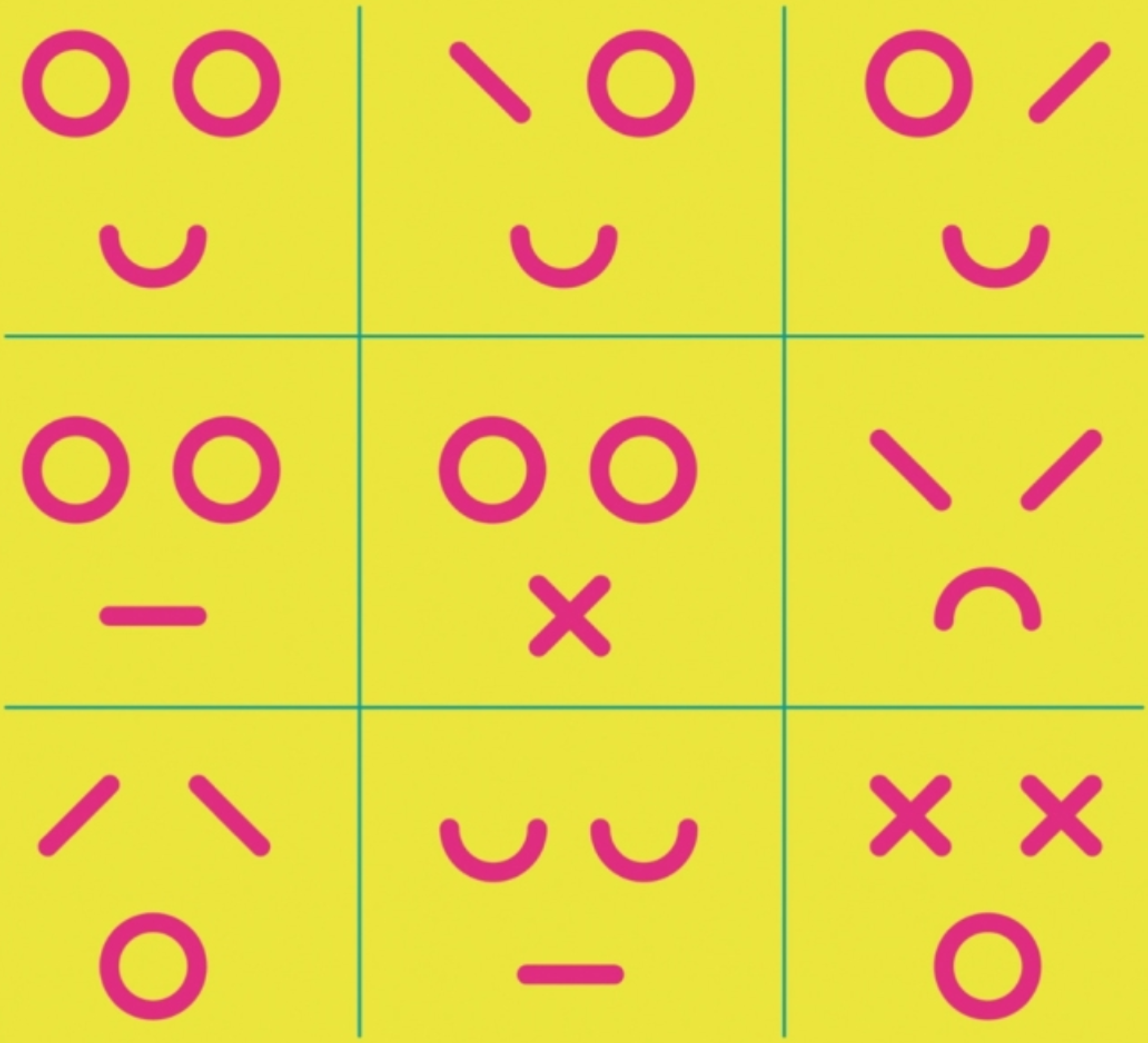
AWKWARD INTELLIGENCE

WHERE AI GOES WRONG, WHY IT MATTERS, AND WHAT WE CAN DO ABOUT IT



KATHARINA A. ZWEIG

TRANSLATED BY
NOAH HARLEY

Original title: *Ein Algorithmus hat kein Taktgefühl. Wo künstliche Intelligenz sich irrt, warum uns das betrifft und was wir dagegen tun können*, by Katharina Zweig

The translation of this work was supported by a grant from the Goethe-Institut.

