KDD Cup 2007 – How often will that movie be rated?

James Malaugh
Inductis
571 Central Ave, Suite 105
New Providence, NJ 07974
908-743-1181
jmalaugh@inductis.com

Sachin Gangaputra
Inductis
30 Broad St, 41st Floor
New York, NY 10004-2304
908-743-1100
sgangaputra@inductis.com

Nikhil Rastogi
Inductis
2nd Floor, Tower B, Vatika Atrium
Sector-53, Main Sector Road,
Gurgaon, Haryana 122001, India
91-124-432-1700
nrastogi@inductis.com

ABSTRACT
In this paper, we describe the process by which we came up with our solution to the task 2 problem of the KDD Cup 2007 competition, which was founded in close connection with the ongoing Netflix prize competition. This task asked competitors to predict the number of times a defined set of movies would be rated by a set of Netflix customers in 2006 based on the historic data in the entire Netflix database up to 2005. This historic data included when and how, on a scale of one to five, a customer rated a movie. Though we did not have to predict actual distribution of ratings in this task, just the actual number of ratings, this rating information was a core predictive input. Solution performance was based on the RMSE between the following pair of transformations on the raw prediction for number of ratings in 2006, X, and actual number of ratings in 2006, Y: ln(X+1) and ln(Y+1). For our submitted solution we discuss the approach we adopted including data set-up and manipulation, exploratory techniques, variable creation, and the final estimation and predictions. We also briefly discuss some other approaches that were tried and ultimately discarded as they did not compare favorably to the approach we finally submitted.

Categories and Subject Descriptors
G.3 PROBABILITY AND STATISTICS: *Correlation and regression analysis*, *Nonparametric statistics*, *Statistical software*, *Time series analysis*

General Terms
Algorithms, Theory.

Keywords
Analytics, data mining, modeling, predictive analytics, KDD Cup, Netflix prize competition, meta-modeling, clustering, fuzzy clustering, segmentation, bootstrapping

KDDCup'07, August 12, 2007, San Jose, California, USA.

1. INTRODUCTION
When the Netflix problem first came out we formed a team to investigate the problem and see what our methods could do with this problem. Little did we realize that it was also to become the focal point of this year's KDD Cup. When the problem statement for this year's Cup was announced we soon came to the conclusion that task 1 was similar in nature to the original Netflix challenge problem, while task 2 was perhaps more accessible to an array of approaches. After moderate successes during the Netflix challenge we felt some techniques more familiar to us, along with some good application of the business problem insights, would perform better than the approaches we had employed during the Netflix challenge. These approaches were concentrated in machine learning and pure data mining with little business problem context. We felt the set up for task 2 would make the application of business problem solving, along with technical and rigorous analytics, much more valuable. The resulting model performed well and we discuss the elements of our solution below.

2. Data Set-Up
From the onset our strategy was to try out several approaches and methodologies in order to come up with our single best shot for the final submission. As we had no strong inclination to one approach over another we made a conscious decision very early on to set up a validation process that would mirror the final submission scoring as close as possible. We would then put a lot of weight on this internal performance measure over any other when deciding which approach to finally use.

Task 2's scoring set-up is basically that you get to use 2005 and earlier ratings data in order to predict the number of ratings in 2006 for each movie. Hence, we decided to build our models with data up to and including 2004, and validate them on 2005 data that we had in hand. Of course, this removed some movies and customers from our sample that were not released into the Netflix database until 2005. However, we felt the confidence the out-of-time validation metric would give us in the approach we would finally choose would be worth it. Once we had determined our best approach on the 2003-2004 dataset we then "rebuild" the approach with the 2005 shifted into the modeling set and score the 2006 data for submission. This out-of-time validation gave us confidence that the final process would provide the most stability and best performance on the final scoring set. We were pleased to see this pay off when our final submission's RMSE was only a

couple of hundredths greater than what we had forecast with our 2005 out-of-time validation set.

