An Application of Case Based Decision Theory to the Netflix Competition

Michael Naaman

Department of Economics, Rice University

September 2012

Abstract

The Netflix competition started out in 2005 as a grassroots competition to improve on the Cinematch recommendation system by 10% in RMSE with the winner receiving one million dollars in prize money. Nearly three years later, there was no single algorithm that won the day, but this paper presents an alternative algorithm that performed better than the Cinematch algorithm, but fell short of the winning blend of algorithms. We will also show how are approach can be combined with other algorithms to make further improvements in recommendation systems.

Introduction

In classic economic modeling, economists assume that preferences are an inherit trait of the decision maker. We generally assume that a decision maker knows their preferences over all possible goods and we can represent this preference structure with a quasi-concave, differentiable, utility function.

The representation of preferences with utility functions requires a certain degree of hypothetical reasoning on the part of the decision maker. For example, assume that a decision maker must express a preference between 2 billion apples and 1 billion oranges. This is an extreme situation that the decision maker will not encounter, but she might reason that she does prefer 2 apples to 1 orange, so it stands to reason that she may also prefer 2 billion apples to 1 billion oranges. In this context, the two situations are very different in their absolute scale; however, they are very similar in the sense that they both express the same preference rule that apples are preferred to oranges when there are twice as many apples as oranges.

In certain circumstances, it is natural to develop a model of preferences evolving over time based on past experiences. This is referred to as "case based decision theory" (CBDT) and allows one to model preferences based on recollections of trade-offs, decisions, and the outcomes of those decisions that have played out in the past. Formally, if we let *A* be the set of acts that are available to the decision maker from some decision problem *p* and let $\mathrm{c} = (\mathrm{a}, \mathrm{p}, \mathrm{r}) \in \mathrm{M}$ be the triple consisting of the act, *a*, chosen in a decision problem, *p*, and the outcome, *r*, that resulted from the act, then for any given subset of memories, *I*, we can express preferences over acts conditional on those memories, which we denote by $\{\succeq_I\}$. If the preference relation satisfies certain axioms we can express the preference relation over acts with a utility function given by

$$\mathrm{U(a)} = \sum_{\mathrm{c}\in\mathrm{M}} \mathrm{s(p,q)u(r)} \qquad \text{where} \quad \mathrm{c} = (\mathrm{a,q,r})$$

So the term $s(p,q)$ is the similarity over the decision problem given that act, *a,* was chosen. But we may also have a situation where there is similarity between act decision problem pairs. For example, if our set of acts consists of buying or selling a stock and our set of decision problems consists of buying the stock when the price is high or low, then we may have a situation where "buying the stock when the price is low" is more similar to "selling the stock when the price is high" than "selling the stock when the price is low". Gilboa and Schmeidler (1997) provide axioms that allow a generalization that includes similarity over the pair of decision problems and acts by

$$\mathrm{U(a)} = \sum_{\mathrm{c}\in\mathrm{M}} \mathrm{w((p,a),(q,b))u(r)} \qquad \text{where c=(q,b,r)}$$

To see the usefulness of cased based utility models, compare this to the standard approach in a classical partial equilibrium tax model, which will be revisited in chapter 2.

Consider another example that will motivate our approach in chapter 3. Suppose a decision maker, DM_1, who has a complete ordering of preferences for *N* movies. Assume we ask the person to partition the movies into two groups such that all movies from group *G* are weakly preferred to group *B* and we ask them to report a 1 for the movies in group *G*, and a 0 for the bad movies in group *B*. Next we take a random draw, *z,* from the *N* movies and we ask her to predict the quality of the movie and we give incentives so that she will seek to minimize her prediction error. If we make the draw with replacement, reporting the proportion of good movies to bad movies will maximize her utility.

Clearly, DM_1 will try to make high quality predictions and the estimate is just the expected utility.

$$\mathrm{EU_1(z) = pU_1(x) + (1\text{-}p)U_1(y) = p = \frac{|G|}{N}} \quad \text{where } \mathrm{U_1(x) = 1} \text{ and } \mathrm{U_1(y) = 0}$$

In this case, there is no need for the case based utility representation; however, if we make the draws without replacement, then over time DM_1 learns about the distribution of the remaining good movies, so her utility over the precision of the estimates will be maximized at the estimator that minimizes the MSE. At draw *t+1*, she can improve on the naive estimator *p* with

$$EU_1\left(z_1^{t+1}\middle|\Omega_1^t\right) = \frac{|G| - t\bar{u}_1^t}{N-t} = p_1^t$$

where $\bar{u}_1^t$ is the mean utility up to time *t* and $\Omega_1^t = \left\{z_1^i, 1 \le i \le t\right\}$

Since we are trying to predict the good movies, we can set the represent this as a weighted average of past cases by using the number of good movies, $\left|G_1^t\right|$, and bad movies, $\left|B_1^t\right|$, up to time *t*, so that $s(p,q_t) = \frac{|G|}{\left|G_1^t\right|} - 1$ if the movie is good and $s(p,q_t) = \frac{|B|}{\left|B_1^t\right|} - 1$ if the movie is bad. Of course, the sum of $s(p,q_t)$ up to time *t* is *N-t*, we must standardize by the similarity factors, which gives

$$EU_1\left(z_1^{t+1}\middle|\Omega_1^t\right) = \frac{|G| - \left|G_1^t\right|}{N-t} = p_1^t.$$

This has the nice interpretation as being the conditional expected utility given the past. Also notice that for any time *t* realization of a good movie increases the bad movie weight and decreases the good movie weight because there is a relatively higher chance of getting a bad movie in the future.

Assume we add another decision maker, DM_2, with a utility function over movies, U_2. To keeps things simple, assume both parties have access to all of the reported utilities and that both decision makers have the same preference structure. The information set has expanded to include DM_2, so the conditional expectation will be given by have

$$\begin{aligned} EU_1\left(z_1^{t+1}\middle|\Omega_1^t \cup \Omega_2^t\right) &= \frac{|G| - \left|G_1^t\right| - \left|G_2^t\right| + \left|G_1^t \cap G_2^t\right|}{N-t} \\ &= p_1^t + p_2^t - \frac{(|G| - \left|G_1^t \cap G_2^t\right|)}{N-t} = \lambda p_1^t + (1-\lambda)p_2^t \end{aligned}$$

for some $\lambda \in [0,1]$. if we allow the decision makers to have different preferences, then we can take the correlation, ρ_t, between the two decision makers' ratings, which will be an estimate of the common good movies for both of the decision makers up to time *t*. For example, if the two decision makers are perfectly correlated, then the second decision maker faces no uncertainty about the quality of the movie. However, we could have perfect correlation up to time *t* with differing preferences at time *t+1*, so it is not clear how DM$_1$ would use this imperfect information. If DM_1 just takes a weighted average, then

$$E\left[E_t\left[U_1\left(z_1^{t+1}\right) | U_2(z_2^{t+1}) = a\right)\right] = E[\lambda p_1^t + (1-\lambda)a] = \lambda p_1 + (1-\lambda)p_2$$

So our estimate of DM_1 is biased, unless $\lambda = 1$ or $p_1 = p_2$, which violates our assumption. Of course, we could demean the report of DM_2 so

$$E[E_t[U_1\left(z_1^{t+1}\right) | U_2(z_2^{t+1}) = a)]] = E[\lambda p_1^t + (1-\lambda)(a - p_2^t)] = \lambda p_1$$

The estimate is still biased unless $\lambda = 1$ and it is unclear how to find λ. However, if we use the best linear predictor based on the estimates of DM_1 about DM_2, then we will be minimizing the MSE over unbiased estimators, so the utility estimator will be

$$EU_1(z_1^{t+1} | U_2(z_2^{t+1}) = x) = p_1^t + \sigma_1^t \rho_t \frac{(x - p_2^t)}{\sigma_2^t}$$

The second part of this term involves the standardized rating of the second user, the correlation, and the average deviation from the mean of DM_2. By standardizing the report of DM_2, we are measuring how much on average DM_2 deviates from her mean. Then we multiply by the correlation, which is our estimate of similarity, and multiply by standard deviation of DM_1 to see how much DM_1 will deviate from her mean given DM_2.

