nothing to do with medical diagnosis. Another simple learner, called the nearest-neighbor algorithm, has been used for everything from handwriting recognition to controlling robot hands to recommending books and movies you might like. And decision tree learners are equally apt at deciding whether your credit-card application should be accepted, finding splice junctions in DNA, and choosing the next move in a game of chess.

Not only can the same learning algorithms do an endless variety of different things, but they're shockingly simple compared to the algorithms they replace. Most learners can be coded up in a few hundred lines, or perhaps a few thousand if you add a lot of bells and whistles. In contrast, the programs they replace can run in the hundreds of thousands or even millions of lines, and a single learner can induce an unlimited number of different programs.

If so few learners can do so much, the logical question is: Could one learner do everything? In other words, could a single algorithm learn all that can be learned from data? This is a very tall order, since it would ultimately include everything in an adult's brain, everything evolution has created, and the sum total of all scientific knowledge. But in fact all the major learners—including nearest-neighbor, decision trees, and Bayesian networks, a generalization of Naïve Bayes—are universal in the following sense: if you give the learner enough of the appropriate data, it can approximate any function arbitrarily closely—which is math-speak for learning anything. The catch is that "enough data" could be infinite. Learning from finite data requires making assumptions, as we'll see, and different learners make different assumptions, which makes them good for some things but not others.

But what if instead of leaving these assumptions embedded in the algorithm we make them an explicit input, along with the data, and allow the user to choose which ones to plug in, perhaps even state new ones? Is there an algorithm that can take in any data and assumptions and output the knowledge that's implicit in them? I believe so. Of course, we have to put some limits on what the assumptions can be, otherwise we could cheat by giving the algorithm the entire target knowledge, or close to it, in the form of assumptions. But there are many ways to do this, from limiting the size of the input to requiring that the assumptions be no stronger than those of current learners.

The question then becomes: How weak can the assumptions be and still allow all relevant knowledge to be derived from finite data? Notice the word *relevant*: we're only interested in knowledge about our world, not about worlds that don't exist. So inventing a universal learner boils down to discovering the deepest regularities in our universe, those that all phenomena share, and then figuring out a computationally efficient way to combine them with data. This requirement of computational efficiency precludes just using the laws of physics as the regularities, as we'll see. It does not, however, imply that the universal learner has to be as efficient as more specialized ones. As so often happens in computer science, we're willing to sacrifice efficiency for generality. This also applies to the amount of data required to learn a given target knowledge: a universal learner will generally need more data than a specialized one, but that's OK provided we have the necessary amount—and the bigger data gets, the more likely this will be the case.

Here, then, is the central hypothesis of this book:

> *All knowledge—past, present, and future—can be derived from data by a single, universal learning algorithm.*

I call this learner the Master Algorithm. If such an algorithm is possible, inventing it would be one of the greatest scientific achievements of all time. In fact, the Master Algorithm is the last thing we'll ever have to invent because, once we let it loose, it will go on to invent everything else that can be invented. All we need to do is provide it with enough of the right kind of data, and it will discover the corresponding knowledge. Give it a video stream, and it learns to see. Give it a library, and it learns to read. Give it the results of physics experiments, and it discovers the laws of physics. Give it DNA crystallography data, and it discovers the structure of DNA.

This may sound far-fetched: How could one algorithm possibly learn so many different things and such difficult ones? But in fact many lines of evidence point to the existence of a Master Algorithm. Let's see what they are.

## The argument from neuroscience

In April 2000, a team of neuroscientists from MIT reported in *Nature* the results of an extraordinary experiment. They rewired the brain of a ferret, rerouting the connections from the eyes to the auditory cortex (the part of the brain responsible for processing sounds) and rerouting the connections from the ears to the visual cortex. You'd think the result would be a severely disabled ferret, but no: the auditory cortex learned to see, the visual cortex learned to hear, and the ferret was fine. In normal mammals, the visual cortex contains a map of the retina: neurons connected to nearby regions of the retina are close to each other in the cortex. Instead, the rewired ferrets developed a map of the retina in the auditory cortex. If the visual input is redirected instead to the somatosensory cortex, responsible for touch perception, it too learns to see. Other mammals also have this ability.

