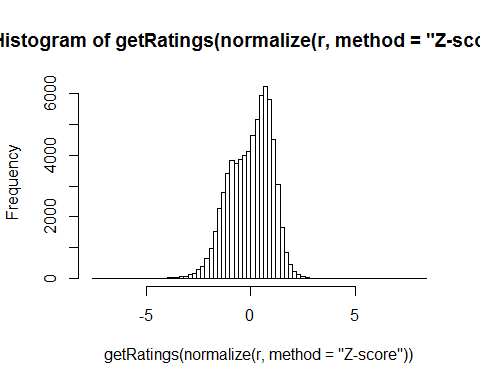
hist(getRatings(r), breaks=100)



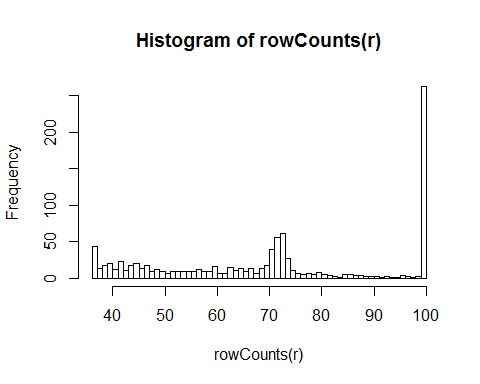
hist(getRatings(normalize(r)), breaks=100)



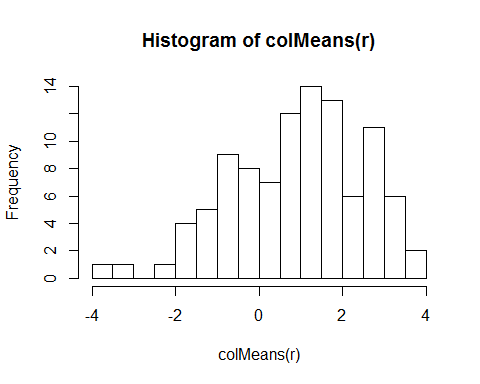
hist(getRatings(normalize(r, method="Z-score")), breaks=100)



hist(rowCounts(r), breaks=50) #how many jokes each user has rated



hist(colMeans(r), breaks=20) #mean rating for each Joke



recommenderRegistry$get\_entries(dataType = "realRatingMatrix")

## $IBCF\_realRatingMatrix  
## Recommender method: IBCF  
## Description: Recommender based on item-based collaborative filtering (real data).  
## Parameters:  
## k method normalize normalize\_sim\_matrix alpha na\_as\_zero minRating  
## 1 30 Cosine center FALSE 0.5 FALSE NA  
##   
## $PCA\_realRatingMatrix  
## Recommender method: PCA  
## Description: Recommender based on PCA approximation (real data).  
## Parameters:  
## categories method normalize normalize\_sim\_matrix alpha na\_as\_zero  
## 1 20 Cosine center FALSE 0.5 FALSE  
## minRating  
## 1 NA  
##   
## $POPULAR\_realRatingMatrix  
## Recommender method: POPULAR  
## Description: Recommender based on item popularity (real data).  
## Parameters: None  
##   
## $RANDOM\_realRatingMatrix  
## Recommender method: RANDOM  
## Description: Produce random recommendations (real ratings).  
## Parameters: None  
##   
## $SVD\_realRatingMatrix  
## Recommender method: SVD  
## Description: Recommender based on EM-based SVD approximation from package bcv (real data).  
## Parameters:  
## approxRank maxiter normalize minRating  
## 1 NA 100 center NA  
##   
## $UBCF\_realRatingMatrix  
## Recommender method: UBCF  
## Description: Recommender based on user-based collaborative filtering (real data).  
## Parameters:  
## method nn sample normalize minRating  
## 1 cosine 25 FALSE center NA

r <- Recommender(Jester5k[1:1000], method = "POPULAR")  
r

## Recommender of type 'POPULAR' for 'realRatingMatrix'   
## learned using 1000 users.

names(getModel(r))

## [1] "topN" "ratings" "normalize" "aggregation" "verbose"

getModel(r)$topN

## Recommendations as 'topNList' with n = 100 for 1 users.

recom <- predict(r, Jester5k[1001:1002], n=5)  
recom

## Recommendations as 'topNList' with n = 5 for 2 users.

as(recom, "list")

