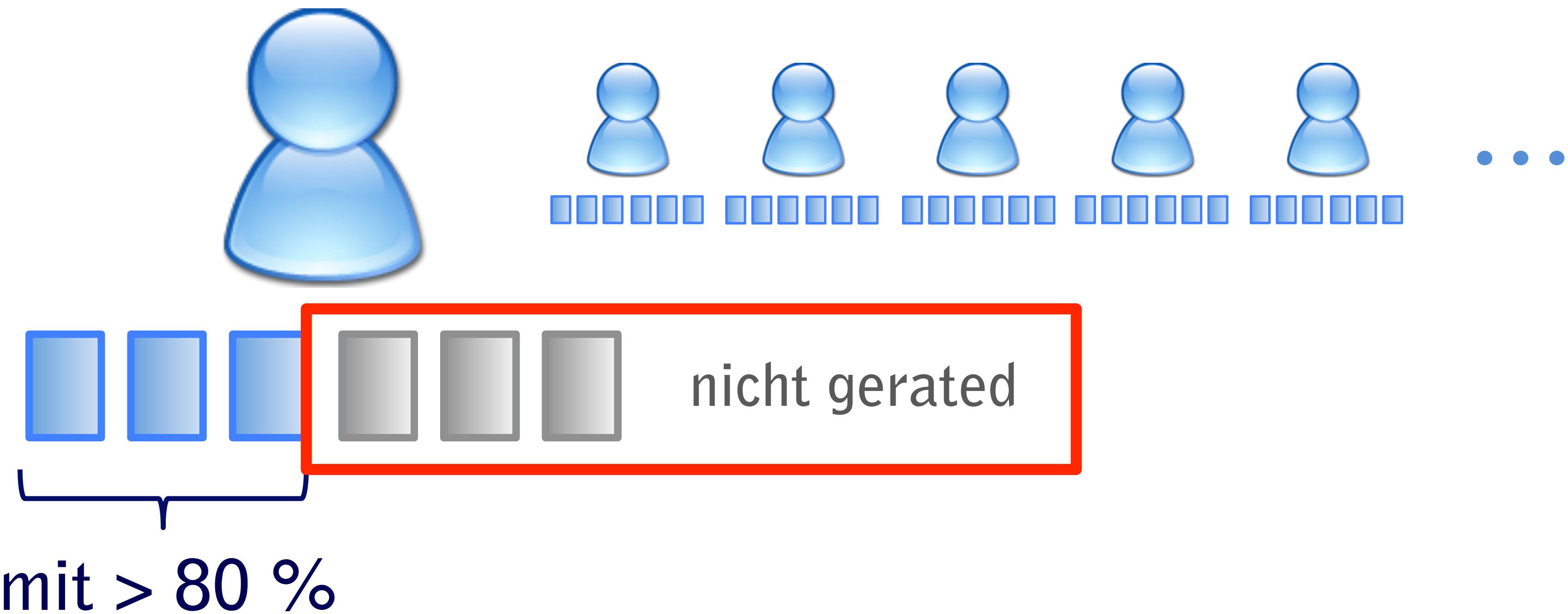
Track 2

Caroline Fetzer,
Martin Köppelmann,
Sebastian Stange

Track 2 – Assignment 2011

#User	#Items	#Ratings	#Train Ratings	#Test Ratings
249.012	296.111	62.551.438	61.944.406	607.032



Winner of 2007

Who Rated What: a combination of SVD, correlation and frequent sequence mining*

Miklós Kurucz András A. Benczúr Tamás Kiss
István Nagy Adrienn Szabó Balázs Torma
Data Mining and Web search Research Group, Informatics Laboratory
Computer and Automation Research Institute of the Hungarian Academy of Sciences
{realace, benczur, kisstom, iscsi, aszabo, torma}@ilab.sztaki.hu

ABSTRACT
KDD Cup 2007 focuses on predicting aspects of movie rating behavior. We present our prediction method for Task 1 "Who Rated What in 2006" where the task is to predict which users rated which movies in 2006. We use the combination of the following predictors, listed in the order of their efficiency in the prediction:

- The predicted number of ratings for each movie based on time series prediction, also using movie and DVD release dates and movie series detection by the edit distance of the titles.
- The predicted number of ratings by each user by using the fact that ratings were sampled proportional to the margin.

