A combination of approaches to solve Task "How Many Ratings?" 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

Daniel Vélez
Neo Metrics
C/ Arequipa 1
28043 Madrid, Spain
+34 91 382 45 54
daniel.velez@neo-metrics.com

José Luis Flórez
Neo Metrics
C/ Arequipa 1
28043 Madrid, Spain
+34 91 382 45 54
jose.luis.florez@neo-metrics.com

ABSTRACT
This paper presents a solution to the KDD CUP 2007 task "How Many Ratings?". The combination of three different approaches is used to produce a final solution which improves the results obtained by each of these procedures by itself.

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

Keywords
Predictive modeling, forecasting, data mining.

1. INTRODUCTION
The KDD CUP 2007 task 2 is based on a competition proposed by Netflix (http://www.netflixprize.com). For the Netflix competition a training data set of more than 100 million ratings associated to user-movie pairs is provided [1]. This data was collected between October 1998 and December 2005. The aim of the contest is to estimate around two million ratings achieving an average prediction error lower than a prefixed value.

The purpose of this second task of the KDD competition was to forecast the number of ratings to be obtained during 2006 by 8.863 movies randomly chosen from the Netflix data set. An important constraint for this task was that only ratings given by users existing in the Netflix data file could be taken into account, that is, ratings of users registered in 2006 were not considered.

To accomplish such goal, three methodologies were developed:

- Memory-Based Reasoning Techniques: Producing an expected value for a movie in 2006 computed by a weighted sum of expected values in 2005 associated with movies showing similar behaviors.
- ARMA models: Adjusted to the time series defined by monthly movie ratings. Each of them was transformed to avoid being biased by new-user effects.
- The third methodology was a basic procedure which allowed estimation of rating percentages based on those registered for the previous year. All the movies whose ratings started in the same month were included in the same group.

None of these methods provided an optimal result, but their estimations, in addition to other factors, provided an interesting vector of exogenous variables for the construction of a final model.

Next, all methodologies introduced above will be described. Special attention will be paid to how the pronounced drop in the number of of new users at the end of 2005 was taken into account.

2. FIRST STEP: ANALYSIS OF NETFLIX RATING DATA
An important property of the Netflix data file is the drastic reduction observed in the number of users and movies which began to review or to be reviewed respectively, at the end of 2005.

The following figures (Figure 1 and 2) show both, the number of users and movies, relative to their starting month. In the first case (number of users), a gradual descent along the latest two months can be observed, while in the second (number of movies), the value is almost zero in the latest two months.

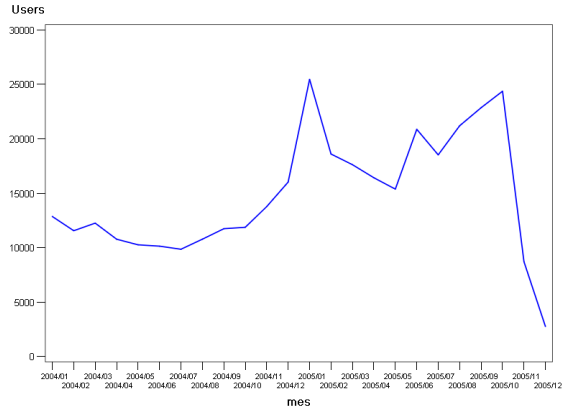

Figure 1. Number of new users by month.

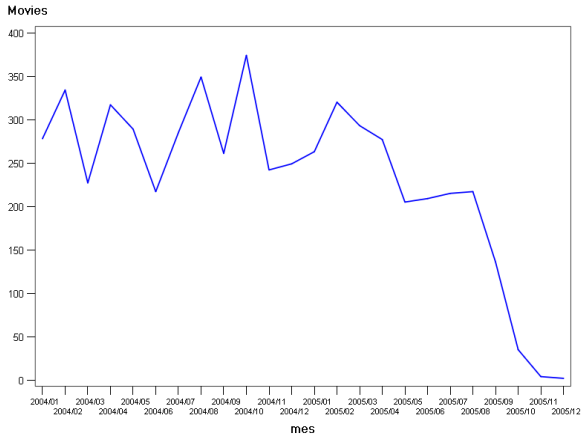

Figure 2. Number of new movies by month.