We can reformulate the problem into the CBDT framework. For any case *c* we have the act, *a*, which is the reported rating of movie *p* and the resulting utility function over acts is given by

$$\mathrm{U(a \mid p)} = \sum_{\mathrm{c \in M}} \mathrm{w(a,b)u(r)} \text{ where c=(q,b,r)}$$

Unfortunately, we don't know the decision maker's utility, but we do have the reported utility, $\tilde{U}(a \mid p)$. Presumably, the decision maker is trying to reveal her preferences, so if the decision maker gets utility $u(r, p)$ from act *a* in decision problem *p*, then we can assume $\mathrm{U(a \mid p) = f(\mid u(r) - \tilde{U}(r) \mid, p)}$ where *f* is some increasing function so that the decision maker maximizes her utility be reporting her utility as truthfully as possible relative to her other reports. If we restrict our attention to cases where the decision maker actually reported a utility, then whenever *a* is chosen, we have

$$U(a \mid p) = \sum_{c \in M} s(p,q)\tilde{U}(c)$$

It can be shown that the solution of this maximization problem is to report $\tilde{U}(p) = E[u(p) \mid M]$. This means our similarity function should be a probability measure, but the weighting matrix is only unique up to a scale factor. This implies we need to estimate a model of the form

$$(I - \lambda W)\tilde{U} = \varepsilon$$

Of course, that assumes we had the weighting matrix in hand, but we can estimate *W* assuming we can adequately represent the memories that are being conditioned upon. In order to do this, we use some distance measure, like correlation, and use this distance measure to estimate *W* with a nonparametric regression. We assume that any distance measure that preserves the similarity ordering will be the same asymptotically, but proving this is beyond the scope of our paper.

Since this person has never faced this decision problem, we would like to gain insight about it. For instance, in our last example, we could set

$$w(a,c) = w(a,(a,t,x)) = \frac{\sigma_1^t \rho_t}{\sigma_2^t}$$
$$I(c) = I(a,t,x) = x - p_2^t$$

Using other peoples reported utility is justified because $I(c)$ represents the information the decision maker has for case *c*, which could be information about some other decision maker that is relevant. Information about similar decision makers is fundamentally different, so we would like to restrict the cases, c_i, to be positive weights. For example, if a good friend loved some movie, we might take that into account in our decision about what rating to assign that movie. Suppose the movie was a musical and the decision maker hates musicals; but since her friend loved the movie, the decision maker might be inclined to give the movie a higher rating because it was a high quality musical.

$$U(a \mid I, p) = \sum_{i=1}^{n} \sum_{c_i \in M} w(a,c)U(b \mid q) + w(a,c_i)I(c_i)$$

If this new specification can still represent the decision makers' preferences, then it must be the case that our estimating matrix is a linear combination of the true weighting matrix, which means we must scale the estimated weighting matrix.

Our approach, for scaling the weighting matrix, is to assume the error is normal and perform maximum likelihood on the model

$$\left(I-\lambda_1(I\otimes W_1)-\lambda_2(W_2\otimes I)\right)U=\varepsilon \text{ where } \varepsilon \sim N(0,I)$$

In order for this model to be identified, *W* must be invertible. To ensure this is to normalize the rows of W_1 and W_2 one and then restrict $\lambda_1+\lambda_2<1$. This allows us to partition the model into the similar cases for the decision maker and similar decision makers for this decision problem. Since our weights are positive and sum to unity, then λ_1 and λ_2 can be interpreted as evidence for the expected utility given similar decision problems. Finally, from a statistical perspective, the model is symmetric in movies and users, so *apriori* there is no reason to expect the estimates to be different, but our model shows such a stark contrasts between the models that it gives evidence for the case based utility over similar chosen decision problems.

The Netflix project was a competition to help solve an information problem. Netflix is a company that rents movies on the Internet. Customers make a list of movies that they are interested in and Netflix mails them those movies as they become available. Then the customer watches the movies and sends it back to Netflix, but Netflix charges only a membership fee without any late fees or rental fees. Thus Netflix can only increase its revenue by getting new members or by getting its old members to upgrade to a more expensive membership. For example, a customer could upgrade from two movies being sent at a time to three movies at a time.

They beauty of being an Internet company is that Netflix has a huge centralized collection of movies that can be distributed cheaply, but that is also the problem: Netflix has so many movies that the customers are flooded with choices. It was easy for customers to search for movie they wanted, but there was no knowledgeable rental store clerk to recommend a good drama like there was in a physical movie rental store.

The solution was to try to make a virtual recommendation system so that people could find movies that were unknown to them or maybe even a forgotten classic. This way Netflix could send the customers recommended movies every time they logged on or added a movie to their rental queue. Hopefully, these recommendations might also be added to the queue and the customer would get more movies they enjoyed. So Netflix allowed its customers to rate any movie in the catalogue, including the movies the customer had rented or browsed. Then the company developed an algorithm called Cinematch that tried to predict which movies the customer might like based on the ratings of other customers. The idea was to find a movie that John Doe might like, but hadn't already rented by figuring out which other customers were very similar to John Doe. Then for any particular movie that John Doe had never rated or rented, the similar customers could be used as a proxy for John Doe's preferences, thereby allowing Netflix to make recommendations based on the similar customers' preferences.

As Netflix saw it, the quality of its recommendation system was what would make them stand apart from future competitors, so they outsourced it to everyone. They developed the Netflix Prize in which anyone who could beat the Cinematch program by 10% in RMSE, root mean square error, would win a million dollars and publish the results.

It turned out to be quite difficult to reach that 10% improvement and took almost three years. There was no single idea that won the day.   The winning team, BellKor, was a blend of algorithms from the most successful teams and there were 107 different estimators used in the winning algorithm. As Abu-Mostafa (2012) points out, even the winning algorithm was only a 10.06% improvement on the CineMatch algorithm. In fact this bound was so tight that the second place team, The Ensemble, submitted a solution that tied the BellKor team, but alas it was submitted 20 minutes too late. While the 10% improvement was chosen rather arbitrarily, it proved to be a monumentally difficult task.

The setup of the contest was to supply the competitors with three things: the quiz set, the probe set, and the training set. The training set contains data on the 480189 customers and 17770 movies. The movie titles are given, but for privacy reasons the customer names are not given. For each customer there is a file that contains all of the ratings that customer has ever made, except for the ones that have been removed for the quiz set, and the date of that rental. The quiz set is a randomly chosen subset of the training set with the ratings removed. The quiz set is where teams make their predictions and send them to Netflix. Netflix computes the RMSE for the quiz set and sends the results back. Finally, the probe set is another subset of the training set, but this time the ratings are not removed. The idea of the probe set is that teams could practice with a similar dataset in order to hone their algorithms.