By combining the predictions by linear regression we obtained a prediction with root mean squared error 0.256; the first runner up result was 0.263 while a pure all zeroes prediction already gives 0.279, indicating the hardness of the task.

Categories and Subject Descriptors
J.4 [**Computer Applications**]: Social and Behavioral Sciences; G.1.3 [**Mathematics of Computing**]: Numerical Analysis—*Numerical Linear Algebra*

*This work was supported by the Mobile Innovation Center, Hungary, a Yahoo Faculty Research Grant and by grant *ASTOR* NKFP 2/004/05

General Terms
data mining, recommender systems

Keywords
singular value decomposition, item-item similarity, frequent sequence mining

1. INTRODUCTION
Recommender systems predict the preference of a user on a given item based on known ratings. In order to evaluate methods, in October 2006 Netflix provided movie ratings from anonymous customers on nearly 18 thousand movie titles [5] called the *Prize dataset*. The KDD Cup 2007 tasks were related to this data set. For Task 1 "Who Rated What in 2006" the task was to predict which users rated which movies in 2006 while for Task 2 "How Many Ratings in 2006" the task was to predict the number of additional ratings of movies.

In this paper we present our method for Task 1 "Who Rated What in 2006". The task was to predict the probability that a user rated a movie in 2006 (with the actual date and rating being irrelevant) for a given list of 100,000 user–movie pairs. The users and movies are drawn from the Prize data set, i.e. the movies appeared (or at least received ratings) before 2006 and the users also gave their first rating before 2006 such that none of the pairs were rated in the training set. We give a detailed description of the sampling method in Section 2.2 since it gives information that we use for the prediction.

Our method is summarized as follows:

1. A naive estimate based on a user–movie independence assumption that uses time series analysis and event prediction from the IMDB movie and the `videoeta.com` DVD release dates as well as the user rating amount reconstructed from sample margins.
2. The implementation of an SVD and an item-item similarity based recommender as well as association rule mining for the KDD Cup Task 1.
3. Method fusion by using the machine learning toolkit Weka [26].

We use the *root mean squared error*

1. Place: "Who rated What"
Informatics Lab - Hungary

A classical predictive modeling approach for Task "Who rated what?" of the KDD CUP 2007

Jorge Sueiras, Neo Metrics, C/ Arequipa 1, 28043 Madrid, Spain, +34 91 382 45 54, jorge.sueiras@neo-metrics.com
Alfonso Salafranca, Neo Metrics, C/ Arequipa 1, 28043 Madrid, Spain, +34 91 382 45 54, alfonso.salafranca@neo-metrics.com
Jose Luis Florez, Neo Metrics, C/ Arequipa 1, 28043 Madrid, Spain, +34 91 382 45 54, jose.luis.florez@neo-metrics.com

ABSTRACT
This paper describes one possible way to solve task "Who rated what?" of the KDD CUP 2007. The proposed solution is a history-based model that predicts whether a user will vote a given movie. Key points to our approach are (1) the estimation of the model baseline, (2) the definition of the explanatory variables and (3) the mathematical model form. Given the binary outcome of the problem, the estimation of the true baseline (ratio of 1's in the test data) is critical in order to correctly make predictions. In parallel, to improve the model predictive power, we have developed a careful construction of the input variables. These explanatory variables can be grouped as: user voting behaviour variables, the movie characteristics and user-movie interactions. Finally, the mathematical model form (linear logistic regression) has been chosen among various model form competitors.

Categories and Subject Descriptors
I.5.1 [**Pattern Recognition**]: Models – *statistical*

Keywords
Predictive modeling, data mining.

1. INTRODUCTION
Task 1 at the KDD CUP 2007 is based on the competition organized by Netflix (http://www.netflixprize.com) which provides a historic database of more than 100 million movie ratings [1]. Netflix training data lasts up to December 2005 and the Netflix Competition goal is to build a model which predicts the rating given by a user to a movie. In order to accurately estimate the mean prediction error for each proposed model, Netflix uses a test dataset with 2 million user ratings.