GOETHE
INSTITUT

This book was set in Adobe Garamond Pro by Jen Jackowitz. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Names: Zweig, Katharina A., author. | Harley, Noah, translator.
Title: Awkward intelligence : where AI goes wrong, why it matters, and what we can do about it / Katharina A. Zweig ; translated by Noah Harley.
Description: Cambridge, Massachusetts : The MIT Press, [2022] | Includes bibliographical references and index.
Identifiers: LCCN 2021060550 | ISBN 9780262047463 | ISBN 9780262371858 (pdf) | ISBN 9780262371865 (epub)
Subjects: LCSH: Artificial intelligence. | Artificial intelligence—Philosophy.
Classification: LCC Q334.7 .Z84 2022 | DDC 006.301—dc23/eng20220521
LC record available at https://lccn.loc.gov/2021060550

10 9 8 7 6 5 4 3 2 1

CONTENTS

Preface ix

PART I THE TOOLKIT 1

1 ROBO-JUDGES . . . WITH POOR JUDGMENT 3

2 THE FACT FACTORIES OF THE NATURAL SCIENCES 13

PART II THE ABCs OF COMPUTER SCIENCE 25

3 ALGORITHMS: INSTRUCTIONS FOR COMPUTERS 27

4 BIG DATA AND DATA MINING 55

5 COMPUTER INTELLIGENCE 89

6 MACHINE LEARNING VERSUS PEOPLE 135

7 ARE WE LITERATE YET? 151

PART III THE PATH TO BETTER DECISIONS, WITH AND WITHOUT MACHINES 153

8 ALGORITHMS, DISCRIMINATION, AND IDEOLOGY 155

9 HOW TO STAY IN CONTROL 177

10 WHO WANTS MACHINES MAKING DECISIONS ABOUT PEOPLE, ANYWAY? 193

11 IT'S TIME FOR "THE TALK" ABOUT STRONG AI 205

Postscript 217
Glossary 223
Notes 229
Index 249

PREFACE

The most important thing about this book, dear reader, is you! That's because artificial intelligence, AI for short, will soon find its way into every corner of our lives and make decisions about, with, and for us. And for AI to make those decisions as well as possible, we all have to think about what actually goes into a good decision—and whether computers can make them in our stead. In what follows I take you on a backstage tour so that you can see for yourself just how many levers computer and data scientists are actually pulling to wrest decisions from data. And that's where you come in: what matters at moments like these is how *you* would decide. That's because society should leave its important decisions to machines only if it is confident those machines will behave according to its cultural and moral standards. This is why more than anything else, I want this book to empower you. I hope to dispel the sense of helplessness that creeps in when the conversation turns to algorithms; to explain the necessary terms and point out how and where you can intervene; and finally, to rouse you to action so that you can join computer scientists, politicians, and employers in debating where artificial intelligence makes sense—and where it doesn't.

And how is it that artificial intelligence will soon find its way into every corner of our lives, you ask? For one, because AI can make things more efficient by relieving us of the burdensome, endlessly repetitive parts of our work. Yet I also see a tendency at present toward thinking AI should make decisions about people. That might occur when using data to determine whether a job applicant should receive an interview or a person is fit enough for a medical study, for example, or if someone else may be predisposed to acts of terrorism.

