ARTIFICIAL INTELLIGENCE

MACHINES THAT THINK FOR THEMSELVES

New techniques for teaching computers how to learn are beating the experts

By Yaser S. Abu-Mostafa

Yaser S. Abu-Mostafa is a professor of electrical engineering and computer science at the California Institute of Technology.

A COUPLE OF YEARS AGO THE DIRECTORS OF A WOMEN'S clothing company asked me to help them develop better fashion recommendations for their clients. No one in their right mind would seek my personal advice in an area I know so little about—I am, after all, a male computer scientist—but they were not asking for my personal advice. They were asking for my machine-learning advice, and I obliged. Based purely on sales figures and client surveys, I was able to recommend to women whom I have never met fashion items I have never seen. My recommendations beat the performance of professional stylists. Mind you, I still know very little about women's fashion.

Machine learning is a branch of computer science that enables computers to learn from experience, and it is everywhere. It makes Web searches more relevant, blood tests more accurate and dating services more likely to find you a potential mate. At its simplest, machine-learning algorithms take an existing data set, comb through it for patterns, then use these patterns to generate predictions about the future. Yet advances in machine learning over the past decade have transformed the field. Indeed, machine-learning techniques are responsible for making computers "smarter" than humans at so many of the tasks we wish to pursue. Witness Watson, the IBM computer system that used machine learning to beat the best *Jeopardy* players in the world.

The most important machine-learning competition did not involve talking *Jeopardy*-playing machines, however. A few years ago Netflix, the online movie rental company, wanted to help its customers find movies that they would love—especially films that were not high-demand "new release" titles but rather from their largely ignored back catalogue. The company already had an in-house movie recommendation system, but executives knew it was far from perfect. So the company launched a competition to improve on existing efforts. The rules were simple: the first entry to beat the performance of the in-house system by 10 percent would earn a $1-million prize. Tens of thousands of people from around the world signed up.

For a machine-learning researcher, the competition was a dream (and not just for the prize money, attractive though it was). The most critical components of any machine-learning system are the data, and the Netflix prize offered 100 million points of real data, ready to download.

TRAINING DAYS

THE NETFLIX COMPETITION lasted for almost three years. Many groups attacked the problem by breaking down individual movies into long arrays of different attributes. For example, you could score any movie on various traits, such as how funny it is, how complicated it is or how attractive the actors are. For each viewer, you go back and take a look at the movies he has reviewed to see how much he values each of these attributes—how much he enjoys comedy, whether he prefers simple or complicated plots, and how much he likes to look at attractive movie stars [*see box on page 81*].