Task 1 at KDD Cup'07 is based on the Netflix data; but the goal is slightly different: Here we are asked to predict whether a user has rated a given movie during 2006. Therefore the model must have a binary outcome.

The first difficulty in this task is to accurately determine the rate of positive events (baseline) on the provided data. In fact, having a look to the final results of the task 1 KDD Cup'07, one can see that just five teams manages to perform better than a benchmark model constructed by assigning to each pair in the scoring data the baseline probability.

Our modeling approach consists of the classical two steps:

1. Model and variable selection. We built a predictive model whose target variable is the binary event of rating a movie in 2005 and whose input variables are created with data up to December 2004. This step includes variable and model form selection.
2. Prediction. Given the model formulation and parameter estimates defined above, the input variables are recalculated using the whole dataset (including 2005). Finally the required predictions for the score dataset are obtained.

The paper is organized as follows: first, we describe how we solved the estimation of the baseline for the year 2006 and how it was used to build the training table. Then the input variables are described and finally the relevance of such variables is discussed.

2. BASELINE ESTIMATION
In order to estimate the baseline we must pay attention to the KDD Cup'07 FAQ's. The FAQ document states that the 100.000 score pairs were selected by randomly picking up pairs (user, movie) with probability proportional to the number of times each component appears in the 2006 ratings; Furthermore the user and the movie are chosen independently.

We consider that correct estimation of the baseline is important in order to attain a good solution to the problem posed.. For baseline estimation we shall proceed to replicate the procedure used to create the scoring data, in order to produce a training dataset with similar characteristics. The sampling algorithm is as follows:

1. Define the time range for the target variable, in our case a whole year, and select those users and movies that

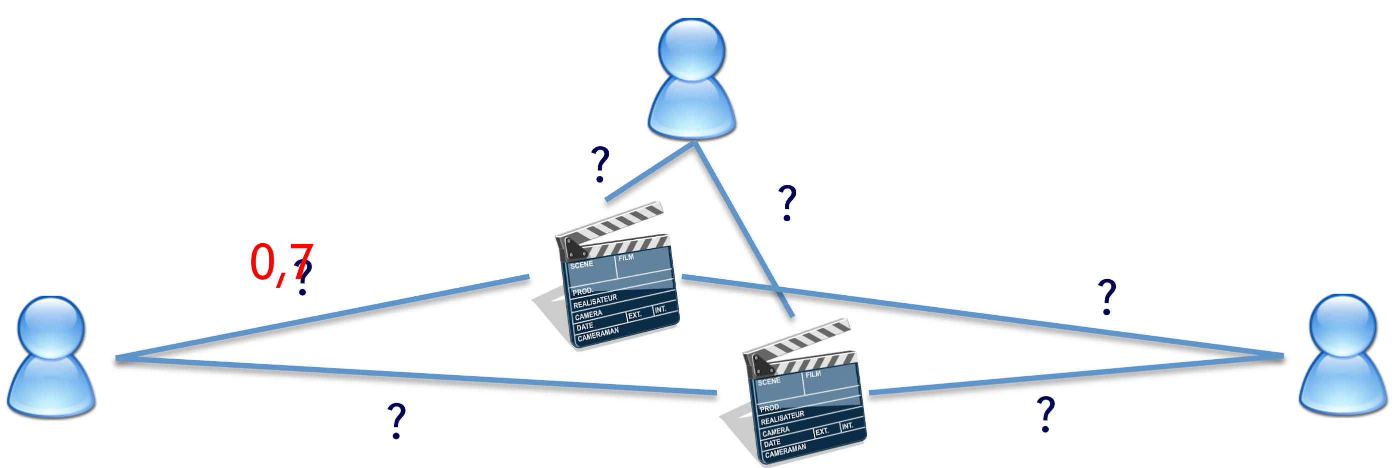
2. Place: "A classical predictive modeling approach"
Neo Metrics – Spain

Track 1 – Assignment 2007

"Which user rated which movies in 2006"

- Data set of the last years
- 100.000 user-movie pairs of 2006

→ Probability that user-movie-pair is rated



Track 2 – Assignment 2007

"How many Ratings"