How did we get here in the first place, to the point where it became possible for so many of us to entertain the notion that machines are better judges of people than we ourselves are? Well, for starters computers are clearly capable of processing data in quantities that humans cannot. What strikes me, however, is a present lack of faith in the human capacity to judge. It's not as though we first came to perceive humanity on the whole to be irrational, liable to manipulation, subjective, and prejudiced when Daniel Kahneman was awarded the Nobel Prize in 2002 for his research on human irrationality, or more recently with the introduction of Richard Thaler's concept of nudging in 2017.[1] In our dim view of human judgment, it is, of course, always other people who are the irrational ones—all the more so if they have utterly failed to appreciate us for the highly individual and complex beings we are![2] This in turn leads us to hope that machines will unerringly arrive at more objective decisions and then, with a bit of "magic," will discover patterns and rules in human behavior that have escaped the experts thus far, resulting in sounder predictions.

Where do such hopes spring from? In recent years, teams of developers have demonstrated that by using artificial intelligence, computers are able to solve tasks quickly and effectively that just two decades ago would have posed a real challenge. Every day, machines manage to sift through billions of websites to offer up the best results for our search queries or to detect partially concealed bicyclists and pedestrians in images and reliably predict their next movements; they've even beaten the reigning champions in chess and go. From here, doesn't it seem obvious that they could also support decision-makers in reaching fair judgements about people? Or that machines should simply make those judgements themselves?

Many expect this will make decisions more objective—something that is also sorely lacking on many counts. Take the United States, one country where algorithmic decision systems are already used in the lead-up to important human decisions. In a land that holds 20 percent of the official prison population worldwide, and where African Americans are roughly six times as likely to be imprisoned as white people, one could only wish for systems that would avoid any and all forms of latent racism—if possible, without having to raise spending significantly. This has led to the use of risk-assessment systems, which estimate the risk that someone with a previous conviction runs of becoming a repeat offender. The algorithms work by automatically analyzing properties that are common among known criminals who go on to commit another offense, and rare among those who don't. I found it deeply unsettling when my research was able to show that one commonly used algorithm in the US resulted in mistaken judgements up to 80 percent of the time (!) in the case of serious crimes. Concretely, this means that a mere one out of every four people the algorithm labeled as "high-risk repeat offenders" went on to commit another serious offense. Simple guesswork based on the general likelihood of recidivism would only have been slightly less accurate, and at least had the advantage of consciously being pure conjecture.

So what's going awry when machines judge people? As a scientist coming from a highly interdisciplinary background, I consider the effects and side effects of software from a particular angle: socioinformatics. A recent offshoot of computer science, as a discipline socioinformatics draws on methods and approaches from within psychology, sociology, economics, statistical physics, and (of course) computer science. The key argument is that interactions between users and software can only be understood when seen as part of a larger whole called a *sociotechnical system*.

For over fifteen years now, my research has focused concretely on how and when we can use computers—and, more specifically, exploit data, or perform data mining—to better understand the complex world we inhabit. That lands me among the ranks of those with the sexiest jobs on planet Earth, even if a weekend spent wading through endless streams of data, sifting for exciting correlations with statistics, may not exactly sound like your idea of fun.[3] Personally, I can't imagine anything better! Yet at the start of my career, I used statistics without really understanding it, always uncertain of whether this, that, or the other method could actually be applied to data to yield interpretable results. This was due to the fact that after graduating high school I initially chose to study biochemistry, a course of study that typically spends little time on mathematics. We learned the basics of biology, medicine, physics, and chemistry—but not a single hour of statistics. They were probably hoping it would seep into our brains by pure osmosis if only we cooked up enough of the lab experiments they assigned.

Later, I came to bioinformatics, an entirely new course at the time that taught us to design and apply methods for examining the biodata that was then piling up in droves. Yet here, too, statistics was missing. Nor for that matter did either course provide any instruction in scientific theory, a baffling and dangerous blind spot present in the curricula of nearly every discipline in the natural sciences that aims to produce facts.