There were thousands of teams all over the world using all different types of algorithms. To fix ideas, we present some of the algorithms other teams have used.

As a good first step one might consider using SVD decomposition in order to get a dimension reduction in the problem. Suppose we have an *m* by *n* matrix, *M*, which is vary sparse. Then we can use an SVD decomposition to find *U* and *V*, so that $M \approx UV^{'}$. The problem is given by

$$\arg\min_{(U,V)} (M - UV^{'})^{'} A (M - UV')$$

where *A* is a matrix of dummy variables that select only the elements for which we have data (Németh 2007).

This is essentially just a factor model with *L* factors that estimate our missing data. In general it is unclear how *L* is to be chosen; however, Kneip, Sickles and Song (2011) present a model that estimates the number of factors by utilizing cubic splines. If we have some panel data set with an endogenous variable *Y* of size *N* by *T* and some exogenous variable *X* of size *P* by *T*, then they consider the model

$$Y_{it}=\beta_0(t)+\sum_{j=1}^{p}\beta_j X_{itj}+v_i(t)+\varepsilon_{it}$$

where $\beta_0(t)$ is an average time varying effect and $v_i(t)$ is the time varying effect of individual, *i*. In order to insure that the model is identified we must also assume that there exists an L-dimensional subspace containing $v_i(t)$ for all $1\le i\le N$ with $\sum_i v_i(t)=0$. Then they outline a methodology utilizing a spline basis, which is beyond the scope of this paper, to estimate $\beta_1,\ldots\beta_p$ and the time varying effects $v_1(t),\ldots v_n(t)$ by minimizing

$$\frac{1}{T}\sum_{i,t}\left(Y_{it}-\bar{Y}_t-\sum_{j=1}^{p}\beta_j\left(X_{itj}-\bar{X}_{tj}\right)-v_i(t)\right)^2+\frac{\kappa}{T}\sum_i\int_1^T\left(v_i^{(m)}(s)\right)^2 ds$$

over β and all *m*-times continuously differentiable functions $v_1(t),\ldots v_n(t)$ with $t\in[0,T]$. There is also a test that can be performed on the size of *L*. This approach is nice because it allows not only the factors to be estimated, but also the number of factors to be used.

While the previous model examined methods to estimate time varying effect, we might also be interested in the spatial relationship of effects. Blazek and Sickles (2010) investigate the knowledge and spatial spillovers in the efficiency of shipbuilding during World War II.

Suppose we are producing a ship, *q*, through some manufacturing process, which takes *L* units of labor to produce one ship. In many manufacturing settings, there is an element of learning by doing based on experience, so assume that shipyard, *i*, in region, *j*, learns to build ship, *h*, according to the equation

$$L_{hij} = AE_{hij}^{\theta}$$

where *A* is a constant, E_{hij} is the experience of shipyard *i* in region *j* that will be used in the production of ship *h* and $\theta < 0$ represents a parameter ensuring that the number of labor units to produce a single output good is decreasing as experience increases. Unfortunately experience is not a measurable quantity, so an econometrician might use the total amount of output as a proxy for experience so that $E_{hij}^{O} = \sum_{m=1}^{T_h} q_{ijm}$ which is the total cumulative output of shipyard *i* up until the time that the production of ship *h* will begin. However, shipyards within any region are likely to be hiring and firing workers from the same labor pool, which means we should expect experience spillover across shipyards within a region. Of course we can represent this learning spillover within a region as

$$E_{hij}^{W} = \sum_{m=1}^{T_h} \sum_{n=1}^{I_j} q_{njm} - q_{ijm} = \left(\sum_{n=1}^{I_j} E_{hnj}^{O} \right) - E_{hij}^{O}$$

This is just the cumulative experience of the entire region *j* without the experience of shipyard *i* so that we capture the effect of the other shipyards in the region. Finally there could also be learning spillovers across regions which can be represented as

$$E_{hij}^{A} = \left(\sum_{m=1}^{T_h} \sum_{n=1}^{J} \sum_{o=1}^{I_j} q_{nom} \right) - E_{hij}^{W} = \left(\sum_{n=1}^{J} \sum_{0=1}^{I_j} E_{hno}^{O} \right) - E_{hij}^{W}$$

In an estimation context, the problem can be represented as a production frontier problem.

$$\ln L_{hij} = \alpha_i + \theta_O \ln E_{hij}^{O} + \theta_W \ln E_{hij}^{W} + \theta_A \ln E_{hij}^{A} + v_{hij}$$

where α_i is the fixed effect of shipyard *i* and v_{hij} is iid normal with mean zero and variance σ_V^2. However, this model doesn't incorporate the inevitable inefficiencies that occur in any manufacturing process such as new workers or changes in wages. Blazek and Sickles (2010) model this inefficiency with a nonnegative random variable, μ_{hij}, that represents the organizational forgetting that occurred in the shipyard. Their model is given by

$$\ln L_{hij} = \alpha_i + \theta_O \ln E_{hij}^O + \theta_W \ln E_{hij}^W + \theta_A \ln E_{hij}^A + \mu_{hij} + v_{hij}$$

$$\mu_{hij} = \delta_0 + \delta_1 SR_{hij} + \delta_2 wage_{hij} + \delta_3 HR_{hij} + \varepsilon_{hij}$$

where SR_{hij} is the separation rate of employees during the production of ship *h*, $wage_{hij}$ is the average hourly wage rate at shipyard *i*, HR_{hij} is the hiring rate of new workers for ship *h* in shipyard *i* and $\varepsilon_{hij} > 0$ is iid truncated normal with mean zero and variance σ^2 so μ_{hij} will be a nonnegative truncation of the normal distribution with variance σ^2 and mean $\delta_0 + \delta_1 SR_{hij} + \delta_2 wage_{hij} + \delta_3 HR_{hij}$. This model seeks to explain the learning that takes place to build ship by comparing firms that close to each other in terms of physical distance. In our model, we will seek to find a measure of distance between customers and movies, but this measure is not a given parameter like distance. This model gives us yet another approach to modeling the interdependent relationships that occur in real world modeling.

One of the most successful approaches to the problem was the neighborhood-based model, (k-NN). Suppose we are trying to predict the rating of movie *i* by customer *u*, call it r_{ui}. First we would use some metric, like the correlation between movies, to choose a subset of the movies, $N(i;u)$, that customer *u* had already rated that were "close" to the movie in question. For simplicity only the *f* closest neighbors are kept, we would have the prediction rule

$$r_{ui} = \frac{\sum_{j \in N(i;u)} w_{ij} r_{uj}}{\sum_{j \in N(i;u)} w_{ij}}$$

where w_{ij} represents the similarity between the movie *i* and movie *j* and w_i is a vector with *f* elements. For example, it could just be the correlation between movies. If our similarity measure is 1 whenever the movie is a drama and 0 otherwise, then our estimator will simply be the average of all the drama movies that the customer rated. However, the similarity weights could also be estimated in some fashion.

