Computational Intelligence Meets the NetFlix Prize

Ryan J. Meuth, Paul Robinette, Donald C. Wunsch II

***Abstract*— The NetFlix Prize is a research contest that will award $1 Million to the first group to improve NetFlix's movie recommendation system by 10%. Contestants are given a dataset containing the movie rating histories of customers for movies. From this data, a processing scheme must be developed that can predict how a customer will rate a given movie on a scale of 1 to 5. An architecture is presented that utilizes the Fuzzy-Adaptive Resonance Theory clustering method to create an interesting set of data attributes that are input to a neural network for mapping to a classification.**

I. Introduction

In the media industry, the ability to suggest products to customers is critical to remain competitive. The accurate suggestion of products can lead to greatly improved customer satisfaction as well as expanded sales and customer retention. To online video rental, suggestion is crucial to the continued operation of a company, as these suggestions drive the growth of sales, as customers are exposed to new and interesting media that they would have never otherwise selected.

The NetFlix Prize is an open competition awarding a $1 million prize to the first team able to develop a rating prediction system that beats the existing CINEMATCH rating system by 10%. Over 2700 teams have participated in the competition in the first year, with the top team only achieving a 8.5% improvement[1].

Many of the top teams utilize a collective filtering approach, combining the weighted output of several, even hundreds of models, in the case of the top rated team, to produce their predictions[2].

II. Data Analysis

The full Netflix dataset consists of 100 million anonymous ratings of 480 thousand customers over nearly 18 thousand movie titles. The data set consists of customer id, movie id and rating triplets. The ratings are on a scale of 1 to 5, where 1 is extremely poor, and 5 is excellent. Several test sets are provided, as well as a 2.8 million record qualifying test set where the ratings have been removed from the triplets.

Since the dataset is so large, initial development has been performed on a small subset of the data, containing the ratings of 1000 users over the top 100 movies. This dataset is still significant, containing over 28,000 records. This allows rapid development of the data mining scheme while providing a benchmark to the total dataset.

The primary data table contains 28,181 entries, and each entry represents a single rating of one movie by one customer. Each entry includes the attributes movie_id, customer_id, rank, and rank_date. movie_id uniquely identfies one of 100 movies covered by this data. customer_id uniquely identifies one of 1,000 customers as the source of the rating. rank is a value in [1,5], with 5 being the most positive rating, and rank_date gives the time the rating was submitted.

The second table has 100 entries over three attributes, movie_id, title, and release_date, one entry for each unique movie. The title is a free text attribute.
The data has the following properties:

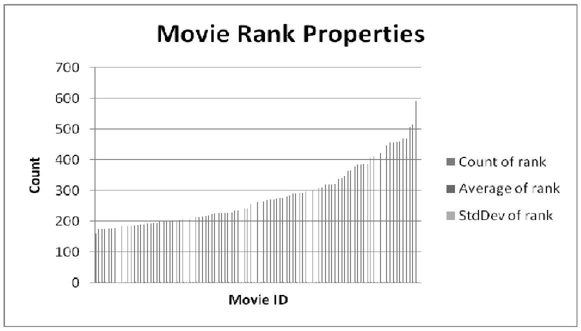

Figure 1. Distribution of the number of rankings for each movie. This shows there is a wide distribution about the mean of 281 rankings per movie.

Manuscript received December, 2007.

Ryan Meuth, Paul Robinette and Donald C. Wunsch II are with the Applied Computational Intelligence Laboratory, Missouri University of Science and Technology, Rolla, MO, 65401 USA (e-mail: rmeuth@mst.edu, pmrmq3@mst.edu and dwunsch@mst.edu, respectively).

978-1-4244-1821-3/08/$25.00©2008 IEEE

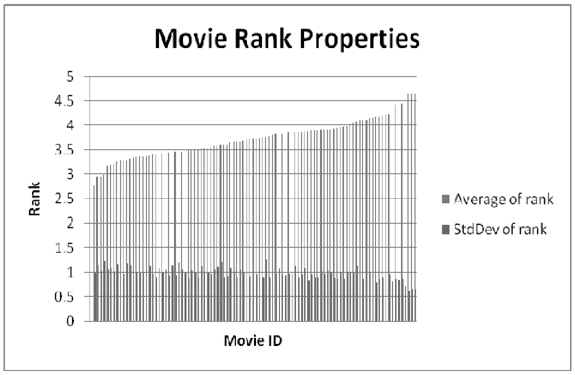

Figure 2. Distribution of Rank Average by Movie. Note that most ranks are very close to the average of 3.7.

