The Million Dollar Programming Prize

Netflix's bounty for improving its movie-recommendation software is almost in the bag. Here is one team's account

BY **ROBERT M. BELL, JIM BENNETT, YEHUDA KOREN** & **CHRIS VOLINSKY**

IT'S 7:50 P.M. on 1 October 2007 at AT&T Labs, in Florham Park, N.J., and three of us are frantically hitting the "refresh" buttons on our browsers. We have just submitted our latest entry in the year-old Netflix Prize competition, which offers a grand prize of US $1 million for an algorithm that's 10 percent more accurate than the one Netflix uses to predict customers' movie preferences. Although we have not reached that milestone, we are hoping at least to do better than anyone else has done so far; if we can make it to 8 p.m. with the best score, we'll win a $50 000 Progress Prize. For most of the summer we'd been ahead of our nearest rivals by a comfortable margin, and as recently as 36 hours before this moment, our victory still seemed to be a slam dunk.

The previous day, though, the lead had started to slip away from us. First, the teams then in fifth and sixth places merged, combining their talents to vault into second place, making us nervous enough to submit our best effort, which we had been saving for a rainy day. But before our improved score appeared, we were hit by another combination when our two remaining serious rivals joined forces to tie us. Worse, their entry had come

RANDI SILBERMAN

72 seconds before ours, meaning that in the case of a tie, they'd win.

Seventy-two seconds! Could we lose this thing for being a minute too late? Then we realized that there were still 25 hours left, and we still had one more chance. We had to make it count.

We began to receive offers from other competitors to combine our scores with theirs. We politely declined them and planned strategies for our last submission. Sure enough, these bumped up our score by a few hundredths of a percent, at which point we could only wait to see the final score from our newly allied foes.

Refresh...refresh...refresh...

SINCE 1997, WHEN NETFLIX, of Los Gatos, Calif., came up with the idea of sending movie DVDs through the mail to subscribers, its customer base has grown to 10 million. That success stems, in part, from the company's giving quick and easy access to movies. But just as important is the Netflix recommender system, Cinematch, which helps customers get the most out of their memberships. The better the system predicts people's likes and dislikes, the more they'll enjoy the movies they watch and the longer they'll keep up their subscriptions.

As a new subscriber to Netflix, you have several ways to choose movies on the company's Web site. You can browse by genre or search by keyword for a title, actor, or director. After receiving your selections by mail or over the Internet and viewing them, you return to the site to order more movies. At any time, Cinematch lets you rate any movie you've seen by clicking on one to five stars.

As is the case with other recommender systems, such as those of Amazon, the Internet Movie Database, and Pandora, it's in the customer's interest to vote thoughtfully, because doing so helps Netflix figure out his or her tastes. Yet even if the customer declines to offer this feedback, Netflix still notes which movies the subscriber actually orders. After the customer has rated a handful of movies, the algorithm will start recommending titles based on the rating the algorithm predicts the customer will give.

RECOMMENDING MOVIES that customers will enjoy is a complex business. The easy part is gathering the data, which Netflix now accumulates at the rate of some 2 million ratings a day. Much tougher is to find patterns in the data that tell the company which movie a customer would enjoy, if only the person would watch it.

Netflix developed Cinematch to tackle this job using a well-known prediction technique (described below), which it designed to handle the billion or so ratings it had already logged. However, while incorporating so many ratings added to accuracy, it also took a lot of work just to keep up with the increasing scale, let alone to test alternative prediction schemes.

The Netflix researchers were nevertheless curious about the many other techniques for making similar predictions that had been published in the scholarly literature. The problem was that those studies had relied on public data sets containing on the order of a few million ratings, and it would take the small Netflix team a long time to explore how well these alternative techniques worked at a scale a thousand times as large. That is, if they did all the work themselves.