Under such circumstances, it should come as little surprise that many computer scientists and engineers are all too sure about their methods obtaining the pure and unadulterated truth from data. Especially with data mining and machine learning (the basis for artificial intelligence), they purport to have discovered the magic formula for solving each and every complex problem. For someone unaware of the fact that she is simply busying herself with models and can never achieve certainty once and for all, it is all too easy to rush into pronouncements like the following:

> Imagine a world where you can maximize the potential of every moment of your life. Such a life would be productive, efficient, and powerful. You will (in effect) have superpowers—and a lot more spare time. Well, such a world may seem a little boring to people who like to take uncalculated risks, but not to a profit-generating organization. Organizations spend millions of dollars managing risk. And if there is something out there that helps them manage their risk, optimize their operations, and maximize their profits, you should definitely learn about it. That is the world of predictive analytics.[4]

And that was just the introduction to *Predictive Analytics for Dummies*! Things take a more serious turn when companies advertise data mining software for "predicting hiring success" with phrases like this: "In the end the predictive possibilities are virtually unlimited, provided the availability of good data . . . let's take the emotion out of the hiring process and replace it with a data-driven approach!"[5] The catchphrase *data-driven* reminds me, however—there's someone here who would like to introduce himself and offer his services. His name is Artie, and he'd like to serve as your fully data-driven companion throughout this book. Artie is an artificial intelligence (AI) and may still be a bit slow on the draw when it comes to truly understanding humans. But he gets an A for effort!

As I'm sure you've guessed from the preceding two quotes, I would caution against the bold confidence of some—and against placing too much trust in Artie. Over the course of this book, I point out the situations in which we shouldn't breezily accept the results of machine learning without further ado. I then go on to make concrete suggestions about when algorithmic decision-making (ADM) systems are unacceptable, whether out of technological or societal considerations. At the same time, it's important to understand the enormous potential of *data mining*, by which I mean processing data with algorithms. In situations where ADM systems are permissible, I thus point out specifically where we have to be on guard, while also

I

THE TOOLKIT

If you want to mess with artificial intelligence, you need the right tools. Going forward, the four tools described in this part will give you a method to work out the possible pitfalls if and when your boss or a state agency plans on using an algorithmic decision-making system—or, alternatively, to sound the all clear, because not everything that looks dangerous really is.

1

ROBO-JUDGES . . . WITH POOR JUDGMENT

It wasn't the first time I had sat there dumbfounded by the results of our research, but it was probably the most memorable. My student Tobias Krafft and I had just finished sifting through the predictions made by a special software used in US courtrooms. We were horrified by just how bad they were—and yet the state was using them in a pivotal setting. The basic idea of using algorithms to predict whether a person will commit a crime or not calls the film *Minority Report* to mind. The movie is based on a short story from 1956 by the famous science fiction writer Philip K. Dick. In it, Tom Cruise plays a policeman who is able to identify and arrest potential criminals before they commit a crime, aided by "precogs," people with the gift of clairvoyance. What was a bizarre tale had now become a reality, albeit one where the predictive machinery was sadly lacking in precision.

Unlike in the film, real-world predictive software can of course neither "see" the actual crime nor know its exact timing. Instead, the software is fed basic information about the criminals it is meant to evaluate: how often a person has been arrested in the past, what kind of crimes they've committed, and information about their age and gender.

The computer than calculates a risk score based on the information, which you might compare to risk categories in car insurance, where higher-risk drivers are grouped into one category and lower-risk drivers into another. Yet a funny thing happens when a person is sorted into a category. Even if the driver hasn't done anything herself (yet), she receives the same treatment as others in her class. If those drivers were involved in multiple accidents, say, she pays more; if not, she pays less. This means that when a driver is first assigned to a risk category, what she pays isn't based on her own individual behavior going forward, but the past behavior of the people she resembles. In this way, financial risk is distributed among everyone within the same class.

How does that work when it comes to crimes that may be committed in the future? Well, the principle is the same to begin with. The computer identifies properties common among criminals who become repeat offenders and uncommon among those who regain their footing in society. It then uses those properties to determine a person's risk factor. In the case of car insurance, risk factors include the driver's age and the number of consecutive years without an accident. This isn't necessarily fair, and certainly lacks in complexity. Wouldn't it be better to conduct a personality test, and only after that decide the person's risk category?