## [[1]]  
## [1] "j89" "j72" "j47" "j93" "j76"  
##   
## [[2]]  
## [1] "j89" "j93" "j76" "j88" "j96"

recom3 <- bestN(recom, n = 3)  
recom3

## Recommendations as 'topNList' with n = 3 for 2 users.

as(recom3, "list")

## [[1]]  
## [1] "j89" "j72" "j47"  
##   
## [[2]]  
## [1] "j89" "j93" "j76"

recom <- predict(r, Jester5k[1001:1002], type="ratings")  
recom

## 2 x 100 rating matrix of class 'realRatingMatrix' with 97 ratings.

as(recom, "matrix")[,1:10]

## j1 j2 j3 j4 j5 j6 j7 j8 j9  
## u20089 4.152279 -3.103666 0.8286388 -3.801586 NA NA NA NA -3.025319  
## u11691 NA NA 0.8286388 -3.801586 NA NA NA NA -3.025319  
## j10  
## u20089 -1.108035  
## u11691 NA

#evaluation  
e <- evaluationScheme(Jester5k[1:1000], method="split", train=0.9,given=15, goodRating=5)  
e

## Evaluation scheme with 15 items given  
## Method: 'split' with 1 run(s).  
## Training set proportion: 0.900  
## Good ratings: >=5.000000  
## Data set: 1000 x 100 rating matrix of class 'realRatingMatrix' with 72358 ratings.

r1 <- Recommender(getData(e, "train"), "UBCF")  
r2 <- Recommender(getData(e, "train"), "IBCF")  
r1

## Recommender of type 'UBCF' for 'realRatingMatrix'   
## learned using 900 users.

r2

## Recommender of type 'IBCF' for 'realRatingMatrix'   
## learned using 900 users.

p1 <- predict(r1, getData(e, "known"), type="ratings")  
p2 <- predict(r2, getData(e, "known"), type="ratings")  
p1

## 100 x 100 rating matrix of class 'realRatingMatrix' with 8500 ratings.

p2

## 100 x 100 rating matrix of class 'realRatingMatrix' with 8396 ratings.

error <- rbind(calcPredictionAccuracy(p1, getData(e, "unknown")),calcPredictionAccuracy(p2, getData(e, "unknown")))  
rownames(error) <- c("UBCF","IBCF")  
error

## RMSE MSE MAE  
## UBCF 4.477513 20.04812 3.540851  
## IBCF 4.760657 22.66386 3.607749

#evaluation of topN  
scheme <- evaluationScheme(Jester5k[1:1000], method="cross", k=4, given=3,goodRating=5)  
scheme

## Evaluation scheme with 3 items given  
## Method: 'cross-validation' with 4 run(s).  
## Good ratings: >=5.000000  
## Data set: 1000 x 100 rating matrix of class 'realRatingMatrix' with 72358 ratings.

results <- evaluate(scheme, method="POPULAR", type = "topNList",n=c(1,3,5,10,15,20))

## POPULAR run fold/sample [model time/prediction time]  
## 1 [0sec/0.09sec]   
## 2 [0.02sec/0.11sec]   
## 3 [0.01sec/0.13sec]   
## 4 [0.02sec/0.14sec]

results

## Evaluation results for 4 folds/samples using method 'POPULAR'.

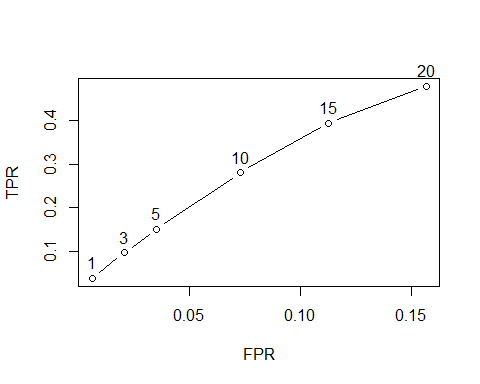
getConfusionMatrix(results)[[1]]