There are 28,181 total rankings across all movies and customers, with an average of 281 ranks per movie, at a standard deviation of 96 ranks. The average movie rank value is 3.7, with a standard deviation of 1.

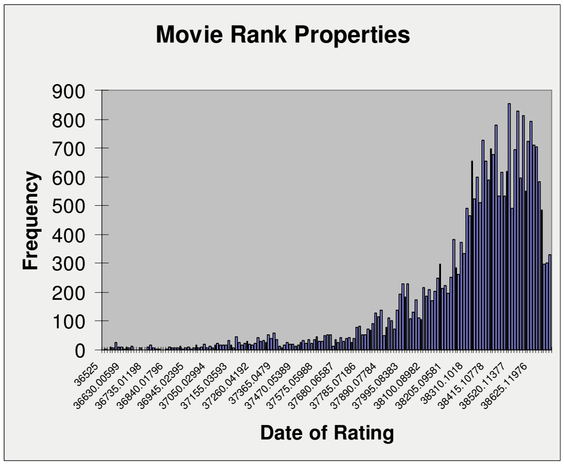

Figure 3. Distribution of Rankings by Date.

The clear majority of rankings were submitted in the latest two years of the time span covered by the data.

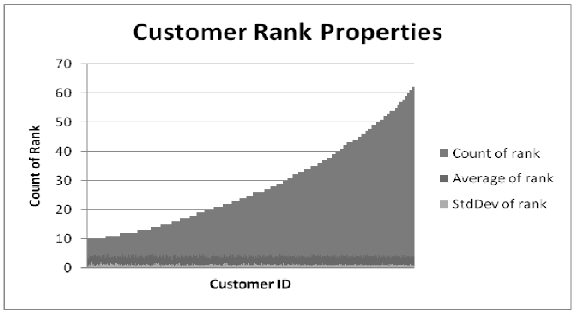

Figure 4. Distribution of Customer Ranks. Customers ranked 28 movies on average, with a standard deviation of 14.

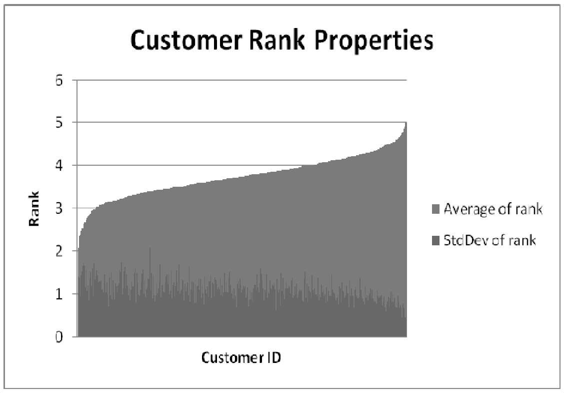

Figure 5. Distribution of Rank Average by Customer. Again, note that most customers rank movies very close to the average of 3.7.

III. Additional Data Sources

Additional data has been collected from the Internet Movie Database utilizing a web-crawler, for each of the 100 movies.

This additional data contains detailed information on each movie of interest. This data includes, for each film, MPAA rating, directors, actors, genres, and box office gross. The goal is to focus on individual Netflix user behavior, and it is possible that several Netflix users give a particular movie a variety of ratings.

IV. Data Transformation

For the MPAA ratings, a scale was assigned to map from a rating to a number. The scale is as follows: 1=G, 2=PG, 3=PG-13, 4=R. This allows the analysis method to directly handle the MPAA ratings.

The data was formatted so that string-based data was re-cast as numerical data. This aids in the ability for our analysis methods to process the data. For example, instead of a list of actors for each movie, the top 20 most popular actors are selected, and 1 attribute was added for each movie that correspond to whether or not an actor in the top 20 starred in a given movie. For genre, 12 attributes were added to each movie, indicating whether or not a movie belonged in that genre. Attributes on average and standard deviation of ratings for each movie were also added. These attributes were also collected for how each user rated movies.

V. Modeling Architecture

The modeling technique that has been implemented is a combination of the Fuzzy ART Clustering Method, parameter optimization, and Back-propagation neural networks.