In congenitally blind people, the visual cortex can take over other brain functions. In deaf ones, the auditory cortex does the same. Blind people can learn to "see" with their tongues by sending video images from a head-mounted camera to an array of electrodes placed on the tongue, with high voltages corresponding to bright pixels and low voltages to dark ones. Ben Underwood was a blind kid who taught himself to use echolocation to navigate, like bats do. By clicking his tongue and listening to the echoes, he could walk around without bumping into obstacles, ride a skateboard, and even play basketball. All of this is evidence that the brain uses the same learning algorithm throughout, with the areas dedicated to the different senses distinguished only by the different inputs they are connected to (e.g that problem, based on ideas from its allied fields of science, and it has a master algorithm that embodies it.

For symbolists, all intelligence can be reduced to manipulating symbols, in the same way that a mathematician solves equations by replacing expressions by other expressions. Symbolists understand that you can't learn from scratch: you need some initial knowledge to go with the data. They've figured out how to incorporate preexisting knowledge into learning, and how to combine different pieces of knowledge on the fly in order to solve new problems. Their master algorithm is inverse deduction, which figures out what knowledge is missing in order to make a deduction go through, and then makes it as general as possible.

For connectionists, learning is what the brain does, and so what we need to do is reverse engineer it. The brain learns by adjusting the strengths of connections between neurons, and the crucial problem is figuring out which connections are to blame for which errors and changing them accordingly. The connectionists' master algorithm is backpropagation, which compares a system's output with the desired one and then successively changes the connections in layer after layer of neurons so as to bring the output closer to what it should be.

Evolutionaries believe that the mother of all learning is natural selection. If it made us, it can make anything, and all we need to do is simulate it on the computer. The key problem that evolutionaries solve is learning structure: not just adjusting parameters, like backpropagation does, but creating the brain that those adjustments can then fine-tune. The evolutionaries' master algorithm is genetic programming, which mates and evolves computer programs in the same way that nature mates and evolves organisms.

Bayesians are concerned above all with uncertainty. All learned knowledge is uncertain, and learning itself is a form of uncertain inference. The problem then becomes how to deal with noisy, incomplete, and even contradictory information without falling apart. The solution is probabilistic inference, and the master algorithm is Bayes' theorem and its derivates. Bayes' theorem tells us how to incorporate new evidence into our beliefs, and probabilistic inference algorithms do that as efficiently as possible.

For analogizers, the key to learning is recognizing similarities between situations and thereby inferring other similarities. If two patients have similar symptoms, perhaps they have the same disease. The key problem is judging how similar two things are. The analogizers' master algorithm is the support vector machine, which figures out which experiences to remember and how to combine them to make new predictions.

Each tribe's solution to its central problem is a brilliant, hard-won advance. But the true Master Algorithm must solve all five problems, not just one. For example, to cure cancer we need to understand the metabolic networks in the cell: which genes regulate which others, which chemical reactions the resulting proteins control, and how adding a new molecule to the mix would affect the network. It would be silly to try to learn all of this from scratch, ignoring all the knowledge that biologists have painstakingly accumulated over the decades. Symbolists know how to combine this knowledge with data from DNA sequencers, gene expression microarrays, and so on, to produce results that you couldn't get with either alone. But the knowledge we obtain by inverse deduction is purely qualitative; we need to learn not just who interacts with whom, but how much, and backpropagation can do that. Nevertheless, both inverse deduction and backpropagation would be lost in space without some basic structure on which to hang the interactions and parameters they find, and genetic programming can discover it. At this point, if we had complete knowledge of the metabolism and all the data relevant to a given patient, we could figure out a treatment for her. But in reality the information we have is always very incomplete, and even incorrect in places; we need to make headway despite that, and that's what probabilistic inference is for. In the hardest cases, the patient's cancer looks very different from previous ones, and all our learned knowledge fails. Similarity-based algorithms can save the day by seeing analogies between superficially