It's argued, of course, that people are classified according to such highly schematic and easily measurable properties for efficiency's sake. At least with car insurance, however, the process is fair to the extent that any driver receiving his or her license at sixteen begins at the same starting point. Any subsequent classification depends exclusively on the individual's driving record and not that of their generation.

That wasn't something Tobias or I could say for the classification method used in COMPAS, the risk assessment system we were researching. Aside from information about previous crimes, an additional questionnaire asked prisoners whether their parents or siblings had committed crimes, or their parents had divorced early. While those circumstances may well leave a mark, they are hardly something for which a person is responsible or might alter themselves.[1] A criminal is thus evaluated and assigned a risk category based on whichever properties the software company deems relevant. If that person lands in a risk category where many of the people have committed another crime in the past, the software assumes that this person, too, will become a repeat offender.

The algorithm is advertised on the merit that it results in the right decisions around 70 percent of the time.[2] That number alone struck both Tobias and me as disturbingly low for software used by a public authority in court. In medicine, for example, such a low percentage would be considered unacceptable. Yet now we found ourselves face to face with results that proved how many people assigned to the highest risk category did actually relapse. The number was in fact somewhat higher than 70 percent for criminal acts in general—but only around 25 percent when it came to violent crimes. That meant that only one out of every four people who set off clear alarm signals as liable to commit another serious act of violence actually did. What was more, other colleagues had shown the average layperson to be capable of predictions that were just about as accurate.[3]

I've spent the last three years trying to understand why anyone would want to use algorithms that make such bad predictions and why governments would want to commission or purchase them. I also wanted to answer the million-dollar question, of course: How can we develop better software? And might there not be situations where algorithms shouldn't make decisions about people in the first place?

But what does any of this have to do with you, gentle reader? Isn't it all so technical that there's no room for any say in the matter? Your and my experience both over the past few years has been that we stand zero chance of changing the algorithms that help to define our lives. Google, Facebook, Amazon—it's all too confusing, too removed from the everyday. As individuals, but also at a societal level—certainly across Germany, maybe even Europe—we seem to go weak at the knees when confronted by the algorithms streaming across the Atlantic. The feeling of losing control owes in part to the fact that around the world, Google and company move in wherever they find the laws and regulations most congenial. Yet as data scientists, we need you backstage with us, as employees, consumers, and citizens. This book assembles a toolkit that will get you on the job, and which I'd now like to introduce briefly.

THE TOOLS IN YOUR DECISION-MAKING KIT

The instruments described in detail over the following chapters will enable you to recognize three things: (a) whether you actually have to get involved; (b) if so, where you can intervene; and (c) the impact your perspective will have on the regulated use of machines. It isn't always necessary to get involved. To help you decide, I present the first tool in your kit, the *algoscope*, which helps us filter out which systems should be our primary focus of concern.

Are all systems that use AI suspicious per se? A great deal of thought has been given to questions like this in recent years. In their 2013 book *Big Data: A Revolution that will Transform how We Live, Work and Think*, Viktor Mayer-Schönberger and Kenneth Cukier propose a sort of overarching algorithmic safety administration that considers each and every algorithm coming to market.[4] As I show later on, this is neither sensible nor necessary in that particular form for a number of reasons. Mainly, however, it isn't necessary because not every ADM system needs to be brought before the witness stand. By and large, *the only systems* that call for regulation and for their internal mechanisms to be monitored are those making decisions about the following:

- People
- Resources that concern people
- Issues that affect people's ability to participate in society[5]

A small portion of all possible algorithms, in other words. The algoscope lets us focus on the ADM systems that carry ethical implications. Parts II and III of this book explain in detail why it is essentially only these systems that require tighter control and regulation.