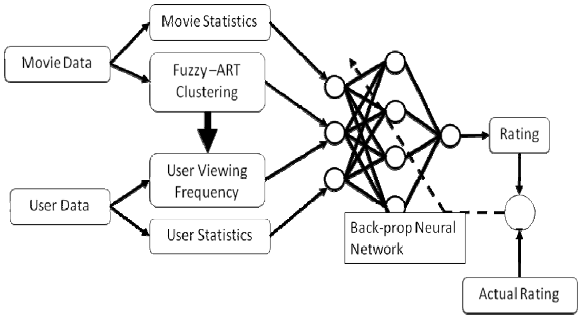

Figure 6 – Overview of Modeling Architecture

The modeling architecture consists of a Fuzzy-ART unit providing input to a back-propagation trained neural network. The input data set is divided into two groups – user data and movie data. The movie data is clustered utilizing the Fuzzy-ART unit into categories based on the movie's genre, MPAA rating, Box office grosses and other parameters. For each movie in the database, rather than using only the category that the movie best matches, a fuzzy category membership vector is produced. For example, the movie "The Terminator" may be most strongly associated with other action movies, but it will hold some similarity with science fiction movies. This additional information is useful, and can be used to paint a more detailed picture of the customer preferences

.

For each customer, a movie category frequency is calculated based on the customer's viewing history. This frequency is calculated by accumulating the fuzzy membership vectors of the movies in a viewer's rating history, weighted by the user's rating of that movie. Then the accumulated history is normalized to the 1-0 domain using min-max scaling. The weighting of the membership vectors is intended to model the customer's selection criteria, so that characteristics of movies that have historically appealed to the customer are emphasized in the frequency, and vice versa for movies that the customer did not like.

The modeling method assumes that both a large and diverse body of movies, customers, and customer histories exist. Though this is the case for both the number of customers and the number of movies, the customer rating histories are not always extensive. New customers with small viewing histories may be misclassified by the system, so a separate method may be needed to handle these cases. Fortunately, all customers in the training set have a viewing history of 10 movies or greater, with an average of 25 movies.

VI. Modeling Architecture Pseudo-Code

Given C, the set of all Customers and c is an element of C.

Given $H(c)$, the Set of Movie-Rating Pairs for customer c, where $h(c)$ is an element of $H(c)$. Each $h(c)$ contains two elements, a movie ID j, and an integer rating r, valued between 1 and 5.

Given P, the set of all movie properties, where p is an element of P.

Given function $m = FuzzART(p(j))$ which returns the fuzzy membership vector of the movie with ID j.

Let M be the set of fuzzy membership vectors with m an element of M.

Let F be the set of viewing frequencies to be calculated, with size $|C|$ and let f be an element of F.

Let V be the set of training vectors, with T target values. Let v be an element in V, and t an element in T.

For each c in C, find Average and Standard deviation of Ratings, S_c.

For each p in P, find Average and Standard deviation of Ratings, S_p.

```
For Each p in P at some vigilance λ {
  m(p)=FuzzART(p)
}
```

```
For Each c in C, {
  f(c)=0;  // Intialize Viewing Frequency
  For Each h(c) in H(c) {
    m(p(h(c).j) = FuzzART(p(h(c).j));
    f(c)  = f(c) + h(c).r*m;  // Update Viewing Frequency
  }
  f(c) = ||f(c)||;  // Normalize
}
```

```
For Each c in C {
  For each h(c) in H(c) {
    Construct v(h(c)) by concatenating the following
  elements:
      f(c), m(p(h(c).j)), S_c(c), Sp(h(c).j));
    If Training
      t = h(c).r;
  }
}
```

If training use V and T to train the network, otherwise apply V to the network obtain a set of predictions.

VII. Fuzzy Adaptive Resonance Theory

Adaptive resonance theory (ART) was developed by Carpenter and Grossberg as a solution to the plasticity and stability dilemma, i.e., how adaptable (plastic) should a learning system be so that it does not suffer from catastrophic forgetting of previously-learned rules[3-5]. ART can learn arbitrary input patterns in a stable, fast, and self-organizing way, thus overcoming the effect of learning instability that plagues many other competitive networks. ART is not, as is popularly imagined, a neural network architecture. It is a learning theory hypothesizing that resonance in neural circuits can trigger fast learning.