The molecular biology of living cells is such a mess that molecular biologists often quip that only people who don't know any of it could believe in intelligent design. The architecture of the brain may well have similar faults—the brain has many constraints that computers don't, like very limited short-term memory—and there's no reason to stay within them. Moreover, we know of many situations where humans seem to consistently do the wrong thing, as Daniel Kahneman illustrates at length in his book *Thinking, Fast and Slow*.

In contrast to the connectionists and evolutionaries, symbolists and Bayesians do not believe in emulating nature. Rather, they want to figure out from first principles what learners should do—and that includes us humans. If we want to learn to diagnose cancer, for example, it's not enough to say "this is how nature learns; let's do the same." There's too much at stake. Errors cost lives. Doctors should diagnose in the most foolproof way they can, with methods similar to those mathematicians use to prove theorems, or as close to that as they can manage, given that it's seldom possible to be that rigorous. They need to weigh the evidence to minimize the chances of a wrong diagnosis; or more precisely, so that the costlier an error is, the less likely they are to make it. (For example, failing to find a tumor that's really there is potentially much worse than inferring one that isn't.) They need to make *optimal* decisions, not just decisions that seem good.

This is an instance of a tension that runs throughout much of science and philosophy: the split between descriptive and normative theories, between "this is how it is" and "this is how it should be." Symbolists and Bayesians like to point out, however, that figuring out how we should learn can also help us to understand how we do learn because the two are presumably not entirely unrelated—far from it. In particular, behaviors that are important for survival and have had a long time to evolve should not be far from optimal. We're not very good at answering written questions about probabilities, but we are very good at instantly choosing hand and arm movements to hit a target. Many psychologists have used symbolist or Bayesian models to explain aspects of human behavior. Symbolists dominated the first few decades of cognitive psychology. In the 1980s and 1990s, connectionists held sway, but now Bayesians are on the rise.

For the hardest problems—the ones we really want to solve but haven't been able to, like curing cancer—pure nature-inspired approaches are probably too uninformed to succeed, even given massive amounts of data. We can in principle learn a complete model of a cell's metabolic networks by a combination of structure search, with or without crossover, and parameter learning via backpropagation, but there are too many bad local optima to get stuck in. We need to reason with larger chunks, assembling and reassembling them as needed and using inverse deduction to fill in the gaps. And we need our learning to be guided by the goal of optimally diagnosing cancer and finding the best drugs to cure it.

Optimal learning is the Bayesians' central goal, and they are in no doubt that they've figured out how to reach it. This way, please . . .

Bayesians and symbolists agree that prior assumptions are inevitable, but they differ in the kinds of prior knowledge they allow. For Bayesians, knowledge goes in the prior distribution over the structure and parameters of the model. In principle, the parameter prior could be anything we please, but ironically, Bayesians tend to choose uninformative priors (like assigning the same probability to all hypotheses) because they're easier to compute with. In any case, humans are not very good at estimating probabilities. For structure, Bayesian networks provide an intuitive way to incorporate knowledge: draw an arrow from A to B if you think that A directly causes B. But symbolists are much more flexible: you can provide as prior knowledge to your learner anything you can encode in logic, and practically anything can be encoded in logic—provided it's black and white.

Clearly, we need both logic and probability. Curing cancer is a good example. A Bayesian network can model a single aspect of how cells function, like gene regulation or protein folding, but only logic can put all the pieces together into a coherent picture. On the other hand, logic can't deal with incomplete or noisy information, which is pervasive in experimental biology, but Bayesian networks can handle it with aplomb.