The effect of this drop is very important, since ratings in 2006 will be expected to decrease, not only because of the fact that 2006 new users will not be counted, but also because the number of users that appeared at the end of 2005 is considerably lower than expected.

To solve this problem, the number of users which should have qualified was estimated, comparing month by month new user percentages in 2005 and the average value associated to the 2002-2004 period.

Computing the difference between estimated and real data (stored in the Netflix training dataset), historical data were corrected, eliminating an identical percentage of new users in the final months of the previous years.

The next table (Table 1) shows the percentages of users which started their reviews within a month that were eliminated in order to emulate the behavior observed in 2005 according to the latest months of the analyzed period. This procedure was adapted for movies and applied to the KDD CUP 2007 task 1. A full description of it can be found in the paper associated to task 1.

Table 1. Percentage of users left out by month.

Starting month	Users
December	91.3%
November	68%

3. SOLUTION A: "K- NEIGHBOURS METHOD"

The K-neighbors method represent movies as vectors. The values contained in these vectors, make reference to the number of monthly ratings, excluding reviews given by users registered on the year associated to the month.

These vectors have been split in two parts:

- The first part is composed by 12 values, representing the evolution of the number of ratings given by users to each movie during year 2004.
- The second part is defined as the image of the previous vector. It was calculated as the sum of the ratings obtained by the movie in 2005.

Based on these vectors, the expected value for a movie in 2006 after a sequence of values S2004 representing the number of ratings in 2005, has been computed as a weighted mean of the images corresponding to the k vectors whose distances to S2004 are the smallest. The weights were defined in relation to the distances between S2004 and the vectors associated to its neighbors.

3.1 Algorithm used

Let $T_{Movie,Year}$ be the vector defined by the monthly ratings received by a movie throughout a given year:

$$T_{Movie,Year} = \left(T_{Movie,Year}(1),...,T_{Movie,Year}(12)\right)$$

Suppose we wish to compute the number of ratings Movie1 will obtain throughout 2006.

3.1.1 Step 1: Search for neighbors

Starting with the number of monthly ratings obtained by Movie1 in 2005, $T_{Movie1,2005}$, the k closest trajectories to Movie1 are considered by computing Euclidean distances with respect to numbers of movie ratings from 2004

$$d_j = \left\| T_{Movie1,2005} - T_{Movie1\_j,2004} \right\|_2 \quad \forall j \in \{1,2,...,k\}$$

3.1.2 Step 2: Prediction computation

The expected value for Movie1 in 2006, $P_{Movie1,2006}$, is established as the weighted mean of expected value sums for 2005 of the previous k trajectories from 2004, where weighting is established as a function of the distance vector:

$$P_{Movie1,2006} = \frac{\sum_{j=1}^{k} \frac{1}{d_j} * S_{Movie1\_j,2005}}{\sum_{j=1}^{k} \frac{1}{d_j}}$$

where

$$S_{Movie1\_j,2005} = \sum_{i=1}^{12} T_{Movie1\_j,2005}(i) \quad \forall j \in \{1,2,...,k\}$$

To finish with this methodology, some comments related to cautionary measures taken are presented:

- Search for trajectories neighboring a given one for a specific year is performed by looking for similar trajectories from the previous year. Since trajectory variability increases with time (heteroskedasticity effect), it was decided to perform a prior logarithmic transformation on the trajectories in order to enable comparisons among them. In addition, applying this transformation was consistent with the error measure used for model quality assessment.

- Both the highest and the lowest image were suppressed in the estimation of the weighted mean in order to avoid possible "bad influence" of movies with extreme images.

Finally, the forecast error achieved by this method over the scoring data was 0.5828.

4. SOLUTION B: "ARMA METHOD"

This approach adjusts ARMA models [2] to as many monthly rating time series as movies. Once again only ratings given by users registered before the beginning of the year were taken into account.

All the series were smoothed, eliminating the trend effect caused by the annual increase of users and correcting monthly level shifts. The user registration effect was suppressed by computing annual factors as the ratio between existing users in the considered year and users existing at the end of 2005. Monthly level shifts were corrected by using monthly factors which allowed an increase in the number of ratings in months where this quantity had been lower or decreased otherwise.

In a last stage, ARMA models were adjusted to these smoothed series. The procedure was only applied to movies over one year old.