What does this look like concretely? Systems that decide whether or not a screw is defective and should be taken off the production line do not fall into this category, nor for that matter does a system that distributes fertilizer over a field with pinpoint accuracy. A self-driving car that could potentially get into an accident, on the other hand, definitely makes the

Figure 2
The *algoscope* helps us describe which kind of software we have to keep a closer eye on: algorithmic decision-making systems that directly or indirectly affect humans.

list. Systems that only recognize images or translate languages tend not to belong—unless of course they are built into self-driving cars, where they may lead to an accident. AI systems in the realm of medicine definitely belong, although it is less systems that recommend over-the-counter products than those that make decisions about treatment.

When your AI alarm bell goes off, then, first consider what the system is supposed to be deciding. If it neither directly nor indirectly impacts human well-being, you can return to the breakroom.

In cases where people's well-being is involved, the quality of a machine's decisions will depend on the following factors:

- The quality and quantity of data the machine has been fed
- The underlying assumptions about the nature of the issue at hand
- What society considers a "good" decision to be in the first place

A computer scientist might talk about this last point in terms of a *model for a good decision*; a philosopher would call it a kind of *morality*—that is, a set of standards or principles that any "good" decision should obey. For an algorithm to adhere to such a morality, however, the extent to which a given

decision does so must be made measurable to it; only then can a computer attempt to make "the best" decisions. And this is no simple matter. Suppose software is used to assign children to schools so as to make their way to school as short as possible. Does that mean the trek should be short on average? Or instead that a specific maximum length should be set for every child? Deciding how to later assess the quality of an algorithmic decision allows us to measure how good of a solution it really is. In computer science, this process is called *operationalization*.

To continue with the example of school assignment, there's also the question of precisely what kind of information the computer is being fed to calculate the distance from school. Are ideal travel times or actual travel times being taken as a basis? Has the walk to the bus stop been figured in? These types of decisions create the *model of the problem* that the computer is supposed to solve.

For the results of data processing to observe the moral principles we've established ahead of time, operationalization (O), the model of the problem (M), and the algorithm (A) must all work in concert. Together, they make up the second tool in your toolkit: the *OMA principle*. Beginning in chapter 2, we'll look at a number of examples that show what exactly this principle involves and how to go about using it.

Yet the OMA principle isn't sufficient on its own to determine whether and when machines should play a part in human decision-making. To do that, it's also necessary to consider their role in the overall process.

Figure 3 illustrates the long and winding process of developing and implementing algorithmic decision-making systems. It's a process I call the *long chain of responsibility*, one I explain step by step over the course of this book.[6] Ultimately, its length is problematic because it lets responsibility for individual decisions rest with so many people that later on it becomes difficult to hold any one person accountable. From the outset, though, it's important to recognize that *there are only a few points* along the chain where some form of technical know-how is necessary. By contrast, every step along the way includes aspects on which you can chime in. The long chain of responsibility weaves in and out of the topics raised in this book like a common thread, and with it you now have a third tool in your kit that shows where in the process you have to look.

Just how carefully we have to monitor a machine depends by and large on how much damage the decisions it is calculating can cause and how well we can shield ourselves from that damage. To this end, I present you with

The long chain of responsibility. Only two links in the chain require some degree of technical knowledge, and you both can and should get involved at every step along the way. The following chapters discuss each individual step and what can go wrong in greater detail. A gear icon next to a box indicates some technical knowledge is necessary for the decisions involved at this step, while the icon with two people indicates that common sense is enough and/or social discourse is needed.

a fourth tool linked to a variety of control measures: a risk matrix that indicates how much regulation a given AI system might require. I'll explain this tool in greater depth with a couple of examples once we've completed the backstage tour.

With these four tools, your kit is complete. Once you've become a bit better acquainted with them, you'll be able to determine for yourself when they're called for.