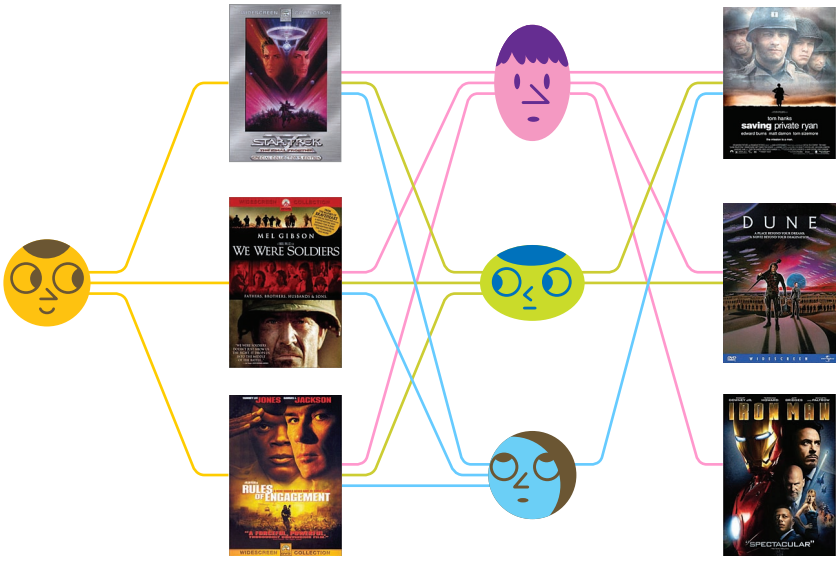
Reed Hastings, the chief executive of Netflix, suggested running a contest. He observed that Netflix had the means, the motive, and the opportunity. And the company had already staged a contest internally between the standard Cinematch system and an alternative algorithm, which did slightly, tantalizingly better.

The Netflix team came up with the basic structure of the contest. They would provide 100 million ratings that 480 000 anonymous customers had given to 17 000 movies. The data set would just fit in the main memory of a typical laptop, allowing almost anyone to compete. Netflix would withhold 3 million of the most recent ratings and ask the contestants to predict them. A

The Neighborhood Model

THE NEAREST-NEIGHBOR METHOD works on the principle that a person tends to give similar ratings to similar movies. Joe likes the three movies on the left, so to make a prediction for him, find users who also liked those movies and see what other movies they liked. Here the three other viewers all liked *Saving Private Ryan,* so that is the top recommendation. Two of them liked *Dune,* so that's ranked second, and so on.

PHOTOS: PARAMOUNT PICTURES



Netflix computer would assess each contestant's 3 million predictions by comparing predictions with actual ratings. The system would use the traditional metric for predictive accuracy, the root-mean squared error (RMSE). The more accurate a set of predictions, the smaller the RMSE will be. The score would then be reported back immediately to the contestant and reflected on an online leaderboard for all to see.

Each such scoring provides valuable information about the hidden ratings—so valuable, in fact, that under certain circumstances it could be used to game the system. The teams were therefore scored once a day, at most. But to help teams estimate how well they might do, Netflix also provided them each with a small representative data set and the score Cinematch had been able to attain for it. Contestants could use that set to test their systems as often as they wanted.

On 2 October 2006, Netflix launched the competition, and within days thousands of teams from hundreds of countries signed up. Within weeks the Netflix Prize Web site was getting hundreds of submissions per day. An online forum was created so that participants could share ideas and techniques, even code. Even more gratifying to Netflix, within months a handful of teams did several percent better than Cinematch. The question then was how much the accuracy would improve in the first year.