Now prediction becomes a simple matter of matching the

IN BRIEF

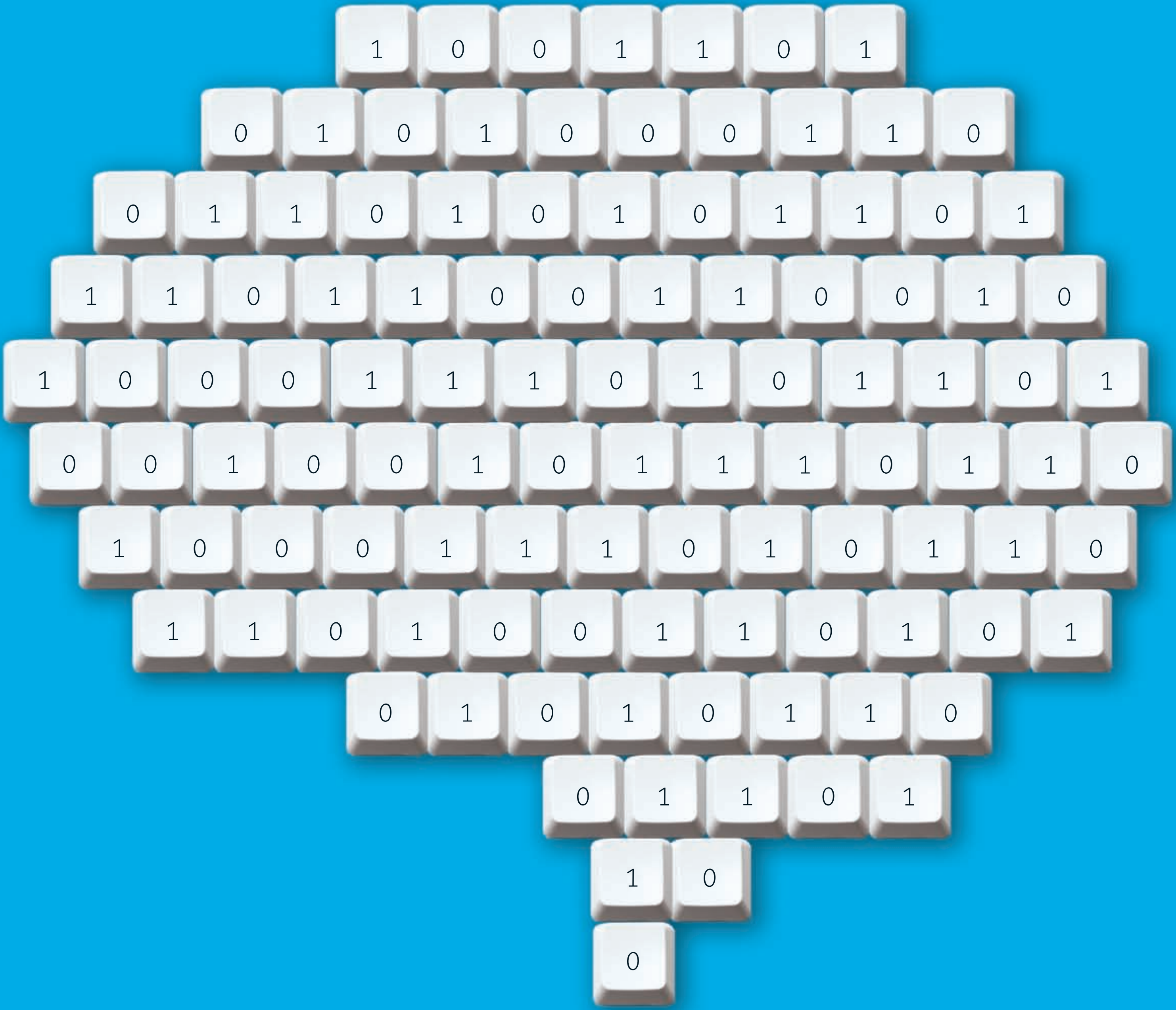
Machine learning is a branch of computer science that combs through data sets to make predictions about the future.

It is used to identify economic trends, personalize recommendations and build computers that appear to think.

Although machine learning has become incredibly popular, it only works on problems with large data sets.

Practitioners of machine learning must be careful to avoid having machines identify patterns that do not really exist.

PHOTOILLUSTRATION BY SCIENTIFIC AMERICAN


1 0 0 1 1 0 1
0 1 0 1 0 0 0 1 1 0
0 1 1 0 1 0 1 0 1 1 0 1
1 1 0 1 1 0 0 1 1 0 0 1 0
1 0 0 0 1 1 1 0 1 0 1 1 0 1
0 0 1 0 0 1 0 1 1 1 0 1 1 0
1 0 0 0 1 1 1 0 1 0 1 1 0
1 1 0 1 0 0 1 1 0 1 0 1
0 1 0 1 0 1 1 0
0 1 1 0 1
1 0
0

viewer's tastes to the new movie's attributes. If he loves comedies and complex plots, chances are he might like a knotty caper such as *Some Like It Hot* or *A Fish Called Wanda*. After the algorithm matches dozens of these attributes, the final recommendation should be a good predictor of how the viewer will like the movie.

We naturally think in easily identifiable attributes such as "comedy" or "complex plot," but algorithms need make no such distinctions. In fact, the entire process is automated—researchers never bother with analyzing movie content. The machine-learning algorithm will start with random, nameless attributes. As it gets data about how viewers rated movies in the past, it fine-tunes attributes until they correspond to how viewers rate movies.

For example, if people who like movie A also tend to like movies B, C and D, the algorithm will come up with a new attribute that is common to A, B, C and D. This happens in the so-called training phase, where the computer searches through millions of viewer ratings. The goal of this phase is to create a set of objective attributes that are based on actual ratings, not on subjective analysis.

It may be hard to interpret the different attributes that the machine-learning algorithm produces; they may not be as straightforward as "comedy content." In fact, they can be quite subtle, even incomprehensible, because the algorithm is only trying to find the best way to predict how a viewer would rate a movie, not necessarily explain to us how it is done. If a system performs well, we do not insist on understanding how it does so.

This is not the way the world is used to operating. Early in my career I developed a credit-approval system for a bank. When I was done, the bank wanted me to interpret what each attribute meant. The request had nothing to do with the system's performance, which was fine. The reason was legal: banks cannot deny credit to someone without articulating a rationale, and they cannot just send a letter to someone saying that the application was denied because *X* is less than 0.5.

