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 were rated before the defined time range.
2. Choose by simple random sampling with replacement 200.000 ratings and save the corresponding user ids, from the fixed time range. That is, draw with replacement a 200.000 sample of users, with probability proportional to the number of ratings per user.
3. Repeat the previous step but keeping the id_movies. That is, draw with replacement a 200.000 sample of movies, with probability proportional to numbers of times a movie was rated.

4. Join both lists randomly (id_users and id_movies) and throw away any duplicated pair.
5. Throw away any (id_user, id_movie) pair that already existed in the historic data.
6. Keep the first 100.000 pairs.

The direct application of this algorithm to the time range of 2005 provides a rating of 1's of around 20%. But a closer look to the data shows that most of those ratings belong to either new users (those whose first rating was at the end of 2004) or new movies (those who were first rated at the end of 2004). Therefore these two issues must be taken into account: proportion of new users and movies that is present in the data prior to the given time frame. In fact the proportion of new users and movies at the end of 2005 is much lower than the one at the end of 2004, Therefore a lower baseline is expected for the 2006 period.

This result shows that the model under construction is time-dependent (on the specific year), since there are practically no ratings from new users and new movies in 2005, by contrast to 2004 and preceding years, where many ratings from new users and new movies could be found. This fact makes it necessary to search for a procedure to make model results independent from the year of application, so as to correct the previous baseline. This solution is simpler than to build a time dependent model.

For this reason we decided to clean the data used to estimate the baseline and therefore we avoid the possible bias produced for those users and movies that appeared at the end of 2004. The following percentages of movies and users by first appearance month were eliminated. These percentages are obtained by comparing the average monthly percentage of new users in 2002, 2003 y 2004 with the percentage of new users in 2005, and dividing for each month. .

Table 1. Percentage of users and movies left out by month.

Month	Users	Movies
Dec	91.3%	100%
Nov	68%	100%
Oct	0%	90%
Sep	0%	43%

Table 1 shows the percents of movies and users eliminated from the data used to estimate the baseline.

Applying the algorithm described in the previous section to the cleaned data we obtain a baseline between 4.5% and 5% for 2005. Note that these values are more realistic than the initial 20%.

The algorithm was applied to the years 2003, 2004 and 2005. Then a linear model was used in order to forecast those estimated baseline using information from the previous years. The final model is quite simple and predicts the baseline for the next year as a linear combination of the percent of new movies in the current year and the percent of new users in the current year. The application of this model provides an estimated baseline for 2006 of 3.8%. Note that the real rating obtained once the scoring data was released was 7.8%, but a wrong baseline of 20% could have been used if the baseline estimation is not performed.

3. MODELING APPROACH

Our modeling approach tries to reproduce the scoring task, that is, to predict the rating events during 2006 based on information recorded up to 2005.

To do so, the training data was divided in two pieces. The first part (ratings with date prior to January 1, 2005) was used to create the input variables. The second part (ratings with date posterior to January 1, 2005) was used to build the target.

In order to reproduce the sampling scheme used to create the scoring data, the sampling algorithm introduced in the previous section was used. A training dataset of 500,000 samples was created by iterating the sampling algorithm five times. Therefore, the target variables in the training data consist on 500.000 binary events (1 if the user has rated the given movie during 2005 and 0 otherwise).

Additionally, in order to correct for the issue described in the previous section, the training data was cleaned by eliminating those users and movies that first appeared at the end of 2004; using the percentages shown in table 1. This step is necessary; if omitted, definitions of input variables built for the training dataset in 2004 would differ from those for the scoring dataset in 2005, as they would be built on two scoring histories that would be quite different in nature – Influx of new users and new movies is quite different in either case.

Once training dates were cleaned, 250.000 out of the 500.000 observations were left out for test purposes and the remaining observations were used to build the predictive model. The model used is a logistic model built over '*intelligent variables*'. Each *intelligent variable* was created by first summarizing the raw input information and then transforming it into a discrete variable using supervised decision trees. The relevance of these discrete variables is critical, they add nonlinearities, and saturation effects to the model and are constructed in order to maximize its predictive power.

Once cleaned, the training data was used to build three types of explicative variables:

- Movie variables.
- User variables
- User-Movie interactions

These variables are described in detail below.