Bayesian learning works on a single table of data, where each column represents a variable (for example, the expression level of one gene) and each row represents an instance (for example, a single microarray experiment, with each gene's observed expression level). It's OK if the table has "holes" and measurement errors because we can use probabilistic inference to fill in the holes and average over the errors. But if we have more than one table, Bayesian learning is stuck. It doesn't know how to, for example, combine gene expression data with data about which DNA segments get translated into proteins, and how in turn the three-dimensional shapes of those proteins cause them to lock on to different parts of the DNA molecule, affecting the expression of other genes. In logic, we can easily write rules relating all of these aspects, and learn them from the relevant combinations of tables—but only provided the tables have no holes or errors.

Combining connectionism and evolutionism was fairly easy: just evolve the network structure and learn the parameters by backpropagation. But unifying logic and probability is a much harder problem. Attempts to do it go all the way back to Leibniz, who was a pioneer of both. Some of the best philosophers and mathematicians of the nineteenth and twentieth centuries, like George Boole and Rudolf Carnap, worked hard on it but ultimately didn't get very far. More recently, computer scientists and AI researchers have joined the fray. But as the millennium turned around, the best we had were partial successes, like adding some logical constructs to Bayesian networks. Most experts believed that unifying logic and probability was impossible. The prospects for a Master Algorithm did not look good, particularly since the existing evolutionary and connectionist algorithms couldn't deal with incomplete information or multiple data sets, either.

Luckily, we have since cracked the problem, and the Master Algorithm now looks that much closer. We'll see how we did it in Chapter 9 and take it from there. But first we need to gather a very important, still-missing piece of the puzzle: how to learn from very little data. That might seem unnecessary in these days of data deluge, but the truth is that we often find ourselves with reams of data about some parts of the problem we want to solve and almost none about others. This is where one of the most important ideas in machine learning comes in: analogy. All of the tribes we've met so far have one thing in common: they learn an explicit model of the phenomenon under consideration, whether it's a set of rules, a multilayer perceptron, a genetic program, or a Bayesian network. When they don't have enough data to do that, they're stumped. But analogizers can learn from as little as one example because they never form a model. Let's see what they do instead.

answer correctly in the future, take a moment to trawl through the later results pages for the relevant links and click on them. More generally, if a system keeps recommending the wrong things to you, try teaching it by finding and clicking on a bunch of the right ones and come back later to see if it did.

That could be a lot of work, though. What all of these illustrate, unfortunately, is how narrow the communication channel between you and the learner is today. You should be able to tell it as much as you want about yourself, not just have it learn indirectly from what you do. More than that, you should be able to inspect the learner's model of you and correct it as desired. The learner can still decide to ignore you, if it thinks you're lying or are low on self-knowledge, but at least it would be able to take your input into account. For this, the model needs to be in a form that humans can understand, such as a set of rules rather than a neural network, and it needs to accept general statements as input in addition to raw data, as Alchemy does. All of which brings us to the question of how good a model of you a learner can have and what you'd want to do with that model.

## The digital mirror

Take a moment to consider all the data about you that's recorded on all the world's computers: your e-mails, Office docs, texts, tweets, and Facebook and LinkedIn accounts; your web searches, clicks, downloads, and purchases; your credit, tax, phone, and health records; your Fitbit statistics; your driving as recorded by your car's microprocessors; your wanderings as recorded by your cell phone; all the pictures of you ever taken; brief cameos on security cameras; your Google Glass snippets—and so on and so forth. If a future biographer had access to nothing but this "data exhaust" of yours, what picture of you would he form? Probably a quite accurate and detailed one in many ways, but also one where some essential things would be missing. Why did you, one beautiful day, decide to change careers? Could the biographer have predicted it ahead of time? What about that person you met one day and secretly never forgot? Could the biographer wind back through the found footage and say "Ah, there"?