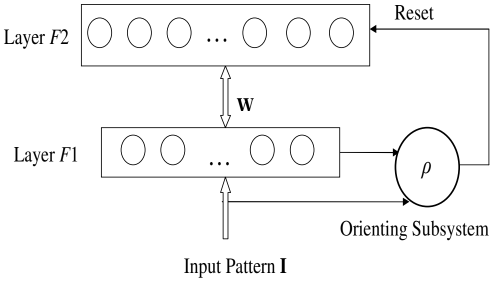

Figure 7. Topological structure of Fuzzy ART. Layers $F1$ and $F2$ are connected via adaptive weights W. The orienting subsystem is controlled by the vigilance parameter ρ.

Fuzzy ART (FA) incorporates fuzzy set theory into ART and extends the ART family by being capable of learning stable recognition clusters in response to both binary and real-valued input patterns with either fast or slow learning. The basic FA architecture consists of two-layer nodes or neurons, the feature representation field F_1, and the category representation field F_2, as shown in Fig. 1. The neurons in layer F_1 are activated by the input pattern, while the prototypes of the formed clusters are stored in layer F_2. The neurons in layer F_2 that are already being used as representations of input patterns are said to be committed. Correspondingly, the uncommitted neuron encodes no input patterns. The two layers are connected via adaptive weights, $\mathbf{W}_j$, emanating from node j in layer F_2. After layer F_2 is activated according to the winner-take-all competition, which occurs between a certain number of committed neurons and one uncommitted neuron, an expectation is reflected in layer F_1 and compared with the input pattern. The orienting subsystem with the pre-specified vigilance parameter ρ ($0\leq\rho\leq1$) determines whether the expectation and the input pattern are closely matched. If the match meets the vigilance criterion, weight adaptation occurs, where learning starts and the weights are updated. This procedure is called resonance, which suggests the name of ART. On the other hand, if the vigilance criterion is not met, a reset signal is sent back to layer F_2 to shut off the current winning neuron, which will remain disabled for the entire duration of the presentation of this input pattern, and a new competition is performed among the remaining neurons. This new expectation is then projected into layer F_1, and this process repeats until the vigilance criterion is met. In the case that an uncommitted neuron is selected for coding, a new uncommitted neuron is created to represent a potential new cluster.

FA exhibits fast, stable, and transparent learning and atypical pattern detection. The Fuzzy-ART method has the benefit of being a highly efficient clustering method, with a linear runtime complexity.

VIII. Artificial Neural Networks

Artificial neural networks (ANN) attempt to capture the adaptability of biological neurons in a mathematical model for information processing. Artificial Neural networks consist of a series of layers of nodes, known as artificial neurons, connected by weights. Each node in a layer is connected to every node in the previous layer by a series of weights. The network operates by applying a vector to the input of the network. At each node in the first layer, the input vector is multiplied by the node's set of weights, and these values are summed together and a transfer function is applied to get an activation level for the node. Typically the transfer function is a logarithmic sigmoid or linear function. This process of accumulating weighted values and computing activations is repeated through the layers of the network until the output layer is reached.

Neural networks are not programmed; rather they are trained using one of several kinds of algorithms. The typical structure of a training algorithm starts at the output of the network, calculating an error between the actual network output and a target output for a given input vector. This error is used to adjust the weights of the network based on the amount of influence that a given weight had on the output. This process repeats from the output side to the input side, and is thus known as error back-propagation. There are many methods for how to make these weight adjustments.

IX. Parameter Selection

The parameters of the Fuzzy-ART unit and the neural network are found empirically. For the Fuzzy-ART unit, this is a simple procedure, where a range of vigilances are applied, and the resulting number of categories is plotted. Ranges of vigilance values where the number of categories remains constant indicate a natural divide in the data at that sensitivity level.

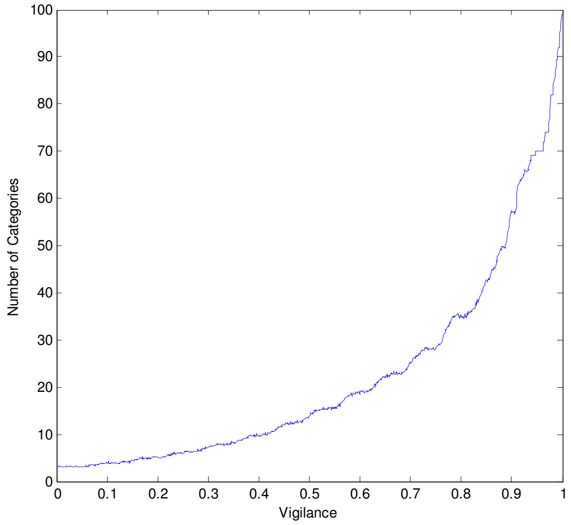

Figure 8. Movie Data Clustering Profile. The largest category plateau falls within the vigilance range of 0.5 to 0.55.