A more advanced approach tries to estimate the similarity coefficients, which was the BellKor team's approach. The first step in their algorithm was to remove all of the global effects by running the regression

$$Y = X\beta + \varepsilon$$

where *Y* is a vector of the ratings by users for different movies and *X* contains global information like movie indicators, time, user indicators, and combinations of the previous. The rest of the analysis will focus on predicting the residual, r_{ui}, from this regression, so our final prediction will be given by

$$prediction_{ui} = \left(X\hat{\beta}\right)_{ui} + \hat{r}_{ui}$$

As a way to improve upon the k-NN models, a least squares approach might be taken to minimize the error in our prediction rule. if $U(i)$ is the set of customers that rated movie *i*, then for each customer $v \in U(i)$ there is a subset $N(i;u,v) \subseteq N(i;u)$ of the movies that customer *v* has rated within the neighborhood of customer *u*. Initially consider the case where all of ratings by person *v* are known, then the least squares problem can be written as.

$$\min_{w} \sum_{v \in U(i)} \left(r_{vi} - \frac{\sum_{j \in N(i;u,v)} w_{ij} r_{vj}}{\sum_{j \in N(i;u,v)} w_{ij}} \right)^2$$

This approach gives equal weight to all customers, but we would like to give more weight to customers that are more influential, so they use a weighting function $c_i = \left(\sum_{j \in N(i;u,v)} w_{ij} \right)^2$ for each user resulting in the following optimization problem.

$$\min_{w} \sum_{v \in U(i)} \frac{c_i}{\sum_{v \in U(i)} c_i} \left(r_{vi} - \frac{\sum_{j \in N(i;u,v)} w_{ij} r_{vj}}{\sum_{j \in N(i;u,v)} w_{ij}} \right)^2$$

Following Bell (2007), we can rewrite this problem as an equivalent GMM problem subject to nonlinear constraints

$$\min_{w,\lambda \geq 0} w^{'} Qw + \lambda \left(1 - \sum w_i\right)^2$$

where $Q_{jk} = \frac{\sum_{v \in U(i)} \delta_{jk} \left(r_{vj} - r_{vi}\right)\left(r_{vk} - r_{vi}\right)}{\sum_{v \in U(i)} \delta_{jk}}$ and $\delta_{jk} = \begin{Bmatrix} 1 & j,k \in N(i;u,v) \\ 0 & otherwise \end{Bmatrix}$. However, this approach ignores some information between customers. If we have some measure, s_{jk}, of the similarity between customers, then we have the simple modification

$$\delta_{jk} = \begin{Bmatrix} s_{jk} & j,k \in N(i;u,v) \\ 0 & otherwise \end{Bmatrix}$$

Previously it had been assumed that the ratings were known, but in reality the number of terms that determine the support for Q_{jk} can vary greatly within the data set, so a shrinkage factor was used

$$\hat{Q}_{jk} = \frac{\sum_{v\in U(i)} \delta_{jk}\left(r_{vj} - r_{vi}\right)\left(r_{vk} - r_{vi}\right) + \frac{\alpha}{f^2}\sum_{jk} Q_{jk}}{\alpha + \sum_{v\in U(i)} \delta_{jk}}$$

where f^2 is the number of elements in Q and α is a shrinkage parameter. Of course we can repeat this process by reversing the roles of customers and movies, but it is less effective, however, the two different results can be combined for further improvements.

Instead of removing the global effects and then estimating the residuals, a refinement can be made that estimates the global effects and the residuals simultaneously. The basic problem is given by

$$\min_{\beta,w,c} \sum_{(v,i)} \left(r_{vi} - \beta_0 - \beta_v - \beta_i - \frac{\sum_{j\in R^k(i;v)} w_{ij}\left(r_{vj} - \beta_v - \beta_j - \beta_0\right)}{\sqrt{\left|R^k(i;v)\right|}} - \frac{\sum_{j\in N^k(i;v)} c_{ij}}{\sqrt{\left|N^k(i;v)\right|}} \right)^2$$

$$\sum_{j=0}^{i+v} \beta_j^2 + \sum_{j\in N^k(i;v)} c_{ij}^2 + \sum_{j\in R^k(i;v)} w_{ij}^2 \geq 0$$

where $R^k(i;v)$ is the set of the *k* most similar movies to movie *v* that have available ratings. This set takes into account the information of the levels of the ratings, but information is also available implicitly because the act of rating a movie says provides information that should be utilized. In order to use this implicit information, $N^k(i;v)$ is the set of *k* most similar movies to movie *v* that are rated by customer *i*, even if the actual rating is unavailable because it is part of the quiz set. Previously the β term was estimated by a fixed effects approach, but it will be driven by the data with this approach.

An alternative to k-NN is a latent factor approach. Paterek, A. (2007) approached the problem by utilizing an augmented SVD factorization model. Under this model, the optimization problem becomes

$$\min_{\beta,p,q}\sum_{(v,i)}\left(r_{vi}-\beta_0-\beta_v-\beta_i-q_i^T p_v\right)^2+\lambda\left(\|p_v\|^2+\|q_i\|^2+\sum_{j=0}^{i+v}\beta_j^2\right)$$

where this sum is taken over all known customer-movie pairs with known ratings. This turned out to be a very effective approach, but a refinement can be made that includes implicit feedback as in the previous model.

$$\min_{\beta,p,q,y}\sum_{(v,i)}\left(r_{vi}-\beta_0-\beta_v-\beta_i-q_i^T m_v\right)^2$$

$$m_v = p_v+\frac{\sum_{j\in N^k(i;u)} y_j}{\sqrt{\left|N^k(i;u)\right|}}$$

$$\sum_{j=0}^{i+v}\beta_j^2 \geq 0$$

$$\|p_v\|^2+\|q_i\|^2+\sum_{j\in N^k(i;u)} y_j^2 \geq 0$$

Here implicit information is being applied to the user portion of the matrix factorization, which improves RMSE. To get an idea on how these two different algorithms are combines. This approach still doesn't include the movie-customer interaction term of the SVD model, so the two approaches can be combined into a single optimization problem given below

$$\min_{\beta,w,c} \sum_{(v,i)} \left(r_{vi} - \beta_0 - \beta_v - \beta_i - q_i^T m_v - \frac{\sum_{j\in R^k(i;v)} w_{ij}\left(r_{vj} - \beta_v - \beta_j - \beta_0\right)}{\sqrt{\left|R^k(i;v)\right|}} - \frac{\sum_{j\in N^k(i;v)} c_{ij}}{\sqrt{\left|N^k(i;v)\right|}} \right)^2$$

$$m_v = p_v + \frac{\sum_{j\in N^k(i;u)} y_j}{\sqrt{\left|N^k(i;u)\right|}}$$

$$\sum_{j=0}^{i+v} \beta_j^2 \geq 0$$

$$\sum_{j\in N^k(i;v)} c_{ij}^2 + \sum_{j\in R^k(i;v)} w_{ij}^2 \geq 0$$

$$\|p_v\|^2 + \|q_i\|^2 + \sum_{j\in N^k(i;u)} y_j^2 \geq 0$$

This is just one example of combining two different algorithms into a single approach. Over the course of the competition, many of the teams collaborated and started to use many different algorithms until the winning algorithm, which was composed of 3 teams and 107 algorithms. One of the most important lessons learned during the competition was the importance of using a diverse set of predictors in order to achieve greater accuracy.