★★★★★

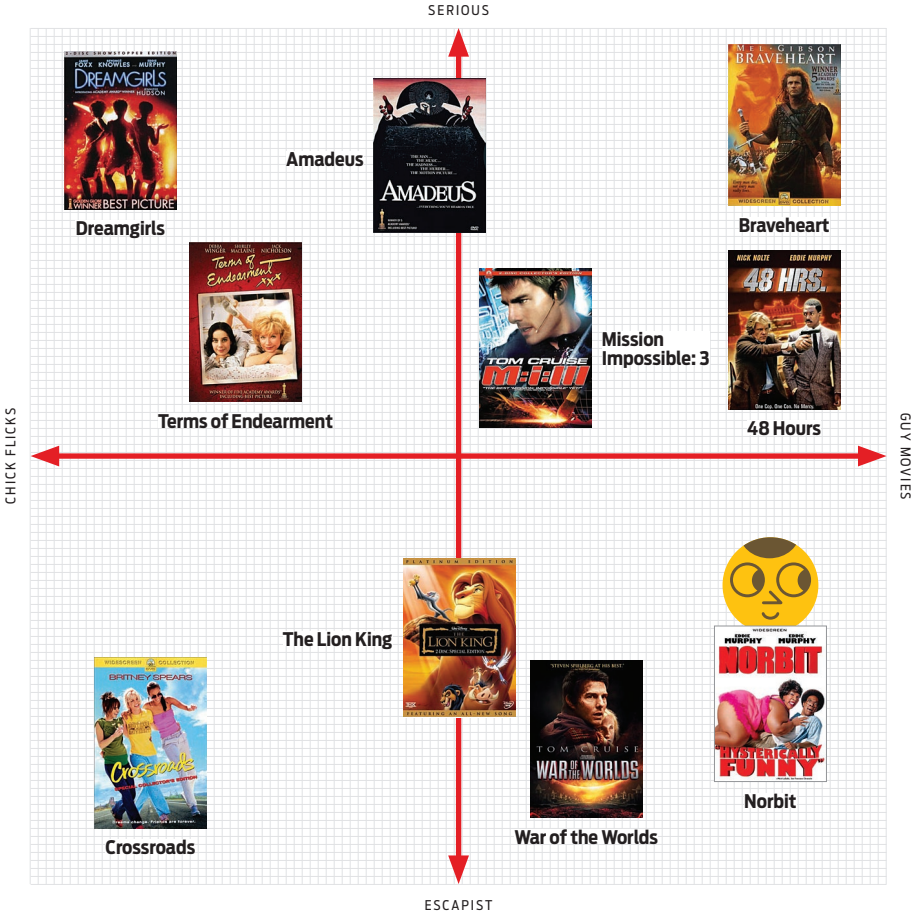
LIKE MOST OF THE other top competitors, the three of us at AT&T Labs consulted the rich body of research on ways of solving problems in this domain, known as collaborative filtering.

One of the main areas of collaborative filtering we exploited is the nearest-neighbor approach. A movie's "neighbors" in this context are other movies that tend to be scored most similarly when rated by the same viewer [see illustration, "The Neighborhood Model"]. For example, consider *Saving Private Ryan* (1998), a war movie directed by Steven Spielberg and starring Tom Hanks. Its neighbors may include other war movies, movies directed by Spielberg, or movies starring Tom Hanks. To predict a particular viewer's rating, we would look for the nearest neighbors to *Saving Private Ryan* that the viewer had already seen and rated. For some viewers, it may be easy to find a full allotment of close neighbors; for many others, we may discover only a handful of neighboring movies. Our version of the nearest-neighbor approach predicted ratings using a weighted average of the viewer's previous ratings on up to 50 neighboring movies. (We have since developed a way to use all past ratings, allowing an unlimited number of neighbors.)

The Latent-Factor Approach

A SECOND, COMPLEMENTARY method scores both a given movie and viewer according to latent factors, themselves inferred from the ratings given to all the movies by all the viewers. The factors define a space that at once measures the characteristics of movies and the viewer's interest in those characteristics. Here we would expect the fellow in the southeast corner of the graph to love *Norbit*, to hate *Dreamgirls*, and, perhaps, to rate *Braveheart* about average.

PHOTOS: AMADEUS, THE SAUL ZAENTZ COMPANY; ALL OTHERS, PARAMOUNT PICTURES



A second area of collaborative-filtering research we pursued involves what are known as latent-factor models. These score both a given movie and a given viewer according to a set of factors, themselves inferred from patterns in the ratings given to all the movies by all the viewers [see illustration, "The Latent-Factor Approach"]. Factors for movies may measure comedy versus drama, action versus romance, and orientation to children versus orientation to adults. Because the factors are determined automatically by algorithms, they may correspond to hard-to-describe concepts such as quirkiness, or they may not be interpretable by humans at all. Factors for viewers measure how much the viewer likes movies

that score highly on the corresponding movie factor. Thus, a movie may be classified a comedy, and a viewer may be classified as a comedy lover.

The model may use 20 to 40 such factors to locate each movie and viewer in a multidimensional space. It then predicts a viewer's rating of a movie according to the movie's score on the dimensions that person cares about most. We can put these judgments in quantitative terms by taking the dot (or scalar) product of the locations of the viewer and the movie.

Both collaborative-filtering techniques work even if you don't know a thing about what's in the movies themselves. All you need to care about is how the viewers rate the movies. However, neither approach is a panacea. We found that most nearest-neighbor techniques work best on 50 or fewer neighbors, which means these methods can't exploit all the information a viewer's ratings may contain. Latent-factor models have the opposite weakness: They are bad at detecting strong associations among a few closely related films, such as *The Lord of the Rings* trilogy (2001–2003).