The selected mathematical model form is a logistic regression [2]. In parallel three other types of model formulations were analyzed but they all yielded poorer results on the test data:

- Boosting algorithms over decision trees each of them having no more than 8 leaves [6].
- Neural Network Multilayer Perceptron with a six neurons hidden layer [7].
- Arcing techniques on the model above [7].

En la siguiente tabla se compara el rendimiento según el RMSE de de los modelos anteriores y de la baseline estimada y real sobre los datos de score proporcionados para la tarea.

Table 2. Most important variables under the final model.

Model	RMSE
Baseline estimada 3.8%	0.271
Baseline Real 7,8%	0.268
Boosting algorithms over decision trees each of them having no more than 8 leaves	0.265
Neural Network Multilayer Perceptron with a six neurons hidden layer	0.268
Arcing techniques on the model above.	0.268
Logistic regression (final model selected)	0.263

Figure 1 shows the expected captured response computed using the test data. Given this figure, we were expecting a 10% improvement over the estimated baseline which was 0.038. The scoring data showed a 6% improvement over our baseline.

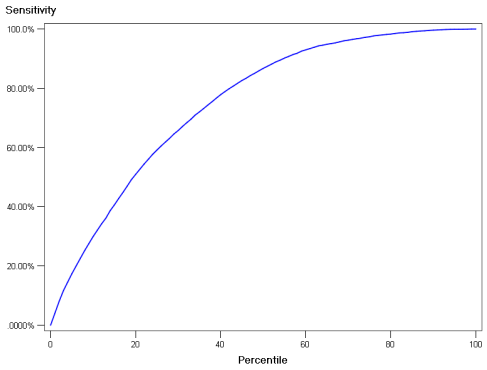

Figure 1. Model sensitivity.

4. MOST SIGNIFICANT VARIABLES

Here we described the most significant variables considered. These variables can be classified into 3 categories:

- User related variables. Variables built by focusing on the historic rating behaviour of each user.
- Movie related variables. Variables built by focusing on the historic rated behaviour of each movie.
- User-movie pair variables. Variables built by focusing on the common historic behaviours of users and movies.

4.1 User related variables

- Number of historic user ratings.
- Number of next year user ratings estimated using a decay curve (see the task 2 paper).
- Number of months since the first rating of the user
- Percent of 1-star ratings of the user.
- Percent of 5-star ratings of the user.
- Average rating of the user.
- Standard deviation of the user ratings.
- Number of months since the last rating of the user.
- Percent of ratings of the user during the last year.
- Percent of ratings of the user during the last three months.
- Percent of ratings of the user during the three last months over ratings during the last year.
- Ratio between number of ratings during the last year and number of ratings during the year before.
- Percent of ratings of the user to movies that have been rated for more than one year.

4.2 Movie related variables

- Number of historic ratings received by the movie.
- Number of future ratings the movie will receive next year, estimated using a decay curve (see task 2 paper).
- Number of months since the movie was first rated
- Percent of 1-star ratings received by the movie.
- Percent of 5-star ratings received by the movie.
- Average rating received by the movie.
- Standard deviation of the ratings received by the movie.
- Months since the last time the movie was rated.
- Percent of ratings received by the movie during the last year.
- Percent of ratings received by the movie during the last three months.
- Percent of ratings received by the movie during the last three months over ratings received during the last year.
- Ratio between the number of ratings received during the last year and the number of ratings received the year before.
- Percent of ratings coming from users that have been rating for more than year.

4.3 User-movie pairs

These variables intend to add cross-section information; that is to include information on user and/or movies that behave similarly. In order to build these variables we proceed to identify groups of users which might have similar rating behaviours and also groups of movies that were rated in similar ways.

The algorithm for the identification of these groups is as follows:

1. Create the (n by m) "Ratings Matrix", $\boldsymbol{X}$, which contains the ratings of n users for m movies (prior to January 1st 2005).
2. Perform a Singular Value Decomposition [3] on $\boldsymbol{X}$, and get the matrices $\boldsymbol{U}$ (n by c) and $\boldsymbol{V}^*$ (c by m), where c is fixed to 300, a value with which a sufficiently high percentage of explained variance was obtained.