Once our set-up was complete the next decision involved the layout of the set. Do we use a single cross-sectional slice from 2005 with predictors incorporating some historical information? This approach would entail having a single observation per movie, with variables designed to capture the information about the movie's historic ratings. Or do we set-up the data for an annual or monthly time series methodology? The answer to these questions actually differed depending on the approach that was tried. Our final submission was based on an unbalanced panel data approach where each movie had multiple observations in the data set based on the number of ratings the movie received in 2003, 2004, or 2005. For example, a movie that first became available in the Netflix database in 2004 would have two observations in the dataset, one with the dependent variable as the number of ratings in 2004 and the second where the dependent variable was the number of ratings in 2005. For this movie we use the terminology "Observation Year 2004" and "Observation Year 2005" to refer to the separate observations, respectively. A movie put into the Netflix database in 2001 would have three observations, one for each of the years 2003, 2004, and 2005. An illustrative layout for such data can be found in Table 1 below. We did not include observations for number of ratings in 2002 or earlier, as the smaller number of movies available for rating and smaller customer base seemed to add more complication than possible resulting benefit. In this set up, each observation required that any predictors whose definition depended on time related factors to be appropriately tenurized. For example, for a certain movie, the *time_since_first_release* variable could be *x* for its 2003 observation, would be *x+1* for its 2004 observation. This is also illustrated in Table 1.

Table 1. Illustrative combined 2004-2006 Modeling Data

Movie ID	Observation Year	Release Year	Number of Ratings	Time Since First Release
99999	2004	2004	2,304	0
99999	2005	2004	11,452	1
010101	2005	2005	34,184	0
55555	2003	2001	18,342	2
55555	2004	2001	17,983	3
55555	2005	2001	13,591	4

As a contrast we note that within our testing set-up, in which we used a combined 2003 and 2004 dataset to predict number of 2005 ratings, the unbalanced panel data modeling set contained either one or two observations per movie.

3. METHODOLOGY

Once our data was arranged we set about deriving as much insight as possible from it. This involved an iterative process of exploratory analysis, variable creation, model estimation, and performance analysis. As the performance measure was to be taken on the logged transformed space we looked at the log transformation of the dependent variable for most analyses.

3.1 Variable Creation

Variable creation was driven mainly from exploratory data analysis but included some business problem driven hypotheses also. Initially, we created general time and rating related variables and then used them to create other variables, including certain cluster derived variables and segmentation dummies. Though we derived many variables to be utilized in subsequent estimations, most of these variables fall into one of the following groups: movie specific, movie-customer specific, movie-cluster specific, movie-neighbor specific, and segmentation variables and dummies. Through certain distribution and bivariate analyses we also determined variables on which to apply appropriate transformations, such as translations, banding, or dummy creation.

3.1.1 Movie Specific

This category accounts for variables created based on a single movie's characteristics. Some simple examples include: year of movie release, length of movie title, historic number of ratings, average rating, average number of ratings over time, and movie tenure. Certain variables were also annualized for records where the observation year coincided with the year of release. These variables would only differ from their corresponding non-annualized versions for the observation in which in observation year and movie release year coincided. For example, for a movie released in September 2003, its data point corresponding to the number of ratings in 2003 would have different values for number of ratings that year and number of annualized ratings that year. The 2004 observation would see no difference in these variables in that record, as would the 2005 observation.

3.1.2 Movie-Customer Specific

This category includes variables that require more knowledge of the customer base as a whole, and how the customer base relates to a particular movie. One example is the concept of an "available customer". For each movie observation, this variable takes the value of the number of current customers in the database that have not yet rated the movie. Another set of variables is derived from a segmentation of customers based on their tenure. This segmentation separates customers that have recently been added to the database from older customers. A movie is then described by a set of variables defined by using the raw number and percentage of customers from each of these segments that have rated that movie. These variables created the base for other types of variables by interacting with other movie-customer and movie-specific type variables. We found these variables important as a way to describe portfolio dynamics not captured by some of the base variables.

3.1.3 Movie-Cluster Specific

The movie cluster variables we derived were a holdover from our earlier experience in the Netflix prize competition when we were investigating more machine-learning oriented techniques. The premise is that we can approximate an individual movie's behavior by analyzing the behavior of a set of movies we deem as being "similar" to that movie. What you deem as the set of "similar" movies is dependent on the techniques and metrics used. In our case, we fixed a data year, and set up a customer by movie matrix with a 1 if that customer had rated a particular movie in the past, 0 otherwise. The common least squares metric was chosen to

define distance between movies in this matrix set-up. A clustering was performed with some tweaking of initial seeds, number of iterations, cluster diameters, and cluster sizes. We found these clusters to be fairly stable regardless of these tweaks however. The clusters could have been re-defined for each observation year, reclassifying some 0's as 1's as more customers rated movies over time and re-running the cluster analysis. However, after some investigation we found movie clusters to be fairly stable regardless of the observation year chosen. Hence, for simplicity we stuck to one clustering. Table 2 below illustrates a random sample from an output cluster we dubbed "Horror Movies".