→ number of total ratings

Paper 1 – Result & Approach

Result: RMSE (stdv) = 0,256

0,5533 * base prediction

\+ 0,1987 * singular value decomposition

\+ 0,029 * item item similarity

\+ -0,0121 * association rules – 0,0042

Paper 1 – Base Prediction

Result: RMSE (stdv) = 0,256

prob (user-movie pair x) = 0

→ 10th – 13th place with stdv = 0,279

guessing correct factors for baseline

→ 5th – 6th Place with stdv = 0,268

Paper 1 – Base Prediction

$$p_{um} = (N_u * N_m) / M * U$$

user-movie-relation independent

to estimate
(Track 2 - 2007)

N_u = number of ratings of the user
N_m = number of ratings of the movie
M = total number of movies
U = total number of users

Paper 1 – Base Prediction

$$p_{um} = (N_u * N_m) / M * U$$

we know
(Track 2 - 2011)

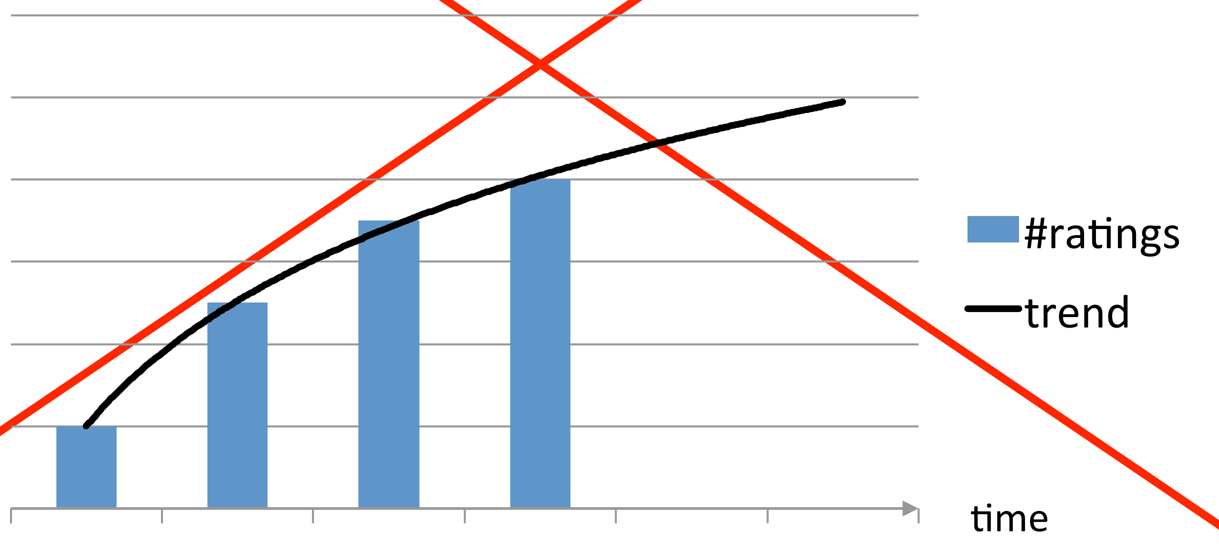
user-movie-relation independent

N_u = number of ratings of the user
N_m = number of ratings of the movie
M = total number of movies
U = total number of users

Paper 1 – Prediction #Ratings/Movie N_m

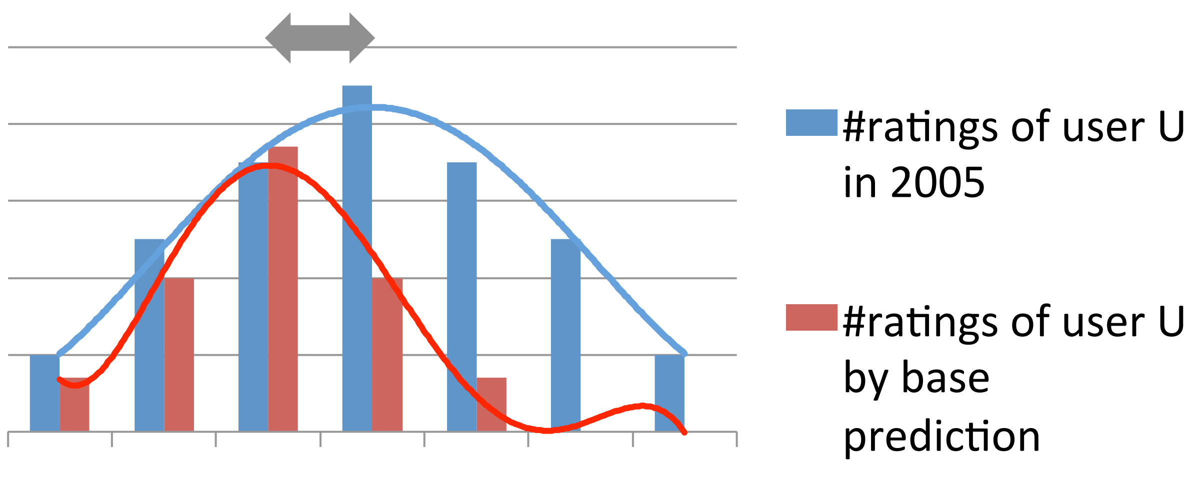
Secondary Information:
(DVD-Release, IMDB Movie Release, series continuation releases)