Because these two methods are complementary, we combined them, using many versions of each in what machine-learning experts call an ensemble approach. This allowed us to build systems that were simple and therefore easy to code and fast to run.

What's more, our ensemble approach was robust enough to protect against some of the problems that arise within the system's individual components. Indeed, the solution we had just submitted on 1 October 2007 was a linear combination of 107 separate sets of predictions, using many variations on the above themes and different tuning parameters. Even so, the biggest improvements in accuracy came from relatively few methods. The lesson here is that having lots of ways to skin this particular cat can be useful for gaining the incremental improvements needed to win competitions, but practically speaking, excellent systems can be built using just a few well-selected strategies.

We don't want to give the impression that heaping together a lot of methods was enough to reach the leaderboard. The Netflix Prize data set poses some huge challenges beyond its immense size. For one, there was enormous variation among viewers and among movies. A rating system must be sensitive enough to tease out subtle patterns associated with those few viewers who rated 1500 movies, without overfitting things—that is, expecting prolific raters' patterns to apply to the many more users who rated 15 or fewer movies. It can indeed be hard to make predictions for a viewer who has provided just a handful of ratings. We improved existing ad hoc methods, designed to address this concern rigorously.

Another critical innovation involved focusing on *which* movies a viewer rated, regardless of the scores. The idea is that someone who has rated a lot of fantasy movies will probably like *Lord of the Rings*, even if that person has rated the other movies in the cate-

Want Advice? Try an Expert

IF YOU want recommendations and you'd rather not rely exclusively on your customers, you can always do the daring thing and consult actual experts. That's the idea behind Pandora, a free Internet radio service that employs musicians to rate songs according to a checklist of criteria, such as pace, rhythm, even the voice of the performer.

The Oakland, Calif.–based company must be doing something right. In the three years since it opened shop, Pandora Media has registered 22 million listeners. So far, all of them are in the United States, although the company is negotiating its way back into Europe, which it left after having problems with music licenses there.

About 2 million people listen to the service on a given day, typically while sitting in front of their computers at work or, increasingly, while clutching their iPhones on the commute home, making for an average session of 6 hours. No wonder Pandora streams more data than any other site except YouTube.

Here's how it works. The listener creates a virtual "channel" by selecting a song, artist, or composer. If a song is chosen, the site compares it to its database of 600 000 songs, each rated by one of its musical experts. The site then selects *another* song it deems to be a close relative and keeps on playing such relatives. (Pandora can't give you your first choice because its licensing contracts ban it from playing songs to order.)

When I selected "A Hard Day's Night," by the Beatles, the first song I heard was "She Loves You," by the same band. I listened for a long time before getting my first choice.

SONG CENTRAL: Tim Westergren and a few CDs waiting to be rated.
PHOTO: RAFAEL FUCHS

You rate a song by clicking on either a thumbs-up or a thumbs-down icon, and the algorithm adjusts its weighting of the musical checklist it uses to select subsequent songs. What's more, a thumbs-down will keep the channel from ever playing the same song again. You have to be careful, because the more thumbs-down you give, the narrower the channel becomes, and in the extreme case you may "thumb yourself into a corner," says Tim Westergren, founder of Pandora.

Even then, he notes, you'd only hobble that one channel. Nothing you do on one channel affects the others, and you may create as many channels as you want.

Westergren says he got the idea for Pandora when he was a young musician working on scores with moviemakers who had very different likes and dislikes. He wanted to find a way to encode those differences in a database he dubbed the Music Genome, paying musicians to do the enormous amount of work.

It may seem strange to use so much manpower as a supplement to computer power, but it makes sense when humans alone can handle the job—a peculiar

gory somewhat low. By replacing numerical scores with a binary who-watched-what score, the data set is transformed from one with mostly missing pieces (the cases in which users don't rate movies) to one that is completely full (using the value 1 when there are ratings and a 0 when there aren't). This approach nicely complemented our other methods.

FINALLY, AT 7:58 on that fateful October evening, all of the scores for the top teams were posted for the last time on the leaderboard, and ours came out highest, with an 8.43 percent improvement on Netflix's algorithm. Our nearest rival scored an 8.38 percent improvement. We didn't do well enough to land a million dollars, but still, we won.