The vigilance of 0.55 is chosen to produce approximately 15 clusters.

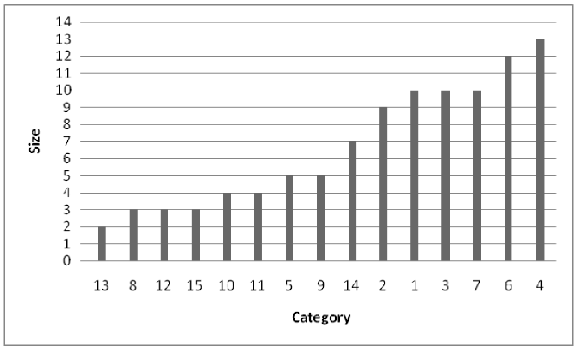

Figure 9. Movie Category Distribution

For the neural network, however, few methods exist for quickly determining optimal parameters, so the architecture and learning parameters are found by trial and error. The architecture was chosen to be a three layer design, with sigmoid activation function for the hidden layer, and a linear output layer. The hidden layer size was chosen to be twice the input layer size, and the output layer is a single node. This is a typical design for function approximation. The default training values of MATLAB neural network toolbox were used. The Resilient Back-propagation training algorithm was used for a balance of speed and accuracy.

The validation data set was used to detect when to stop training. When the mean-squared error of the validation set stays the same or rises over 3 epochs, training is terminated.

X. Results

The modeling architecture is trained using 50% of all customer records, but using all available movie data. 25% of the data is used to determine when to stop training iterations on the neural network. The remainder of the data is used to evaluate the performance of the model.

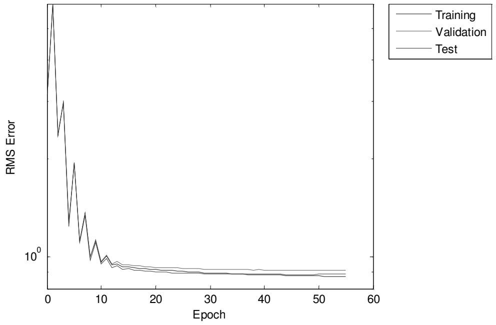

Figure 10. An example training session.

Due to the models non-deterministic properties, the model was tested over 30 runs, using randomly selected sub-sets from the given body of data. Evaluated against the test data-sets, the average RMS Error of the Model is 0.8769, with a standard deviation of 0.005. The minimum RMS Error of the runs was 0.8663. It is expected that the deployed performance of the system would be comparable.

XI. Conclusion

The NetFlix prize is a highly challenging competition, with such a large dataset and highly non-linear relationship between a user's rating history and their future ratings that traditional data analysis methods often fall far short.

An architecture is presented that combines several computational intelligence techniques, as well as novel attribute creation that is able to improve on the accuracy of the existing system with only a linear complexity to the size of the dataset.

With an expected performance of 0.8769 RMS Error, the system only achieves a 7.8% improvement over the Netflix's CINEMATCH system. This does not satisfy the competition objective of 10% improvement, but it is a significant step towards this goal. Placed on the Netflix Prized leader-board, this system would fall within the Top 10.

Future development of the system can include the development of new attributes, particularly related to movie content and plot development. Additional information about customers may be useful, such as the region of residence, i.e.

rural, suburban, urban, etc. Also, marketing information on a movies' target demographic may be helpful, as well as the utilization of more sophisticated modeling techniques such as time-series prediction.

Many other groups have utilized the weighted output of several, sometimes hundreds of models to achieve higher accuracy. Development in this direction may prove useful.

Members of our group will be registering as a development team for the full Netflix Prize challenge.

References

[1] "NetFlix Prize Website." vol. 2007.

[2] R. Bell, Y. Koren, and A. C. Volinsky, "Modeling relationships at multiple scales to improve accuracy of large recommender systems " in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007.

[3] G. A. Carpenter and S. Grossberg, "Fuzzy ART: Fast Stable Learning and Categorization of analog patters by an adaptive resonance system," *Neural Networks,* vol. 4, pp. 759-771, 1991.

[4] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition," *Neural Networks,* pp. 493-504.

[5] S. Grossberg and G. A. Carpenter, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, and Image Processing,* vol. 37, pp. 54-115, 1987.