The ARMA model finally chosen for the fit was an ARMA(1,1):

$$(1-\phi B)\log(X_{Movie,t}) = (1+\theta B)\varepsilon_{Movie,t}$$

where $X_{Movie,t}$ defines the time series associated with the number of rating obtained by the movie within a time unit (a month), appropriately smoothed out as described, and B is the lagging operator.

In this case, the forecast error achieved over the scoring data was 0.9485. This method's poor performance is probably due to generally short data histories available for series construction.

5. SOLUTION C: "FALLING CURVES METHOD"

The third method is quite simple and does not use any mathematical model. The monthly behavior of ratings corresponding to different movies was analyzed, excluding ratings given by new users registered on the year associated to the month under analysis.

For instance, next figure (Figure 3) shows monthly average percentage of ratings during 2005 for movies that began to be rated prior to January 2005, by users registered prior to January 2005. The noticeable decreasing trend shown at the end of Figure 3 gives its name to this method.

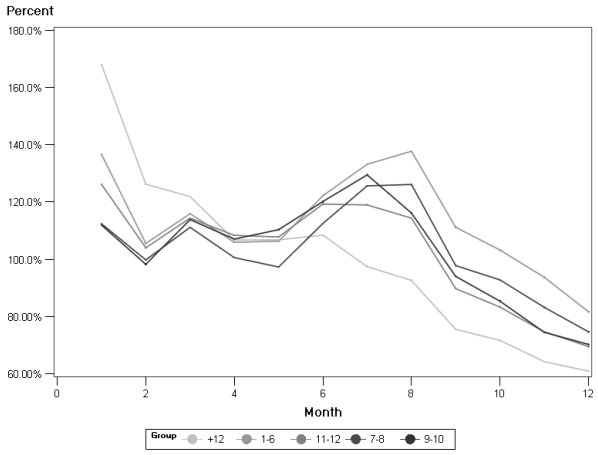

Figure 3. Falling curves applied to each one of the five groups of movies.

The previous observation led us to the following hypothesis: "The way in which the ratings of a movie decrease in the following year is similar for movies with a similar first review date". So, five groups of movies were distinguished depending on their ages (the age of a movie is defined as number of months since its first review):

- Under six-month-old movies.
- Seven or eight-month-old movies.
- Nine or ten-month-old movies.
- Eleven or twelve-month-old movies.
- Over one-year-old movies.

For each of these groups, the respective percentages of ratings for the next twelve months was computed according to the behavior observed in 2005. These percentages were used to estimate the number of reviews expected in 2006.

The forecast error obtained with this basic procedure over the scoring data was 0.9001.

6. FINAL MODEL

The definitive model was built as follows:

1. Netflix data set was split into two subsets. The first one contained movies whose ratings were made prior to January 2005, while the other included the remaining ones.
2. In the first subset, ratings given by users who began their reviews in the latest months were excluded just as it has been previously explained. In addition to the predictions given by each one of the mentioned solutions, this subset was used to generate different input variables.
3. The second subset was used to count the ratings given in 2005 by the users existing in the first subset. The

resulting values (x) were converted by logarithmic transformation (log(x+1)).

4. A data set with input and target variables associated to approximately 14.000 movies which began to be reviewed prior to January 2005, was obtained by joining these subsets. The resulting data set was then partitioned in two tables of equal dimensions which were considered as training and testing data for the final model.

The following variables were included in that model:

- Log(x+1), where x is the rating forecast provided by k-neighbour method for 2005.
- Log(x+1), where x is the rating forecast provided by ARMA method for 2005.
- Log(x+1), where x is the rating forecast provided by falling curves method for 2005.
- Log(x+1), where x is the total number of reviews given to a movie since it was registered.
- Number of months since first review.
- Percentage of low scores ('Stars = 1') given by users.
- Percentage of high scores ('Stars = 5') given by users.
- Average rating of the movie.
- Standard deviation of ratings associated to the movie.
- Number of months since last review.
- Percentage of reviews given in the last year.
- Percentage of reviews given in the last three months regarding the reviews given in the last year.
- Ratio between reviews of the last year and the previous one.
- Percentages of reviews given by users who have been scoring for more than a year.