Case based utility is based on the idea that memories of our past decisions and the results of those decisions generate our preferences. That is to say we can represent our preferences as a linear function of our memories. As discussed in chapter 1, Let *A* be the set of acts that are available to the decision maker from some decision problem *p*. Also let $\mathrm{c} = (\mathrm{a},\mathrm{q},\mathrm{r}) \in \mathrm{M}$ be the triple consisting of the act, *a*, chosen in a decision problem, *p*, and the outcome, *r*, that resulted from the act. For any given subset of memories, *I*, preferences can be expressed over acts conditional on those memories, which we denote by $\{\succeq_I\}$. Gilboa and Schmeidler (1995) prove the existence of a utility function given certain regularity conditions which will be represented as

$$\mathrm{U}(\mathrm{a}|\mathrm{p}) = \sum_{(\mathrm{a},\mathrm{q},\mathrm{r})\in \mathrm{M}} \mathrm{s}(\mathrm{p},\mathrm{q})\mathrm{u}(\mathrm{r})$$

So the term $s(p,q)$ is the similarity over the decision problem given that act, *a*, was chosen. The similarity matrix will not be unique in the sense that the preference structure can be generated by some other similarity matrix, $\tilde{s}$, that satisfies

$$\tilde{s} = \alpha s + \mu i'$$

where α is a positive scalar, μ is an arbitrary column vector, and *i* is a column vector of ones. If we only consider similarity matrices that have rows summing to unity, then it can be shown that the possible weighting matrices must have the form

$$\tilde{s} = \alpha s + (1-\alpha) i (i' i) i'$$

which means $0 \le \alpha \le 1$ because all similarity measures must be positive. For our purposes the dimensions of column vectors will be quite large, so all similarity matrices can be approximated by

$$\tilde{s} \approx \alpha s$$

where $0 \le \alpha \le 1$. This fact can be used to search for the most accurate similarity in a certain class of similarity matrices.

In order to see the relationship between CBDT and the Netflix problem, suppose r_{vp} is the rating of movie, *p*, and the rating of this movie is acted out by customer, *v*, so that in our CBDT language

$$\mathrm{U(v|p) = r_{vp} = \sum_{c \in M} s(p,q)u(r)}$$

where $s(p,q)$ represents the similarity between movies *p* and *q*. Naturally the result, *r*, will be the reported rating of movie *q* acted out by customer *v*, which means

$$\mathrm{r_{vp} = \sum_{q} s(p,q) r_{vq}}$$

In practice the similarity function will be unknown, any number of similarity functions can be chosen to represent the preference structure. For example, the k most correlated movies could be used as weights in a k-NN type estimate that would give

$$s(p,q) = \frac{\sum_q w_{pq} H\left(w_{pq} - w_p^{k-1}\right) r_{vq}}{\sum_q w_{pq} H(w_{pq} - w_p^{k-1})}$$

where w_{pq} is the correlation between movies and w_p^{k-1} is the *k-1* largest correlation for movie *p*. This simply means that only the *k* most highly correlated movies are used to predict the rating. Gilboa and Schmeidler (1995) actually point out that the k-NN approach is a violation of the regularity conditions guaranteeing the CBDT representation of utility. They suggest that all observations be used and simply choose small weights for the less similar cases. In fact this is precisely how the Netflix competitors altered the k-NN approach to produce more precise estimates of customer's movie preferences. Recall the early approach taken by the BellKor team to the Netflix problem.

$$\min_w \sum_{v \in U(i)} \left(r_{vi} - \frac{\sum_{j \in N(i;u,v)} w_{ij} r_{vj}}{\sum_{j \in N(i;u,v)} w_{ij}} \right)^2$$

This can be interpreted as a CBDT optimization where the similarity function is learned from the data. The weights are chosen to minimize MSE and there is no limit on the number of nonzero weights, as there is with a standard k-NN approach.

But we may also have a situation where there is similarity between act decision problem pairs. For example, if our set of acts consists of buying or selling a stock and our set of decision problems consists of buying the stock when the price is high or low, then we may have a situation where "buying the stock when the price is low" is more similar to "selling the stock when the price is high" than "selling the stock when the price is low". Gilboa and Schmeidler (1997) provide axioms that allow a generalization that includes similarity over the pair of decision problems and acts by

$$\mathrm{U(a)} = \sum_{(q,b,r)\in M} \mathrm{w((p,a),(q,b))u(r)}$$

This generalization allows for cases and acts to be separated. There are many possibilities, but Gilboa and Schmeidler (1997) provide a multiplicative approach that satisfies the necessary axioms. It was presented as

$$w((p,a),(q,b)) = w_p(p,q)w_a(a,b)$$

Since the weights are positive, the logarithm or both sides can be taken to derive an additively separable similarity function given by

$$w((p,a),(q,b)) = w_p(p,q) + w_a(a,b)$$

This is the similarity function that will be used in our model. As before the utility of any result is simply the reported ratings of a movie, so

$$\mathrm{r_{ap}} = \sum_{(b,q)\in M} \left[\mathrm{w_p(p,q) + w_a(a,b)}\right]\mathrm{r_{bq}}$$

where r_{ap} is the rating provided by customer, *a*, for movie, *p*. Recall that the movie represents the decision problem and the customer represents the act of providing a rating. This weighting function would have been difficult to implement in practice, so a first order approximation was used.

$$\mathrm{r_{ap}} = \sum_{q} \mathrm{w_p(p,q)r_{aq}} + \sum_{b} \mathrm{w_a(a,b)r_{bp}}$$

This weighting function keeps only the most informative movie ratings, which are presumably the ratings made by customer, *a*, for other similar movies. Similarly the ratings of movie, *p*, are weighted by the most similar customers. By assumption $w(x,x)=1$, this fact can used to rewrite the multiplicatively separable weighting function can be written as

$$\mathrm{r}_{\mathrm{ap}} = \sum_{(\mathrm{b},\mathrm{q})\in \mathrm{M}} \mathrm{w}_{\mathrm{p}}(\mathrm{p},\mathrm{q})\mathrm{w}_{\mathrm{a}}(\mathrm{a},\mathrm{b})\mathrm{r}_{\mathrm{bq}} = \sum_{(a,q)\in \mathrm{M}} \mathrm{w}_{\mathrm{p}}(\mathrm{p},\mathrm{q})\mathrm{r}_{\mathrm{aq}} + \sum_{(b,p)\in \mathrm{M}} \mathrm{w}_{\mathrm{a}}(\mathrm{a},\mathrm{b})\mathrm{r}_{\mathrm{bp}} + \sum_{(b,q)\in \mathrm{M}/(\mathrm{a}\cup\mathrm{p})} \mathrm{w}_{\mathrm{p}}(\mathrm{p},\mathrm{q})\mathrm{w}_{\mathrm{a}}(\mathrm{a},\mathrm{b})\mathrm{r}_{\mathrm{bq}}$$

If the third term is drop, the equivalence can be seen, but a generalization of our model that will incorporate this functional form will be given later.

$$\mathrm{r}_{\mathrm{ap}} = \sum_{\mathrm{q}} \lambda_1 \mathrm{w}_{\mathrm{p}}(\mathrm{p},\mathrm{q})\mathrm{r}_{\mathrm{aq}} + \sum_{\mathrm{b}} \lambda_2 \mathrm{w}_{\mathrm{a}}(\mathrm{a},\mathrm{b})\mathrm{r}_{\mathrm{bp}}$$

As discussed previously, we are interested in the class of weighting matrices that have rows summing to unity. As previously demonstrated the weighting matrices will not be unique, but all of the qualifying matrices can be represented with the functional form above as long as the restriction $0 \le \lambda_1 + \lambda_2 \le 1$ is imposed. Finally our CBDT based model will have the functional form

$$\mathrm{r}_{\mathrm{ap}} = \sum_{\mathrm{q}} \lambda \mathrm{w}_{\mathrm{p}}(\mathrm{p},\mathrm{q})\mathrm{r}_{\mathrm{aq}} + \sum_{\mathrm{b}} (1-\lambda) \mathrm{w}_{\mathrm{a}}(\mathrm{a},\mathrm{b})\mathrm{r}_{\mathrm{bp}}$$

with $0 \le \lambda \le 1$. The details of how such a model can be implemented to predict movie ratings in the Netflix competition will be discussed below.