Different machine-learning systems will develop unique sets of attributes. In the final weeks of the Netflix competition, groups that had been working independently began to blend their algorithms using so-called aggregation techniques. In the final hour of the three-year competition, two teams were still fighting for the top prize. The scoreboard showed a slight edge to The Ensemble, a team that included a Ph.D. alumnus of my research group at the California Institute of Technology, over BellKor's Pragmatic Chaos. Yet the final audited tally put the teams in a statistical dead heat—each achieved a 10.06 percent improvement over the original algorithm. According to the rules of the competition, in the event of a tie the award would go to the team that submitted its solution first. After three years of competition and in the last hour of battle, BellKor's Pragmatic Chaos submitted its solution 20 minutes earlier than The Ensemble. A 20-minute delay in a three-year competition made a difference of a million bucks.

THE PERFECT FIT

THE TYPE OF MACHINE LEARNING used in the movie-rating competition is called supervised learning. It is also used in tasks such as medical diagnosis. For example, we could provide a computer with thousands of images of white blood cells from patients' historical records, along with information about whether each image is of a cancerous or noncancerous cell. From this information, the algorithm will learn to apply certain cell attributes—shape, size and color, perhaps—to identify malignant cells. Here the researcher "supervises" the learning process. For each image in the training data, he or she gives the computer the correct answer.

Supervised learning is the most common type of machine learning, but it is not the only one. Roboticists, for example, may not know the best way to make a two-legged robot walk. In that case, they could design an algorithm that experiments with a number of different gaits. If a particular gait makes the robot fall down, the algorithm learns to not do that any more.

This is the reinforcement-learning approach. It is basically trial and error—a learning strategy we are all familiar with. In a typical reinforcement-learning scenario—human or machine—we face a situation in which some action is needed. Instead of someone telling us what to do, we try something and see what happens. Based on what happens, we reinforce the good actions and avoid the bad actions in the future. Eventually both we and the machines learn the correct actions for different situations.

For example, consider Internet search engines. The founders of Google did not wade through the Web circa 1997 to train its computers to recognize pages about, say, "Dolly the sheep." Instead their algorithms crawled the Web to generate a first draft of results, then they relied on user clicks to reinforce which pages were relevant and which were not. When users click on a page link in the search results, the machine-learning algorithm learns that the page is relevant. If users ignore a link that appears at the top of the search results, the algorithm infers that the page is not relevant. The algorithm combines such feedback from millions of users to adjust how it evaluates pages in future searches.

EXCESS PROBLEMS

RESEARCHERS often use reinforcement learning for tasks that require a sequence of actions, such as playing a game. Consider a simple example, like tic-tac-toe. The computer may start by randomly putting an X in a corner. This is a strong move, and the computer will go on to win these games more often than the games that it opens by placing an X on a side. The action that leads to a win—X in the corner—gets reinforced. Researchers then extend this process to infer what the correct action would be at any future step of the game—and for any game, from checkers to Go. Reinforcement learning is also used in advanced economics applications, such as finding a Nash equilibrium.

Sometimes even reinforcement learning is too much to ask for, because we are unable to get feedback on our actions. In such cases, we must turn to "unsupervised learning." Here the researcher has a set of data but no information about what action should be taken—either explicitly, as in supervised learning, or implicitly, as in reinforcement learning. How could we possibly learn from these data? A first step to making sense of it is to categorize the data into groups based on similarity. This is called clustering. It collects unlabeled data and infers information about their hidden structure. Clustering provides us with a better understanding of the data before we consider what action should be taken. Sometimes clustering is enough—if we want to organize a library, simply grouping books into similar categories is all we need to do. At other times, we might go further and apply supervised learning to the clustered data.

Ironically, the biggest trap that machine-learning practitioners fall into is to throw too much computing power at a problem. Recognizing this fact and being able to deal with it properly are what separate the professionals from the amateurs.

How can more power hurt? Machine-learning algorithms try to detect patterns in the data. If the algorithm is too aggressive—perhaps using too sophisticated a model to fit a limited data sample—it may mislead itself by detecting spurious patterns that happen by coincidence in a sample but do not reflect a true association. A significant part of the research on the mathematical theory of machine learning focuses on this problem of "overfitting" the data. We want to detect genuine connections that fit the data, but we do not want to overdo it and end up picking patterns that cannot be trusted.

To understand how this can happen, imagine a gambler at a roulette table (for the sake of simplicity, we will assume this table has only red and black numbers and does not include 0 or 00). She watches 10 consecutive spins alternate between red and black. "The wheel must be biased," she thinks. "It always goes red, black, red, black, red, black." The player has created a model in her head that the limited data set has confirmed. Yet on the 11th roll, right after she puts down $100 on red, the random nature of the wheel reasserts itself. The wheel stops at black for the second consecutive time, and she loses it all.