The sobering (or perhaps reassuring) thought is that no learner in the world today has access to all this data (not even the NSA), and even if it did, it wouldn't know how to turn it into a real likeness of you. But suppose you took all your data and gave it to the—real, future—Master Algorithm, already seeded with everything we could teach it about human life. It would learn a model of you, and you could carry that model in a thumb drive in your pocket, inspect it at will, and use it for everything you pleased. It would surely be a wonderful tool for introspection, like looking at yourself in the mirror, but it would be a digital mirror that showed not just your looks but all things observable about you—a mirror that could come alive and converse with you. What would you ask it? Some of the answers you might not like, but that would be all the more reason to ponder them. And some would give you new ideas, new directions. The Master Algorithm's model of you might even help you become a better person.

Self-improvement aside, probably the first thing you'd want your model to do is negotiate the world on your behalf: let it loose in cyberspace, looking for all sorts of things for you. From all the world's books, it would suggest a dozen you might want to read next, with more insight than Amazon could dream of. Likewise for movies, music, games, clothes, electronics—you name it. It would keep your refrigerator stocked at all times, natch. It would filter your e-mail, voice mail, Facebook posts, and Twitter feed and, when appropriate, reply on your behalf. It would take care of all the little annoyances of modern life for you, like checking credit-card bills, disputing improper charges, making arrangements, renewing subscriptions, and filling out tax returns. It would find a remedy for your ailment, run it by your doctor, and order it from Walgreens. It would bring interesting job opportunities to your attention, propose vacation spots, suggest which candidates to vote for on the ballot, and screen potential dates. And, after the match was made, it would team up with your date's model to pick some restaurants you might both like. Which is where things start to get *really* interesting.

## A society of models

In this rapidly approaching future, you're not going to be the only one with a "digital half" doing your bidding twenty-four hours a day. Everyone will have a detailed model of him- or herself, and these models will talk to each other all the time. If you're looking for a job and company X is looking to hire, its model will interview your model. It will be a lot like a real, flesh-and-blood interview—your model will still be well advised to not volunteer negative information about you, and so on—but it will take only a fraction of a second. You'll click on "Find Job" in your future LinkedIn account, and you'll immediately interview for every job in the universe that remotely fits your parameters (profession, location, pay, etc.). LinkedIn will respond on the spot with a ranked list of the best prospects, and out of those, you'll pick the first company that you want to have a chat with. Same with dating: your model will go on millions of dates so you don't have to, and come Saturday, you'll meet your top prospects at an OkCupid-organized party, knowing that you're also one of *their* top prospects—and knowing, of course, that their *other* top prospects are also in the room. It's sure to be an interesting night.

In the world of the Master Algorithm, "my people will call your people" becomes "my program will call your program." Everyone has an entourage of bots, smoothing his or her way through the world. Deals get pitched, terms negotiated, arrangements made, all before you lift a finger. Today, drug companies target your doctor, because he decides what drugs to prescribe to you. Tomorrow, the purveyors of every product and service you use, or might use, will target your model, because your model will screen them for you. Their bots' job is to get your bot to buy. Your bot's job is to see through their claims, just as you see through TV commercials, but at a much finer level of detail, one that you'd never have the time or patience for. Before you buy a car, the digital you will go over every one of its specs, discuss them with the manufacturer, and study everything anyone in the world has said about that car and its alternatives. Your digital half will be like power steering for your life: it goes where you want to go but with less effort from you. This does not mean that you'll end up in a "filter bubble," seeing only what you reliably like, with no room for the unexpected; the digital you knows better than that. Part of its brief is to leave some things open to chance, to expose you to new experiences, and to look for serendipity.