→ analyse time distribution and continue



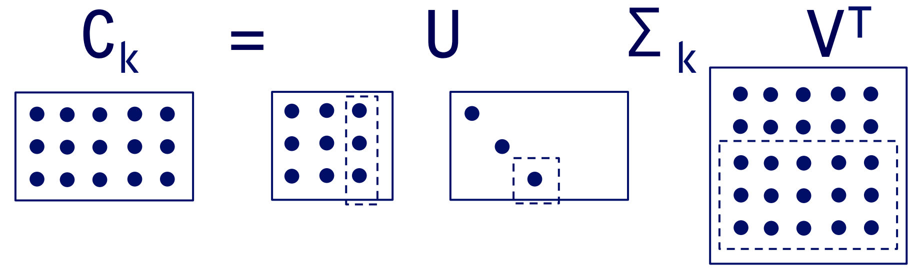
Paper 1 – Prediction #Ratings/User N_u

- same sampling-method as in KDD-Cup 2007
- stdv of sampled ratings of 2005
- stdv of base predictions of 2006
- Compare $stdv_{2006}$ and $stdv_{2005}$
- adapt 2005 to 2006



„Who Rated What […]" - ilab

Paper 1 – SVD



In: u-m-matrix with predictions from several base prediction-values
Out: denser matrix

Eckhart-Young Theorem:
after using svd you got a rank-k-matrix,
which is an approximation of the original matrix

Implementing "Lanczos" (SVD-pack)
Too high number of dimensions leads to "overfitting"
→ Machine learning approach to get optimal k for SVD-partition
→ calculate optimal partition with Frobenius Norm = error value