Table 2. Horror Movie Cluster

Movie ID	Title
4949	House on Haunted Hill
16516	Jeepers Creepers
14528	Eight Legged Freaks
14126	Fear Dot Com
1794	The Rage: Carrie 2
14541	Dracula 2000

These clusters were subsequently used to create movie-cluster specific variables, including versions of the movie-specific variables. For example, average rating and average number of ratings for the all the movies in a particular movie's cluster.

3.1.4 Movie-Neighbor Specific

This approach is a slight alteration on the clustering approach we mentioned above. It involves computing inter-movie distances between pairs of movies based on the overlap of customers who have rated that movie. The definition of the inter-movie distance $D(m_1,m_2)$ follows:

$$D(m_1,m_2) = | C(m_1) \cap C(m_2) | \; / \; | C(m_1) \cup C(m_2) |$$

Here, $C(m_1)$ and $C(m_2)$ represent the set of customers that have rated movies m_1 and m_2, respectively. Two movies that have been rated by the same set of customers will receive an inter-movie distance of D=1. Inter-movie distances are computed for all movie combinations. For each individual movie, the ten movies closest in terms of this metric are defined as its "neighbors". Variables are then defined off the group as a whole. Unlike in clustering where the sets of movies are disjoint, each movie has its own set of ten nearest neighbors which may overlap or coincide with another movie's set of neighbors. Hence, in a sense it is a form of fuzzy clustering where each observation gets to be the seed of its own cluster.

This approach paid major dividends for predicting the number of ratings for movies that were recently introduced into the database. Some of our earlier modeling attempts were being thrown off by poor performance in this segment of the movies. The introduction of these movie neighbor specific variables brought performance in this segment more into line with the models' performance on other movies with longer histories. This suggests that in order to make the most accurate ratings predictions a balancing act between the use of a movie's own ratings history, and those of its neighbors' must be performed. And the tipping point of such a balancing act is a function of the amount of time the movie has been in the database. Movies with a short history can rely more on their neighbors' ratings information, which likely has more historical richness, while a movie with a longer history can rely more on its own historical information.

3.1.5 Segmentation

A variety of segmentations were run to predict the number of ratings for each movie using predictor variables that have been described so far. The results of these segmentations were sometimes taken as standalone models. These, in general, did not give enough granularity to our results. We also used these segmentations to derive input predictor variables for other estimation schemes such as regressions. This was done by creating segment dummies. These are binary variables for individual segments which indicate whether or not the movie fell in those segments.

We also used the segmentations as a base to build different regression models on. In this segmented approach, regression models are built on the individual segments of the population as opposed to the entire population. This can be useful if the segmentation really separates out pockets of the population with inherently different behavior. If this is the case, then the regression models derived on the individual segments need only capture the behavior for this group and not need to try to capture all behavior for the population simultaneously. The benefits of such an approach are diminished if the original segmentation does not really derive segments that are different enough from each other, or if the resulting combination of models leads to over fitting the modeling set, resulting in poor validation.

3.1.6 Transformations

Once a healthy set of candidate predictors had been derived from the previously discussed techniques some additional analyses, including univariate, bivariate, and multivariate analyses, were performed. The results of these analyses were then used to perform a variety of transformations, such as translations, banding, and logarithmic transformations, to name some. Banding is the process of breaking a numeric variable, most often continuous, into a set of dummy variables, in an attempt to capture non-linearities between the continuous predictor and the dependent variable. At a high level, regression estimation techniques force a single best-fit line through any single predictor, albeit in a simultaneous estimation. This often is inadequate, and at times misleading if there is a high degree of non-linearity in the predictor-dependent relationship which cannot otherwise be accounted for by other variables in the regression. An example of a transformation that did prove useful in making predictions more accurate was a transformation relating to some movie-customer specific variables. By multiplying variables by a simple ratio of size of customer base in one year to size of customer base in another, many variables could have their raw magnitudes normalized so that there was more of an apples to apples comparison across observation years. In theory, this attempted to factor out any signal that may be due solely to the increase in customer base size over time.

3.2 Estimation

After the variable creation stage we had a variety of options in order to arrive at the final number of rating predictions. In each individual regression model we went through a rigorous variable selection process using re-sampling schemes and careful analysis

of output diagnostics. Variable selection diagnostics mainly involved analyzing p-values and variance inflation factors (VIFs). The resampling was done mainly to improve model stability and ensure no one sample's random variance dominated model selection. In the end we performed some meta-modeling and averaged the results from a linear regression on the entire population, a twenty-two segment regression ensemble, and a single best standalone segmentation. The out-of-time RMSE for this model on 2005 data was 0.52.