The final model implemented consisted in a neural network model with perceptron architecture [4] and a hidden layer integrated by 5 nodes. The forecast error achieved over testing data with this model was 0.49. Given that the final error over the scoring data was 0.5227 the model was considered to be "overfitted". The reason for this could be the reduced number of movies (around 7.000) that took part on the training.

To end with, two graphs were produced. The first one was residual against forecast using the training set and the second one residual against forecast using the scoring set (Figures 4 and 5). Figure 4 shows that the residuals have been properly adjusted in the first case. However, Figure 5 shows that movies with the highest levels of ratings have been overestimated over scoring data.

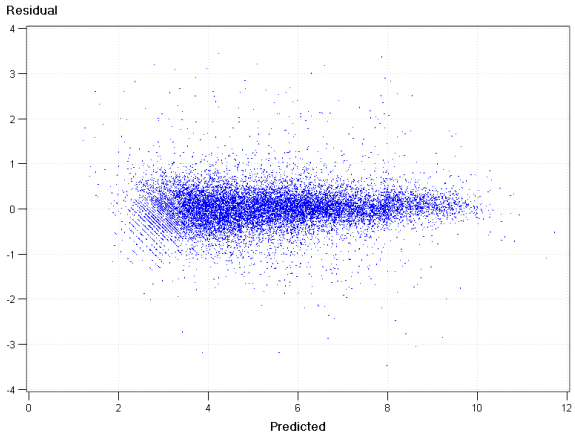

Figure 4. Residual vs. Predicted over training data.

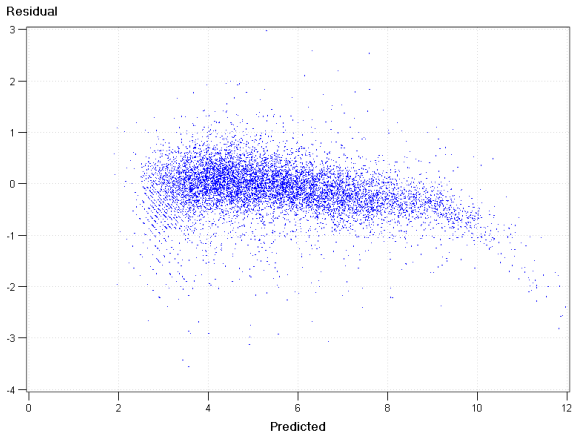

Figure 5. Residual vs. Predicted over scoring data.

7. CONCLUSIONS

In this paper we present our solution to the KDD CUP 2007 task two. In our opinion, there are two basic issues that must be considered in order to achieve a good solution to the problem.

The first of them was the fact that possible reviews given by new users were not counted. Moreover when these users were the ones who gave a higher number of ratings. Because of this, monthly ratings show an obvious decreasing trend that must be taken into account. The second issue was that the entry data set must reflect the noticeable fall of new users at the end of 2005. Overlooking these effects, would greatly increase the average error achieved by the approaches reviewed.

Different mixed models were built by combining predictions resulting from some of the three outlined methodologies. We observed that the absence of any of the predictions would have generated a result considerably worse than the solution finally submitted.

8. ACKNOWLEDGMENTS

We are very grateful to Neo Metrics for the support they have given us since the beginning of this project. In particular we are in debt with Ana Alvarez, Natalia Molina, Maria Sala and Maria Sanchez for their efforts put forth in order to solve this task, and Juan-Carlos Ibañez and Fausto Morales for his help in writing this paper. We would also like to express our thanks to the KDD CUP organizers for the work they have carried out.

9. REFERENCES

[1] J. Bennet and S Lanning. *The Netflix prize*. KDD Cup and Workshop 2007, San Jose, California, Aug 12, 2007

[2] G. E. P. Box and G. M. Jenkins, editors. *Time Series Analysis: Forecasting and Control.* San Francisco: Holden-Day, 1970.

[3] E. Castillo, A. J. Conejo, P. Pedregal, R. Garcia, and N. Alguacil. *Building and Solving Mathematical Programming Models in Engineering and Science.* Pure and Applied Mathematics: A Wiley-Interscience Series of Texts, Monographs and Tracts, 2001.

[4] R. O. Duda, P. Hart, and D. G. Stork. *Pattern classification.* Wiley, New York, 2001.

[5] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer, 2001.