3. Take the first *i* columns from matrix ***U*** and apply a *k*-means cluster analysis on the new (*n* by *i*) matrix. The cluster algorithm is run to generate *j* groups. This task is repeated for *i = (20, 40, 80)* and *j = (100, 300, 1000)*. So nine ways of classifying the Netflix users are obtained.

4. In the same way a cluster analysis is performed on the first *i* columns of ***V**** producing *j* groups. This task is repeated for *i = (20, 40, 80, 150)* and *j = (100, 300, 1000)*. So 12 ways of classifying the Netflix movies are obtained.

A user-movie pair variable is created utilizing the defined user groups. The variable is defined as the ratio between the percentage of users inside the user's group that rated the movie and the overall percentage of users that rated the movie.

In a similar way the movie groups are used to create another user-movie pair variable, defined as the ratio between the percentage of movies in the movie group rated by the user and the overall percentage of movies rated by the user. This last variable has been proven to have a great predictive power.

Table 2 lists the most important variables in descending order, according to the Gini index. Note that the used variables are discrete transformations of the described variables.

Table 3. Most important variables under the final model.

Variable	Gini
Expected number of ratings for next year, estimated using the model developed for task 2	0.4643
Likelihood of rating similar movies more than the mean, 500 groups and 40 variables.	0.3229
Percentage of user ratings corresponding to movies with more than 1 year	0.2870
Percent of 1-star ratings given to the movie	0.2774
Likelihood of similar users rating the movie more than the mean, 1000 groups and 40 variables.	0.2618
Average rating received by the movie	0.2589
Percent of ratings received by the movie in the last three months.	0.2529
Percent of 5-star ratings received by the movie.	0.2320
Expected number of user ratings for next year	0.2053
Percentage of ratings received by the movie during the last 3 months over ratings received during the last year.	0.1975
Percentage of movie ratings in the last year.	0.1772
Number of months since the movie was first rated	0.1548
Percentage of user ratings in the last three months	0.1263
Percentage of user ratings in the last year.	0.1141
Number of months since the first user rating	0.1074

5. CONCLUSIONS

We believe that a great amount of our success relies on the work developed to correctly estimate the baseline model. Although the absolute error of our estimated baseline was over 4%, (3.8% against 7.8%) we must remember that the raw data has a 20% for the year 2005.

On the other hand the variables included are also critical. We emphasize three ideas about the variables:

First note that the model and variable selection is done using the training data previous to 2005, but the final scoring must include the 2005 information. The key point is to correct the raw data so the information up to December 2004 is similar to the whole training data. This is done via the cleaning and sampling methods described in sections 2 and 3.

The second key point is the fact that the most important variable in the model is the output of task 2. The brilliant qualification as first runner up obtained on task 2 has allowed us to use a quality variable available for task 1.

Third, the use of user-movie pair variables. These variables are very important and can be used to solve both task 1 and the original Netflix problem.

- The chances of a user rating a movie are greater if that user tends to rate similar movies more than the mean, where similar movies stands for movies that are rated or not rated in a similar way by the users.
- The chances of a user rating a movie are greater if that movie tends to be rated by similar users more than the mean, where similar users stand for users that rate or not rate movies in a similar way.

Finally we would like to state that we are not currently working on the Netflix problem. Although we have performed the experiment of applying an adapted version of the procedures described in this paper to the Netflix problem. Basically a model with the same inputs have been built but with a multinomial target (ratings ranging from 1 to 5). The result on the 'probe' data gave as an error of 0.958; Still above baseline provided by Netflix and very far from the leading positions at the leaderboard.

6. 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 David Arias, Tania Cámara, Jesus Figueres, Penélope Garzón and Paz Gil-Delgado for the effort put 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.

7. REFERENCES

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

[2] P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice Hall, London, 1982.

[3] Abdi, H. Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD). In *N.J. Salkind (Ed.): Encyclopedia of Measurement and Statistics.* Thousand Oaks (CA): Sage. 2007

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

[5] R. Salakhutdinov, A. Mnih and G. Hinton Restricted Boltzmann Machines for Collaborative Filtering. Machine Learning. In *Proceedings of the 24 th International Conference*, Corvallis, Oregon, USA. ACM Press, 2003, 791-798.

[6] Schapire, R. E., and Singer, Y. Improved boosting algorithms using confidence-rated predictions. In *Machine Learning 37*, 1999, 297-336.

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