While we've since come very close to the goal of 10 percent, experience has shown that each step forward is harder than the one that came before it, presumably because we've already exploited most of the clues in the data. Nonetheless, we continue to made progress. During 2008 we mined the data for information on how users' behavior changed over time. Later, we joined forces with another team to win the 2008 Progress Prize. Currently we stand at 9.63 percent improvement and are still working hard on the problem.

Now that the confetti has settled, we have a chance to look back on our work and to ask what this experience tells us. First, Netflix has incorporated our discoveries into an improved version of its algorithm, which is now being tested. Second, researchers are benefiting from the data set that the competition made available, and not just because it is orders of magnitude larger than previous data sets. It is also qualitatively better than other data sets, because Netflix gathered the information from paying customers, in a realistic setting. Third, the competition itself recruited many smart people in this line of research.

In any case, the new blood promises to quickly improve the state of the art. Such competitions have invigorated other fields. The various X Prizes that have been offered for advances in genomics, automotive design, and alternative energy have shown an excellent return: By some accounts the recent $10 million Ansari X prize, awarded for suborbital spaceflight, generated $100 million of private investment in space travel.

The competition also validates the concept of collective intelligence popularized in James Surowiecki's 2005 book *The Wisdom of Crowds* (Anchor Books). He argues that the sum of many independent votes is often superior to any one vote, even if made by the greatest expert. For Netflix, the vast number of independent ratings allows for surprisingly good predictions. The power of this collective intelligence is also being harnessed in, for instance, Amazon.com's product recommendations and the collaborative editing of the online encyclopedia, Wikipedia. With the rise of social networks on the Web, we can expect to see and contribute to even more powerful examples in the coming years. ❑

field sometimes called artificial artificial intelligence. One example of AAI is setting puzzles, or "captchas," for visitors to a Web site to solve, both to prove that they're human beings and not bots and to perform some useful chore, such as deciphering the blurred letters from a scan of an old book. Other AAI programs lure people to do such work by providing entertainment or, as Amazon's Mechanical Turk does, money.

Westergren got seed money for the Music Genome in 2000, at the very end of the dot-com bubble. When the bubble burst, he and his colleagues labored almost without income for five years before another injection of capital came through. Even now, Westergren says, Pandora is focused solely on growth and so does not turn a profit. It gets most of its revenue from the banner advertisements its site displays to listeners every time they click on something in the site, something they must do from time to time to prevent Pandora from going silent. It also gets a small royalty whenever a listener buys a song by clicking through to a vendor, such as iTunes or Amazon.

One advantage of using experts is that they can categorize songs that are new, by bands that are unknown. They can also provide a way to get at music that fell out of fashion before Internet rating became possible. Such too-new and too-old songs constitute a big part of the "long tail"—the huge inventory of items that each

MUSIC 101: Pandora offers a running commentary on its songs and artists.

sell in very small numbers yet collectively amount to a big part of the online marketplace. Mining that tail is one of the main jobs of any recommender system.

"In book publishing, genres are the equivalent of what we're doing. A brand-new author can say, 'Mine's a historical mystery novel,' and thus put data into the product without having any customer reviews," Westergren says. "But our theory is that it's not good data, not granular enough and not objective."

Tom Conrad, the chief technical officer of Pandora, says that "musical genomes" sometimes turn up connections you'd probably never get with other methods. He cites the '80s pop star Cindy Lauper, who recently recorded a new record that didn't sell in great numbers. "We analyzed it for its genome and found that the record sounds an awful lot like Norah Jones. So we are able to play Lauper's songs when you start a Norah Jones song. There's a Metallica ballad that's musically a nice fit for Indigo Girls. So start an Indigo Girls station and you might get this ballad."

Conrad says that Pandora isn't so proud of its expert-rated system that it can't learn from the collaborative-filtering techniques pioneered at Amazon, Apple, Netflix, and other firms. He contends that the two approaches are complementary.

"We have benefited by peering inside the approaches tried by some of the thinking that went into the Netflix Prize competition, and we've incorporated some of the ideas into our own system," Conrad says. "I'm friendly with the Netflix personalization team; we've talked over the past two years or so. We wanted to have more qualitative information; they wanted more quantitative. Now we both use both. Netflix has human editors who try to capture technical aspects of the movies."

When the two approaches meet, experts will use computers as much as computers use experts. We will have achieved the perfect chimera: a man-machine mind meld. —*Philip E. Ross*