## TP FP FN TN precision recall TPR FPR  
## 1 0.476 0.524 19.432 76.568 0.4760 0.03162236 0.03162236 0.006359941  
## 3 1.356 1.644 18.552 75.448 0.4520 0.08461059 0.08461059 0.020230987  
## 5 2.248 2.752 17.660 74.340 0.4496 0.14236526 0.14236526 0.034000680  
## 10 4.312 5.688 15.596 71.404 0.4312 0.26254597 0.26254597 0.070434521  
## 15 6.312 8.688 13.596 68.404 0.4208 0.37429311 0.37429311 0.107401393  
## 20 7.940 12.060 11.968 65.032 0.3970 0.45840282 0.45840282 0.149651711

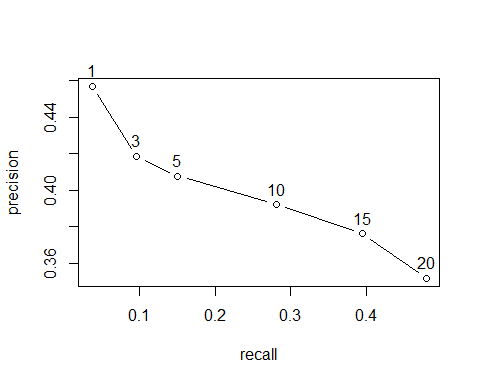
avg(results)

## TP FP FN TN precision recall TPR FPR  
## 1 0.457 0.543 16.702 79.298 0.4570000 0.03729291 0.03729291 0.006409841  
## 3 1.256 1.744 15.903 78.097 0.4186667 0.09603424 0.09603424 0.020957481  
## 5 2.039 2.961 15.120 76.880 0.4078000 0.15073964 0.15073964 0.035591830  
## 10 3.922 6.078 13.237 73.763 0.3922000 0.28077512 0.28077512 0.073134501  
## 15 5.643 9.357 11.516 70.484 0.3762000 0.39472865 0.39472865 0.112719916  
## 20 7.036 12.964 10.123 66.877 0.3518000 0.47940354 0.47940354 0.156708900

plot(results,annotate=TRUE)



plot(results, "prec/rec", annotate=TRUE)



#Comparing top-N recommendations  
set.seed(2016)  
scheme <- evaluationScheme(Jester5k[1:1000], method="split", train = .9,k=1, given=-5, goodRating=5)  
scheme

## Evaluation scheme using all-but-5 items  
## Method: 'split' with 1 run(s).  
## Training set proportion: 0.900  
## Good ratings: >=5.000000  
## Data set: 1000 x 100 rating matrix of class 'realRatingMatrix' with 72358 ratings.

algorithms <- list( "random items" = list(name="RANDOM", param=NULL), "popular items" = list(name="POPULAR", param=NULL), "user-based CF" = list(name="UBCF", param=list(nn=50)), "item-based CF" = list(name="IBCF", param=list(k=50)), "SVD approximation" = list(name="SVD", param=list(approxRank = 50)) )  
  
results <- evaluate(scheme, algorithms, type = "topNList", n=c(1, 3, 5, 10, 15, 20))

## RANDOM run fold/sample [model time/prediction time]  
## 1 [0sec/0.05sec]   
## POPULAR run fold/sample [model time/prediction time]  
## 1 [0.02sec/0.04sec]   
## UBCF run fold/sample [model time/prediction time]  
## 1 [0sec/0.83sec]   
## IBCF run fold/sample [model time/prediction time]  
## 1 [0.35sec/0.04sec]   
## SVD run fold/sample [model time/prediction time]  
## 1 [0.02sec/0.13sec]

results

## List of evaluation results for 5 recommenders:  
## Evaluation results for 1 folds/samples using method 'RANDOM'.  
## Evaluation results for 1 folds/samples using method 'POPULAR'.  
## Evaluation results for 1 folds/samples using method 'UBCF'.  
## Evaluation results for 1 folds/samples using method 'IBCF'.  
## Evaluation results for 1 folds/samples using method 'SVD'.