Model

This model began as a class project at Rice University. Naaman, Dingh, and Taylor (2012) used this class project as a springboard to develop a full-scale algorithm for the Netflix competition. Most of the algorithms focused on using only the information between movies because there were so many more customers than users and each movie has much more data than each customer, which is clearly more effective than just using information between customers. We wanted to directly incorporate this symmetry into our algorithm instead of trying to mash two different results together. What set our algorithm apart is that it tries to combine the two sides sort of like digging a tunnel from both sides of the river instead of just one side. However, the trick was to make sure the tunnel met in the middle.

In calculus one can represent a function at any given point by using the first derivative of the function evaluated at some other point suitably close. This concept guided are thinking in that we could take an expansion of the customer's preferences around a particular customer-movie pair by choosing a small neighborhood of customers and movies that were in some sense "close" to that customer-movie pair. This seemed to point us in the direction of spatial regression and case based utility.

Suppose that we have an $N \times 1$ vector of endogenous variables *y* and that there exists some linear expansion of y_i in terms of the other endogenous variables so that we can write

$$y_i = \sum_j w_{ij} y_j + \varepsilon_i \text{ with } E(\varepsilon_i) = 0$$

In the context of spatial regression, we interpret the weights as being a representation of a point on a map using other landmarks on that map. So it seems reasonable to assume that this is a convex representation, $W \geq 0$ with $\sum_j w_{ij} = 1$ for all *i*. These weights might be the result of some other estimation procedure, which is not pursued in this model. Previously examples were given where some sort of weighted least squares subject to constraints estimated the weights. Switching to matrix notation, we can write

$$(I - W)Y = \varepsilon \text{ with } E(\varepsilon) = 0$$

However, this problem is not identified because $I - W$ is not invertible due to the fact that 1 is an eigenvalue of W. But if we assume that there also exists a scaling factor $0 \leq \lambda < 1$, then we are assured that $(I - \lambda W)^{-1}$ exists and we will have the expansion

$$Y = (I - \lambda W)^{-1}\varepsilon = \varepsilon + \lambda W\varepsilon \_ + \lambda^2 W^2\varepsilon + \ldots$$

Under these conditions, we are also assured that $(I - \lambda W)^{-1}$ will have positive entries and the covariance will be given by

$$E(YY') = \sigma^2\left[(I - \lambda W)'(I - \lambda W)\right]^{-1} \text{ where } E(\varepsilon\varepsilon') = \sigma^2 I$$

This model can easily be extended to the case of an $N \times k$ matrix X of exogenous variables

$$(I - \lambda W)Y = (I - \lambda W)X\beta + \varepsilon$$

The reasoning behind this extension is simply that the exogenous variables should contain the same spatial structure as the endogenous variables in order for the model to make sense. If we assume normality, then $Y - X\beta$ will be multivariate normal with zero mean and covariance given by $\sigma^2(I - \lambda W)'(I - \lambda W)$. This allows us to write down the log-likelihood as

$$\ln L = -\frac{N}{2}\ln\left(2\pi\sigma^2\right) + \ln\left|I - \lambda W\right| - \frac{1}{2}\sigma^2\left(Y - X\beta\right)'\left(I - \lambda W\right)'\left(I - \lambda W\right)\left(Y - X\beta\right)$$

Of course, we can solve this minimization problem with standard MLE techniques. However, the $\ln\left|I - \lambda W\right|$ term must be computed at each iteration of the nonlinear optimization problem, which can prove to be numerically expensive. Ord (1975) showed that the Jacobian determinant term of the likelihood can be written as

$$\ln\left|I - \lambda W\right| = \sum_{1=1}^{n}\ln\left(1 - \lambda\rho_i\right)$$

where $\{\rho_i, 1 \le i \le n\}$is the set of eigenvalues of the spatial weighting matrix. If we appeal to the Schur decomposition of a matrix, in general W will not be symmetric, which allows us to write $W = QSQ^T$ where S is an upper triangular matrix with the eigenvalues of W on the diagonal of S and Q is an orthogonal matrix.

$$\ln|I-\lambda W| = \ln\left|Q(I-\lambda S)Q^T\right| = \ln|Q||I-\lambda S|\left|Q^T\right| = \ln|I-\lambda S|$$

The result follows when we realize that $I-\lambda S$ is an upper triangular matrix with a diagonal entry $S_{ii} = 1-\lambda\rho_i$, so the result follows and we have much faster computation. However, Anselin and Hudak (1992) do find evidence for numerical instability for eigenvalues in matrices with more than 1000 entries, but for our purposes all of our weighting matrices will not be so large. This basic model can be extended in many different directions, which are beyond the scope of this paper, but details can be found in Anselin (1988a).

This approach made sense to us because we felt that correlations could be used as a measure of distance between customers and movies. If we had some meaningful weighting matrices that represented the "distance" between movies and customers, then we could apply some spatial regression techniques to reduce RMSE. However, we couldn't find anything in the literature that matched up with our needs, which meant our approach was ad hoc and the result of trial and error.

The first step was to take correlations between movies over the customers that had rated both movies and take correlations between customers over the movies they had in common. However, this does not give a good measure of similarity because two customers may only have one movie in common which tells us little about how similar their preferences are, so the correlations are weighted based on the number of matches two customers or movies had. So if customer *u* has rated *p* number of movies and customer *v* has rated *q* of the movies that customer *u* has rated, then the weight for the correlation would simply be $\frac{q}{q+s}$ where *s* is a scaling factor that could be *p* or just a parameter. However if the scaling parameter depends on *p*, then the weighted correlation matrix will be asymmetric because the weight for the correlation between customer u and customer v would still be given by $\frac{q}{q+s}$, but the scaling parameter would depend on how many movies customer v has rated. We decided to leave s as a global scaling parameter so that when z is large the weight will be close to unity and when z is small the weight will scale the correlation down to zero. It made sense that if two customers had rated a large number of movies, but they only had a couple movies in common, then they were probably not very similar and the high correlations were due to the small sample size.

The next global step of our model was the same as most of the other competitors and that was to get rid of the fixed effects. Let r_{ui} be the rating that customer u gave to movie i with the convention that $r_{u\cdot}$ is the average rating of customer u over all the movies that customer u has rated; and $r_{\cdot i}$ is the average rating of movie i over all the customers that have rated movie i; and $r_{\cdot\cdot}$ is the grand mean which is the average rating over all customers and all movies. This leaves us with the residuals given by

$$y_{ui} = r_{ui} - r_{u\cdot} - r_{\cdot i} + r_{\cdot\cdot}$$

We also tried a random effects model, but that did not perform as well as the fixed effects approach in terms of out of sample RMSE. Some other panel techniques were also applied, but they provided no improvement in out of sample RMSE. But the rest of our model will be working with the residuals of the fixed effects model.

If we are trying to predict the rating of customer u for movie i, then the next step is to choose a cluster of similar movies and customers. For customer u we choose the c customers that had the largest positive correlation and had rated movie i, but c was capped at thirty in order to make the problem numerically feasible. Repeat this same process for the movie cluster so that we have weighting matrices given by W_c and W_m, which are square matrices of possibly different sizes. The rows of the matrices are ordered by the level of correlation so that the first row of W_c corresponds to customer *u*; and the second row corresponds to the customer that is most correlated with customer u; and the third row corresponds to the customer that is the second most correlated with customer *u* and so on. The rows of W_m are ordered in a similar fashion. Finally we standardized the rows of W_m and W_c to sum to unity which is the standard approach in a spatial regression.