Our goal all along was to create a stable model building process that worked well on our out-of-time validation set-up. When we were happy with this process, we had to include the 2005 data in our modeling set and go back and rebuild, not just re-estimate, the models in order to score the final 2006 data. The set-up and careful variable selection schemes were performed in an attempt to make this extended building process more stable. We were pleased when our scoring RMSE came back at 0.5406, just a couple of hundredths above our 2005 out-of-time validation of 0.52.

For direct comparison of some different approaches we have included Table 3 below. This table compares the in-sample modeling RMSE from the combined 2003-2005 dataset for a few of the previously mentioned approaches. The two top approaches were regressions on the entire population. The first was done with 80% resampling meaning multiple 80% samples were drawn from the modeling set and regressions estimated on each in order to arrive at the final variable list. The "22 – Segment Regression" approach was where we built individual regression models on each of twenty-two segments of the population derived from an earlier segmentation scheme. Similarly, the "3 – Segment Regression" approach was based on building regressions on a three segment segmentation scheme. In this case the segments were based on a segmentation of the number of ratings a moving received. Hence, the segments could be described as rated often, rated an average number of times, and rated infrequently. Finally, the "Single Segmentation" was the best standalone segmentation where we allowed the prediction to come directly from the segmentation.

Table 3. 2003-2005 Modeling RMSEs

Approach	2003-2005 Modeling RMSE
Regression (80% resampling)	0.49
Regression (10% resampling)	0.5
22 - Segment Regression	0.48
3 - Segment Regression	0.50
Single segmentation	0.56

We believe that the similarity in results across methods gives some credence to the hypothesis that the real work being done here is by the variables themselves. It seems that much of the useful information has been captured in these variables and the actual estimation method chosen, at least among those we tried, adds little marginal performance. This is why we settled on a combination of these methods as a final answer, as we felt it may add just a little bit of stability overall.

4. OTHER APPROACHES

Several approaches were attempted before settling on the approach described above. Some alternatives involved different data set-ups where we were not using incomplete panel data. Other approaches just involved applying different strategies to any of the data set-ups. We spent more time on some approaches than others. In brief, we explored options in the following areas at some point: time series, autoregressive models, hazard rate models, and general linear models. For the time series approaches we did analysis at both the annual and monthly level, and at both the movie cluster level and individual movie level. Some of these performed very poorly and others actually performed well, especially on out-of-sample and within time validation methods. However, they did not hold up on the out-of-time performance measure. Since, as we mentioned at the beginning, we were searching for out-of-time stability we passed on these approaches in favor of the one discussed above.

We also at one point looked into building our own database of external movie information, including variables such as movie genre, actor names, directors, critic ratings etc. However, given policies regarding approved data usage, and the effort required to manually create such a database with a reasonable coverage on the movie set in question, we decided it would not be wise to spend our resources on this undertaking. We did, however use a coarse variable regarding movie genre that we created manually. Movies that did not have coverage ended up going into their own category. This variable ended up being used in several places, such as in defining segmentations, but was not as important as movie clusters, which were practically more granular versions of this genre variable.

5. CONCLUSION

We set out with the mindset that we would not limit ourselves to a single approach that we would tweak to perform as well as possible. As opposed to throwing the metaphorical kitchen sink at the problem, we feel with the structure we placed on the process we were able to narrow down to an approach which loaned itself well to the problem and was the most stable among the alternatives tried. We also found that the injection of some business problem insight, especially in the variable creation phase, really helped to explain some of the difficult portfolio dynamics. Not surprisingly, if we had more time there were a couple of things we would have liked to try to tweak the end product a little, but we were happy with the results overall. And there is always next year's Cup!

6. ACKNOWLEDGMENTS

We started with a small team focused on task 2 of the KDD Cup. As Inductis is a managing consultant firm, client work demand drew people both in and out of the team over the course of our endeavor. Besides the authors, other team members included Rahul Shankar, Neha Gupta, Sandeep Gupta, Kushagra Gupta, and Gaurav Lal. The team thanks Krishna Mehta, our methodology head, for allowing us the time to work on such an interesting project.

We would also like to thank the KDD Cup committee. Having hosted modeling competitions ourselves we appreciate that the thought and organization that goes into such a task is certainly not trivial.