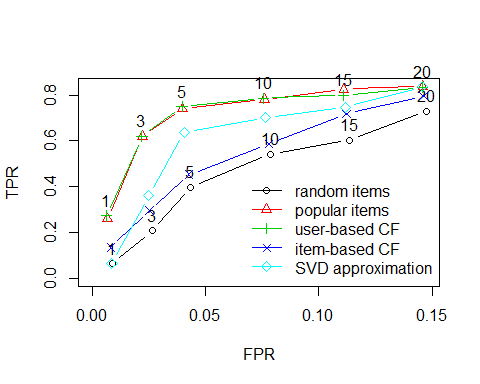
names(results)

## [1] "random items" "popular items" "user-based CF"   
## [4] "item-based CF" "SVD approximation"

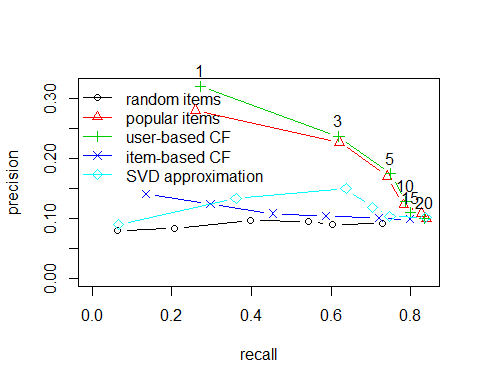
results[["user-based CF"]]

## Evaluation results for 1 folds/samples using method 'UBCF'.

plot(results, annotate=c(1,3), legend="bottomright")



plot(results, "prec/rec", annotate=3, legend="topleft")



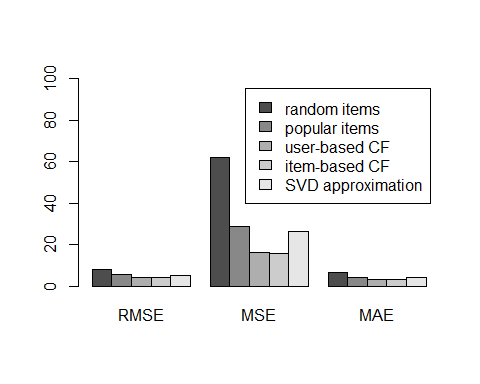
#Comparing ratings  
results <- evaluate(scheme, algorithms, type = "ratings")

## RANDOM run fold/sample [model time/prediction time]  
## 1 [0sec/0sec]   
## POPULAR run fold/sample [model time/prediction time]  
## 1 [0.01sec/0.02sec]   
## UBCF run fold/sample [model time/prediction time]  
## 1 [0sec/0.84sec]   
## IBCF run fold/sample [model time/prediction time]  
## 1 [0.44sec/0.01sec]   
## SVD run fold/sample [model time/prediction time]  
## 1 [0sec/0.08sec]

results

## List of evaluation results for 5 recommenders:  
## Evaluation results for 1 folds/samples using method 'RANDOM'.  
## Evaluation results for 1 folds/samples using method 'POPULAR'.  
## Evaluation results for 1 folds/samples using method 'UBCF'.  
## Evaluation results for 1 folds/samples using method 'IBCF'.  
## Evaluation results for 1 folds/samples using method 'SVD'.

plot(results, ylim = c(0,100))



#Using a 0-1 data set  
Jester\_binary <- binarize(Jester5k, minRating=5)  
Jester\_binary <- Jester\_binary[rowCounts(Jester\_binary)>20]  
Jester\_binary

## 1797 x 100 rating matrix of class 'binaryRatingMatrix' with 65642 ratings.

scheme\_binary <- evaluationScheme(Jester\_binary[1:1000],method="split", train=.9, k=1, given=3)  
scheme\_binary

## Evaluation scheme with 3 items given  
## Method: 'split' with 1 run(s).  
## Training set proportion: 0.900  
## Good ratings: NA  
## Data set: 1000 x 100 rating matrix of class 'binaryRatingMatrix' with 36468 ratings.

results\_binary <- evaluate(scheme\_binary, algorithms,type = "topNList", n=c(1,3,5,10,15,20))

## RANDOM run fold/sample [model time/prediction time]  
## 1 [0sec/0.03sec]   
## POPULAR run fold/sample [model time/prediction time]  
## 1 [0sec/0.36sec]   
## UBCF run fold/sample [model time/prediction time]  
## 1 [0sec/0.53sec]   
## IBCF run fold/sample [model time/prediction time]  
## 1 [0.01sec/0.03sec]   
## SVD run fold/sample [model time/prediction time]  
## 1 Timing stopped at: 0 0 0

## Warning in .local(x, method, ...):   
## Recommender 'SVD approximation' has failed and has been removed from the results!

plot(results\_binary, annotate=c(1,3), legend="bottomright")