We have a panel of $c*m$ data of residuals, which can be stacked giving

$$Y = \begin{matrix} y_{11} \\ \vdots \\ y_{1m} \\ \vdots \\ y_{c1} \\ \vdots \\ y_{cm} \end{matrix}$$

If we let $W = \lambda W_c \otimes I_m + (1-\lambda) I_c \otimes W_m$, then the model can be written

$$Y = WY + \varepsilon$$

In a spatial regression, the parameter λ is a scaling parameter that is restricted to be between zero and one. In that spirit, we also restricted the parameters to sum to less than one, which allows the above representation. If such a representation exists, then the scaling parameter, λ, can be interpreted as the probability the given customer-movie pair can be represented as a weighted sum of the most similar users and $1-\lambda$ is the probability that the customer-movie pair is represented as a weighted sum of the most similar movies.

Finally we are ready to minimize the log likelihood,

$$L = \ln\left|I_{cm} - W\right| - \frac{1}{2} Y' \left(I_{cm} - W\right)' \left(I_{cm} - W\right) Y$$

where $W = \lambda W_c \otimes I_m + (1-\lambda) I_c \otimes W_m$. In order to improve the speed of this nonlinear optimization problem, we can again appeal to the Schur decomposition of a matrix. Let $W_c = PT_cP'$ and $W_m = QT_mQ'$ where P and Q are orthogonal matrices with T_c and T_m being upper triangular matrices with the eigenvalues of W_c and W_m on the diagonals, respectively. This allows the $\ln|I_{cm} - W|$ term on the optimization problem to be rewritten as

$$\begin{aligned}\ln|I_{cm} - W| &= \ln\left|(P \otimes Q)(I_c \otimes I_m - \lambda T_c \otimes I_m - I_c \otimes (1-\lambda)T_m)(P' \otimes Q')\right| \\ &= \ln\left|(I_c \otimes I_m - \lambda T_c \otimes I_m - I_c \otimes (1-\lambda)T_m)\right| = \sum_{i=1}^{c}\sum_{j=1}^{m} \ln\left(1 - \lambda\alpha_i - (1-\lambda)\beta_j\right)\end{aligned}$$

where $\{\alpha_i, 1 \le i \le c\}$ is the set of eigenvalues for W_c and $\{\beta_j, 1 \le j \le m\}$ is the set of eigenvalues for W_m and the last equality follows from the properties of upper triangular matrices. This allows for a much quicker numerical computation of the optimization problem.

Once we have our estimator, $\hat{\lambda}$ in hand, then our prediction for the rating that customer *u* will give to movie *i* is given by $r_{u\cdot} + r_{\cdot i} - r_{\cdot\cdot} + (\hat{W}Y)_1$ and we repeat this process for the next customer-movie pair. We can extend this model by using the rest of the $c * m - 1$ predictions to impute the missing data. When we move on to the next customer-movie pair, there might be some overlap in the predicted values of missing data, but we can simply average the different predictions of the same missing data point. Then we can repeat the whole process using our averaged predictions for the missing data.

In order to get an idea about the relationship between the movie and user effects, we iteratively estimated the RMSE of the different fixed effects. First we calculated the RMSE just using the overall mean as our predictor; then we added the movie effect to our estimator; and finally, we estimated the RMSE over the full fixed effects model consisting of customer and movie effects.

Predictor	RMSE	Improvement
Overall mean	1.130	NA
Movie mean and overall mean	1.053	0.077
Customer mean, movie mean, and overall mean	0.984	0.069

The main thing to notice about the table above is that adding the movie effect produces a larger improvement than adding the customer effect to our predictor. Since there are many more customers than movies, it stands to reason that the movie mean is a more robust predictor than the user mean because on average there are many more customer ratings for any given movie than movie ratings for any given user due to the discrepancy between the number of movies and customers.

Results

We wanted to know the effectiveness of the different parts of our model running a linear regression over our actual RMSE that came from running our model over the entire probe.

$$RMSE_j = \sum_{i=`1}^{9} \beta_i z_{ij} + \varepsilon_j$$

Independent variable	Estimate
Grand mean	$-.00010$ (0.00100)
Movie mean	$-.11608$ (0.00102)
User mean	-0.04138 (0.00107)
Lambda	-0.02369 (0.00113)
Number of ratings in the $c*m$ data set	-0.08695 (0.00118)
Average of the first row of W_m	-0.13912 (0.00194)
Average of the first row of W_c	-0.03404 (0.00154)
Average of all of W_m except for the first row	0.10803 (0.00193)
Average of all of W_c except for the first row	0.03275 (0.00147)

The user mean was not nearly as effective as the movie mean in explaining a small decrease in RMSE of the model and the grand mean was less effective than both the movie mean and user mean. This falls in line with the experiences of the other competitors. In fact, most of the competitors used a movie centric approach which exploited the robustness of the between movie effects. The next thing to notice is that the spatial weight, lambda, is negatively correlated with RMSE, so there is strong evidence supporting a spatial approach. As expected the number of ratings in the data set is also negatively correlated with RMSE, which simply indicates that the model improves as our data set fills up with actual ratings.

In our model, the single most effective factor was the average of the first row of W_m which is the average weight given to the most similar movies that have been rated by the customer in question. As we have more highly correlated movies, the average of the first row of W_m increases which leads to a decrease in RMSE in terms of conditional expectation. As we saw with the customer mean and movie mean, the movie effect is more relevant than the customer effect, but the model is improved by combing both effects.

It seems counterintuitive that the average entry of W_c and W_m, excluding the first row of both matrices, is positively correlated with RMSE. However, this is really a measure of the similarity between the movies and customers that we are using as a basis to make out estimate for any given customer-movie pair. In essence, there is some overlap in our neighborhood of customers and movies. Ideally we would have a set of movies or customers that is very similar to the rating we are trying to predict, but the set of movies or customers themselves are not very similar to each other.

The final RMSE for our model over the probe set is given in the table below with some naïve models as well.

Predictor	RMSE	Improvement on Cinematch
Overall mean	1.130	-17%
Movie mean and overall mean	1.053	-15%
Customer mean, movie	0.984	-2%

mean, and overall mean		
Cinematch	0.965	0%
Spatial Model	0.88	8.8%

This is very close to the 10% improvement that we would need to win the Netflix competition; however, this result is for the probe set which was for practice. When the algorithm was applied to the Quiz set, the RMSE was 0.92 which was a 4.7% improvement on the Cinematch algorithm which was well short of the 10% improvement needed to win. Despite our best efforts, we were never able to explain the discrepancy in accuracy between the two data sets.

Conclusions

In the Netflix competition, we found significant improvements in RMSE by using a spatial model approach that incorporates the interrelationship between movies and customers. One drawback of this approach is its computationally difficulty, but the computational burden can still be handled by most data sets that are encountered in the field and by a judicious choice of the maximum size of the neighborhood used to make predictions.