Even more interesting, the process doesn't end when you find a car, a house, a doctor, a date, or a job. Your digital half is continually learning from its experiences, just as you would. It figures out what works and doesn't, whether it's in job interviews, dating, or real-estate hunting. It learns about the people and organizations it interacts with on your behalf and then (even more important) from your real-world interactions with them. It predicted Alice would be a great date for you, but you had an awkward time, so it hypothesizes possible reasons, which it will test on your next round of dating. It shares its most important findings with you. ("You believe you like X, but in reality you tend to go for Y.") Comparing your experiences of various hotels with their reviews on TripAdvisor, it figures out what the really telling tidbits are and looks for them in the future. It learns not just which online merchants are more trustworthy but how to decode what the less trustworthy ones say. Your digital half has a model of the world: not just of the world in general but of the world as it relates to you. At the same time, of course, everyone else also has a continually evolving model of his or her world. Every party to an interaction learns from it and applies what it's learned to its next interactions. You have your model of every person and organization you interact with, and they each have their model of you. As the models improve, their interactions become more and more like the ones you would have in the real world—except millions of times faster and in silicon. Tomorrow's cyberspace will be a vast parallel world that selects only the most promising things to try out in the real one. It will be like a new, global subconscious, the collective id of the human race.

## To share or not to share, and how and where

Of course, learning about the world all by yourself is slow, even if your digital half does it orders of magnitude faster than the flesh-and-blood probably have proposed a law of accelerating body size. Eukaryotes (us) evolve more slowly than prokaryotes (bacteria). Far from accelerating smoothly, evolution proceeds in fits and starts.

To sidestep the problem that infinitely dense points don't exist, Kurzweil proposes to instead equate the Singularity with a black hole's event horizon, the region within which gravity is so strong that not even light can escape. Similarly, he says, the Singularity is the point beyond which technological evolution is so fast that humans cannot predict or understand what will happen. If that's what the Singularity is, then we're already inside it. We can't predict in advance what a learner will come up with, and often we can't even understand it in retrospect. As a matter of fact, we've always lived in a world that we only partly understood. The main difference is that our world is now partly created by us, which is surely an improvement. The world beyond the Turing point will not be incomprehensible to us, any more than the Pleistocene was. We'll focus on what we can understand, as we always have, and call the rest random (or divine).

The trajectory we're on is not a singularity but a phase transition. Its critical point—the Turing point—will come when machine learning overtakes the natural variety. Natural learning itself has gone through three phases: evolution, the brain, and culture. Each is a product of the previous one, and each learns faster. Machine learning is the logical next stage of this progression. Computer programs are the fastest replicators on Earth: copying them takes only a fraction of a second. But creating them is slow, if it has to be done by humans. Machine learning removes that bottleneck, leaving a final one: the speed at which humans can absorb change. This too will eventually be removed, but not because we'll decide to hand things off to our "mind children," as Hans Moravec calls them, and go gently into the good night. Humans are not a dying twig on the tree of life. On the contrary, we're about to start branching.

In the same way that culture coevolved with larger brains, we will coevolve with our creations. We always have: humans would be physically different if we had not invented fire or spears. We are *Homo technicus* as much as *Homo sapiens*. But a model of the cell of the kind I envisaged in the last chapter will allow something entirely new: computers that design cells based on the parameters we give them, in the same way that silicon compilers design microchips based on their functional specifications. The corresponding DNA can then be synthesized and inserted into a "generic" cell, transforming it into the desired one. Craig Venter, the genome pioneer, has already taken the first steps in this direction. At first we will use this power to fight disease: a new pathogen is identified, the cure is immediately found, and your immune system downloads it from the Internet. *Health problems* becomes an oxymoron. Then DNA design will let people at last have the body they want, ushering in an age of affordable beauty, in William Gibson's memorable words. And then *Homo technicus* will evolve into a myriad different intelligent species, each with its own niche, a whole new biosphere as different from today's as today's is from the primordial ocean.

Many people worry that human-directed evolution will permanently split the human race into a class of genetic haves and one of have-nots. This strikes me as a singular failure of imagination. Natural evolution did not result in just two species, one subservient to the other, but in an infinite variety of creatures and intricate ecosystems. Why would artificial evolution, building on it but less constrained, do so?

Like all phase transitions, this one will eventually taper off too. Overcoming a bottleneck does not mean the sky is the limit; it means the next bottleneck is the limit, even if we don't see it yet. Other transitions will follow, some large, some small, some soon, some not for a long time. But the next thousand years could well be the most amazing in the life of planet Earth.