→ important for us

„Who Rated What [...]" - ilab

Paper 1 – SVD

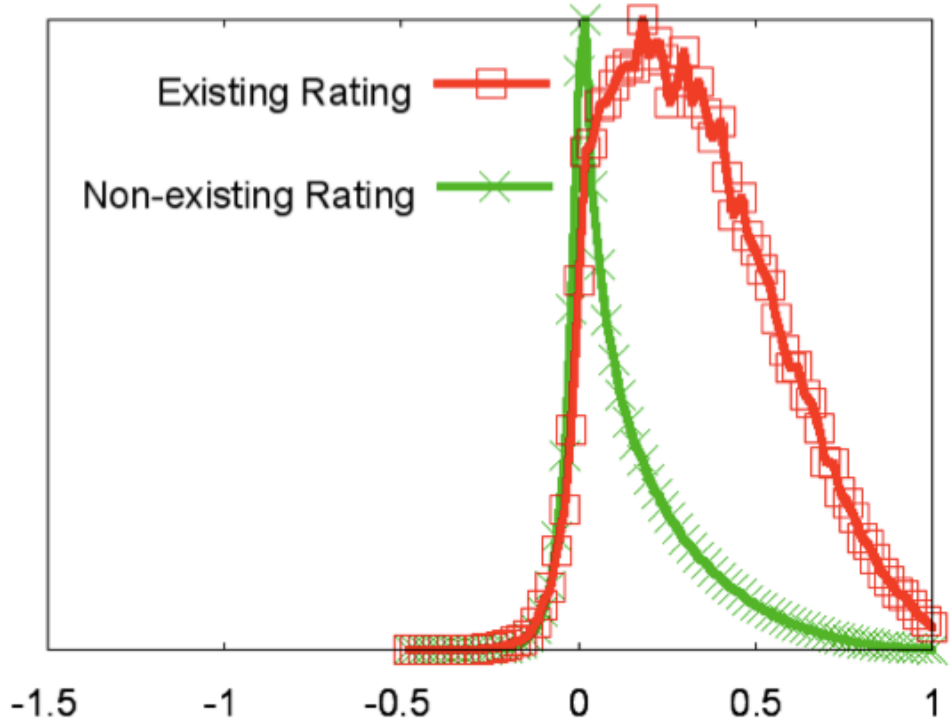


Figure 1: The distribution of the 10-dimensional approximation for user–movie pairs with and without ratings.

Paper 1 – Item Item Similarity

Cosine similarity:

vectors		Item a	Item b	Item c
i	User 1	20	30	40
j	User 2	42	23	66
k	User 3	10	90	30

$$sim(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

Paper 1 – Item Item Similarity

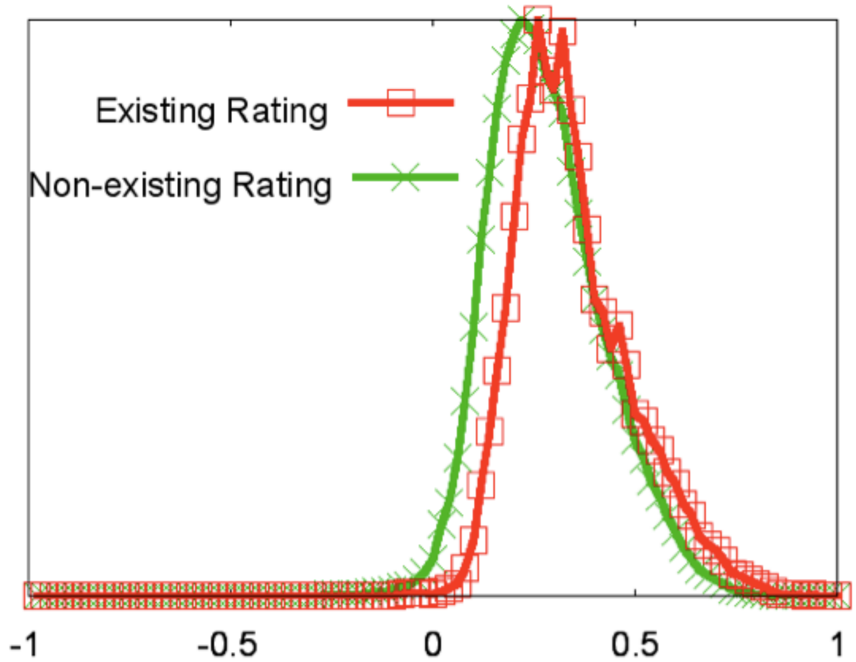
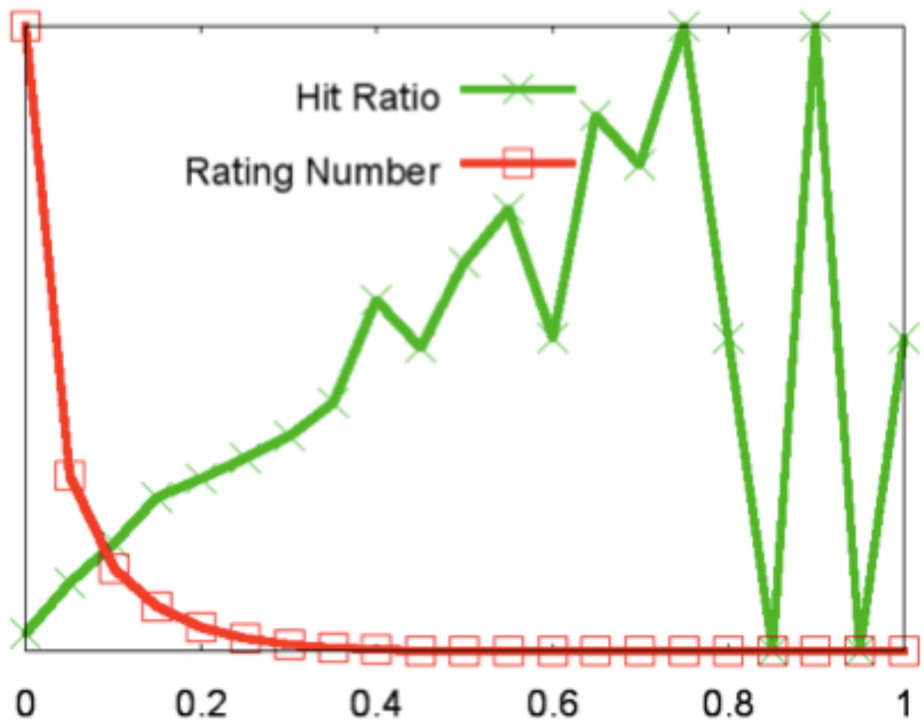