Our gambler was looking for a pattern where none really exists. Statistically, any roulette table has about a one in 500 chance of randomly flip-flopping between red and black 10 times in a row. In roulette, however, past spins have no bearing on the future. The next spin always has a 50 percent chance of coming up red. In machine learning, we have an old saying: if you torture the data long enough, it will confess.

HOW IT WORKS

Rated X (and Y and Z)

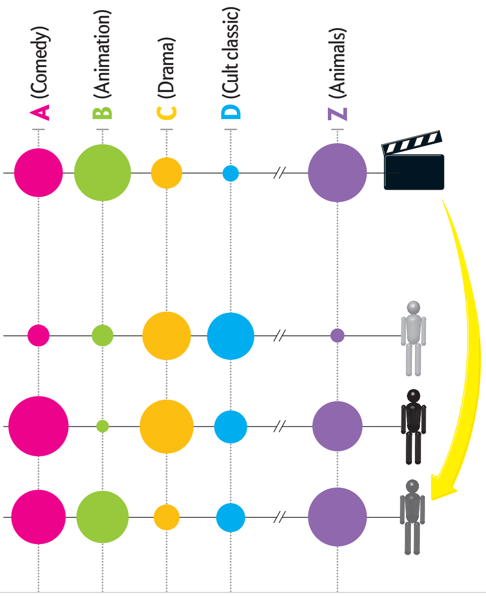
What movie should you watch tonight? Personalized recommendation engines help millions of people narrow the universe of potential films to fit their unique tastes. These services depend on a machine-learning strategy called singular value decomposition, which breaks down movies into long lists of attributes and matches these attributes to a viewer's preferences. The technique can be extended to just about any recommendation system, from Internet search engines to dating sites.

Turn Movies into Data

First a recommendation engine takes a huge data set of films and viewer ratings. Then it uses the collective ratings to break down individual movies into long lists of attributes. The resulting attributes may correspond to easily identifiable qualities such as "comedy" or "cult classic," but they may not—the computer knows them only as X, Y and Z.

Match Viewers to Movies

Now recommendation is a simple matter of decoding an individual's tastes and matching those tastes to the relevant movies. If in the past a person has enjoyed comedies with animals—or with unnamed mystery quality X—the recommendation engine will find similar films.



To avoid this outcome, machine-learning algorithms are biased to keep the models as simple as possible using a technique called regularization. The more complex a model is, the more prone it is to overfitting; regularization keeps that complexity in check.

Researchers will also commonly validate the algorithm on data that are not in the training set. In this way, we ensure that the performance we are getting is genuine, not just an artifact of the training data. The Netflix prize, for instance, was not judged against the original data set provided to the participants. It was tested on a new data set known only to the people at Netflix.

PREDICTING THE FUTURE

IT IS DIFFICULT to get bored if you work in machine learning. You never know what application you could be working on next. Machine learning enables nonexperts in an application area—computer scientists in women's fashion, for example—to learn and predict based merely on data. As a consequence, interest in the field is exploding. This past spring students from 15 different majors took my machine-learning course at Caltech. For the first time, I also posted course materials online and broadcast live videos of the lectures; thousands of people from around the world watched and completed the assignments. (You can, too: see the link below in the More to Explore.)

Machine learning, however, works only for problems that have enough data. Anytime I am presented with a possible machine-learning project, my first question is simple: What data do you have? Machine learning does not create information; it gets the information from the data. Without enough training data that contain proper information, machine learning will not work.

Yet data for myriad fields are becoming ever more abundant, and with them the value of machine learning will continue to rise. Trust me on this—predictions are my specialty. SA

MORE TO EXPLORE

Machines That Learn from Hints. Yaser S. Abu-Mostafa in *Scientific American,* Vol. 272, No. 4, pages 64–69; April 1995.

Recommend a Movie, Win a Million Bucks. Joseph Sill in *Engineering & Science,* Vol. 73, No. 2, pages 32–39; Spring 2010.

Learning from Data. Yaser S. Abu-Mostafa, Malik Magdon-Ismail and Hsuan-Tien Lin. AMLbook, 2012. **http://amlbook.com**

Learning from Data (online course): **http://work.caltech.edu/telecourse.html**

SCIENTIFIC AMERICAN ONLINE
For an interactive look at how movie-recommendation systems work, visit **ScientificAmerican.com/jul2012/rated-x**

Illustration by Jen Christiansen