There are also future extensions to this model by using more advanced weighting matrices. One possible future improvement of this model would to use a more complicated weighting matrix. Recall that the multiplicatively separable model can be written in SBDT form as

$$\mathrm{r}_{\mathrm{ap}} = \sum_{(a,q)\in \mathrm{M}} \mathrm{w}_{\mathrm{p}}(\mathrm{p},\mathrm{q})\mathrm{r}_{\mathrm{aq}} + \sum_{(b,p)\in \mathrm{M}} \mathrm{w}_{\mathrm{a}}(\mathrm{a},\mathrm{b})\mathrm{r}_{\mathrm{bp}} + \sum_{(b,q)\in \mathrm{M}/(\mathrm{a}\cup\mathrm{p})} \mathrm{w}_{\mathrm{p}}(\mathrm{p},\mathrm{q})\mathrm{w}_{\mathrm{a}}(\mathrm{a},\mathrm{b})\mathrm{r}_{\mathrm{bq}}$$

In this paper, the final term was dropped resulting in an additively separable model; however, the model can be generalized to a multiplicatively separable model in the following way.

$$W = \lambda W_c \otimes I_m + (1-\lambda) I_c \otimes W_m - \lambda(1-\lambda) W_c \otimes W_m$$
$$\ln\left|I_{cm} - W\right| = \sum_{i=1}^{c} \sum_{j=1}^{m} \left(1 - \lambda\alpha_i - (1-\lambda)\beta_j + \lambda(1-\lambda)\alpha_i\beta_j\right)$$

where $\{\alpha_i, 1 \le i \le c\}$ is the set of eigenvalues for W_c and $\{\beta_j, 1 \le j \le m\}$ is the set of eigenvalues for W_m. This approach resulted in a smaller RMSE on a subset of the probe, but it was not pursued any further and may be revisited in future work.

Further improvements seem quite likely if the weighting matrices are estimated via the more complicated methods presented earlier. One could also use a kernel approach to estimate the similarity matrices, but the asymptotic theory of such an approach are beyond the scope of this paper. This approach can be applied to other panel data sets as another approach to take into account the relationship between effects and aid in making predictions. While this approach may not have been as successful as some of the other Netflix algorithm, we believe that it can provide a refinement to existing recommendation system algorithms. Spatial regression approaches have typically focused on a single dimension of spatial correlation. Our approach has demonstrated a novel approach to estimating models with more than one type of spatial correlation. Finally one of the greatest lessons of the Netflix competition is the emphasis on multiple approaches being combined to provide more accurate predictions of customer's preferences.

REFERENCES

1. Abu-Mostafa, Yaser S. (2012) Machines that Think for Themselves. *Scientific American 307*, pp 78-81.

2. Anselin, L. (1988a). *Spatial Econometrics: Methods and Models*. Kluwer Academic, Dordrecht.

3. Anselin, L. (1988b). Lagrange multiplier test diagnostics for spatial dependence and spatial heterogeneity. *Geographical Analysis* 20, 1–17.

4. Anselin, L. (1992). Space and applied econometrics. Special Issue, *Regional Science and Urban Economics* 22.

5. Anselin, L. and A. Bera (1998). Spatial dependence in linear regression models with an introduction to spatial econometrics. In: A. Ullah and D. E. A. Giles, Eds., *Handbook of Applied Economic Statistics*, pp. 237–289. New York: Marcel Dekker.

6. Anselin, L. and S. Hudak (1992). Spatial econometrics in practice, a review of software options *Regional Science and Urban Economics* 22, 509–536.

7. Aten, B. (1996). Evidence of spatial autocorrelation in international prices. *Review of Income and Wealth* 42, 149–63.

8. Baltagi, B. and Dong Li (1999). Prediction in the panel data model with spatial correlation. In L. Anselin and R. Florax (Eds.), *Advances in Spatial Econometrics*. Heidelberg: Springer-Verlag.

9. Bell, R., and Y. Koren (2007). Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights. *IEEE International Conference on Data Mining (ICDM'07)*, IEEE.

10. Bell, R., and Y. Koren (2007). Improved Neighborhood-based Collaborative filtering. *KDD-Cup and Workshop*, ACM press.

11. Bell, R., and Y. Koren, and C. Volinsky (2007). The BellKor solution to the Netflix Prize. http://www.netflixprize.com/assets/ProgressPrize2007_KorBell.pdf.

12. Gilboa I. and D. Schmeidler (1995). Case-based decision theory, *Quart. J. Econ.* **110**, 605–639.

13. Gilboa, I. and D. Schmeidler (1997). Act-similarity in case-based decision theory, *Econ. Theory* **9,** 47–61.

14. Gilboa, I. and D. Schmeidler (2001). *A Theory of Case-Based Decision*, Cambridge Univ. Press, Cambridge, UK.

15. Kneip, A., R.C. Sickles, and W. Song (2011). A New Panel Data Treatment for Heterogeneity in Time Trends. *Econometric Theory*.

16. Konstan, J., G. Karypis, J. Riedl, and B. Sarwar (2001). Item-based Collaborative Filtering Recommendation Algorithms. *Proc. 10th International Conference on the World Wide Web*, pp.285-295.

17. Konstan, J., G. Karypis, J. Riedl, and B. Sarwar (2000). Application of Dimensionality Reduction in Recommender System – A Case Study. *WEBKDD'2000*.

18. Koren, Y. (2008). Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model. *Proc. 14th ACM Int. Conference on Knowledge Discovery and Data Mining (KDD'08)*, ACM press.

19. Koren, Y. Factor in the Neighbors: Scalable and Accurate Collaborative Filtering. http://public.research.att.com/~volinsky/netflix/factorizedNeighborhood.pdf, submitted.

20. Linden, G., B. Smith and J. York (2003). Amazon.com Recommendations: Item-to-item Collaborative Filtering, *IEEE Internet Computing* **7**, 76–80.

21. Leroux, J., J.A. Rizzo, and R.C. Sickles (2010). The Role of Self-Reporting Bias in Health, Mental Health and Labor Force Participation: a Descriptive Analysis. *Journal of Empirical Economics*.

22. MaCurdy, Thomas (1983). A Simple Scheme for Estimating an Intertemporal Model of Labor Supply and Consumption in the Presence of Taxes and Uncertainty. *International Economic Review*, 24:2, 265-289.

23. Naaman, M., Trang Dinh, and Jonathon Taylor (2008). Netflix Group Project. *manuscript, Rice University*.

24. Németh, B., Gábor Takács, István Pilászy, and Domonkos Tikk (2007). Major components of the gravity recommendation system. *ACM SIGKDD Explorations Newsletter* **9**: 80.

25. Németh, B., Gábor Takács, István Pilászy, and Domonkos Tikk (2007). "On the Gravity Recommendation System" (PDF), *Proc. KDD Cup Workshop at SIGKDD*, San Jose, California, pp. 22–30, http://www.cs.uic.edu/~liub/KDD-cup-2007/proceedings/gravity-Tikk.pdf.

26. Ord, J.K. (1975). Estimation methods for models of spatial interaction. *Journal of the American Statistical Association* 70, 120–126.

27. Paterek, A. (2007). Improving Regularized Singular Value Decomposition for Collaborative Filtering. *KDD-Cup and Workshop*, ACM press.

28. Riedl, J. (2006). ClustKNN: a highly scalable hybrid model-& memory-based CF algorithm. In Proc. of WebKDD'06: KDD Workshop on Web Mining and Web Usage

Analysis, at 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Philadelphia, PA, USA.

29. Roweis, S. (1997). EM Algorithms for PCA and SPCA. *Advances in Neural Information Processing Systems 10*, pp. 626–632.
30. Tibshirani, R. (1996) Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society B* **58**.
31. Wang, J., A. P. de Vries and M. J. T. Reinders (2006). Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion. *Proc. 29th ACM SIGIR Conference on Information Retrieval*, pp. 501–508.