Figure 2: The distribution of the item-item similarity based prediction for user–movie pairs with and without ratings for a similarity top list of size $K = 5$.

„Who Rated What [...]" - ilab

Paper 1 – Machine Learning

Weka toolkit:
- training data: sample of 2005 (sampling method 2007)
- Applied to data of 2006



„ classical modeling approach [...]“ – Neo Metrics

Paper 2 – Result & Approach

Result: RMSE (stdv) = 0,263

- deliberate selection of variables
- constructing more variables with SVD
- Machine learning over all variables with own training sample

„ classical modeling approach [...]" – Neo Metrics

Paper 2 – Baseline

guessed baseline: ca. 20%

baseline after data cleaning: 3,8%

cleaning over time dependent informations
-> new users and movies are "outlier"
(in 2004 avg. more ratings, in 2005 less)
->eliminate "outliers" to create time independent model

"real" baseline: 7,8%

3

„ classical modeling approach [...]" – Neo Metrics

Paper 2 – Variables

User Variables:

- Number of historic user ratings
- Percent of 1-star ratings of the user
- Stdv of user ratings
- Number of months since the first rating of the user
- ...

Movie Variables:

- Number of historic ratings received by the movie
- Percent of 1-star ratings received by the movie
- Stdv of ratings received by the movie
- ...

User-Movie Interactions (after SVD):

- Likelihood of rating similar movies more than the mean
- Likelihood of similar users rating the movie more than the mean

„ classical modeling approach [...]“ – Neo Metrics

Paper 2 – SVD, Cluster Analysis

- SVD with user-movie-matrix with ratings

→ to group by users / movies in matrix

→ cluster analysis

„ classical modeling approach [...]" – Neo Metrics

Paper 2 – Machine Learning

- training data: sample of 2004 (sample method of 2007)
- Applied to 2005

→ Weighting of all variables

4 Conclusion / our possible Approach

- baseline in both papers very important

→ we cannot use such kind of baseline

- SVD used in two different ways

→ could also be important for us

- Item Item Similarity was less efficient

→ we think more efficient for us

- Machine learning

→ perhaps to weight our methods)

- Work with hierarchies→ for clustering
  - n songs of the same album rated
  - delete users from training data users with incalculable music taste?

Sources:

http://kddcup.yahoo.com/

http://www.cs.uic.edu/~liub/Netflix-KDD-Cup-2007.html#tasks - 26.04.2011

Su, Xiaoyuan & Khoshgoftaar, Taghi M.: "A survey of collaborative filtering techniques"

Miklós Kurucz, András A. Benczúr, Tamás Kiss, István Nagy,
Adrienn Szabó & Balázs Torma: "Who Rated What: a Combination of SVD, Correlation and Frequent Sequence Mining"

Jorge Sueiras: "A classical predictive modeling approach for Task "Who rated what?" of the KDD CUP 2007"

Miklós Kurucz, András A. Benczúr, Károly Csalogány:
"Methods for large scale SVD with missing values"

Book "???" of Arvid ☺ with explanation of SVD

http://de.wikipedia.org/wiki/Logistische_Regression - 24.04.2011

http://de.wikipedia.org/wiki/Gini-Koeffizient - 27.04.2011

http://en.wikipedia.org/wiki/Singular_value_decomposition - 26.04.2011