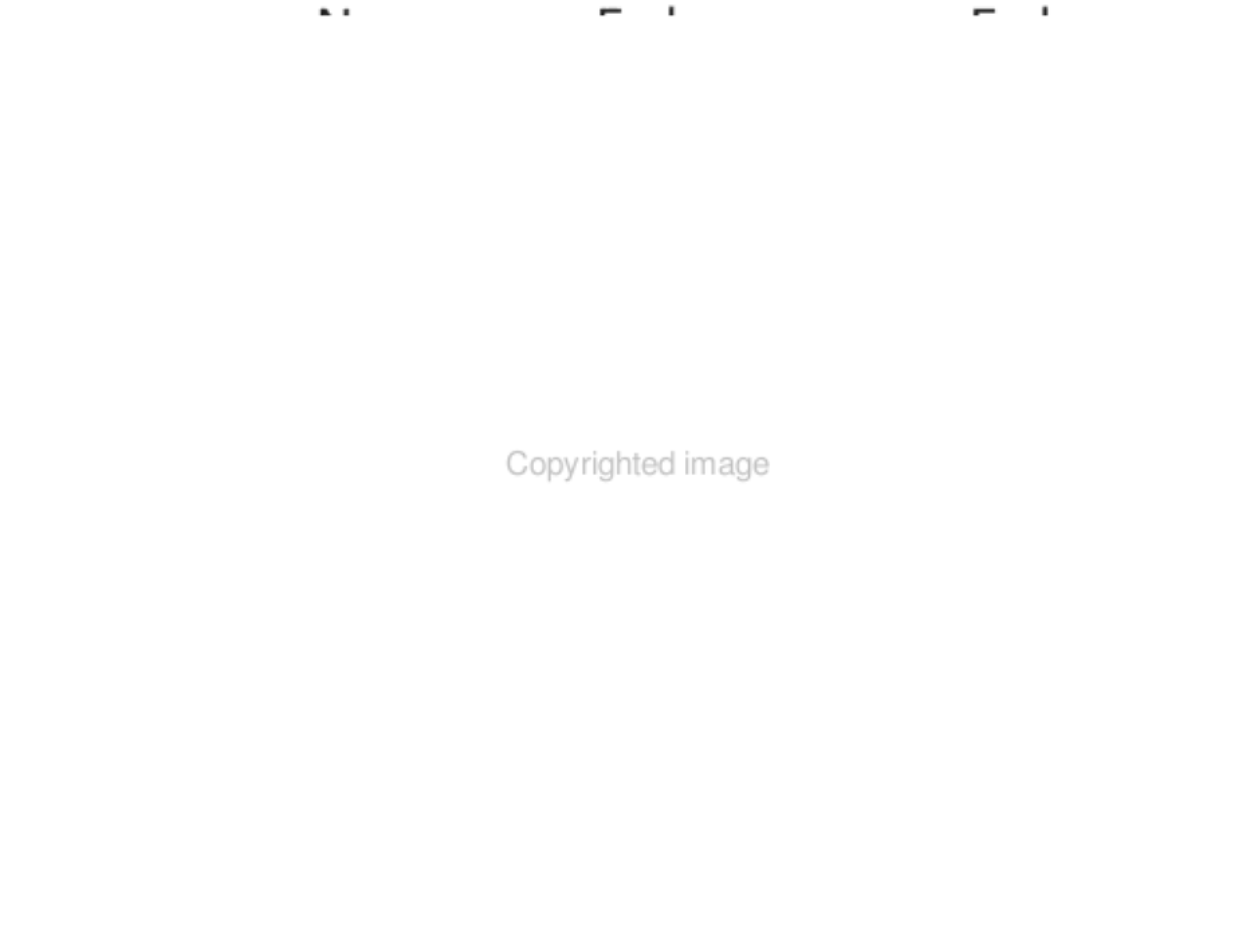
Before we go backstage with AI, though, I want to make a quick detour to the basement, through the laboratories of the natural sciences. Why? Because the goal of artificial intelligence is to reproduce cognitive ability: in particular, the ability to draw conclusions about the world by observing it, that is, making discoveries based on data. And that, of course, is the grand domain of the natural sciences, something they've been doing for centuries now—and with great success.

In one sense, computers do this in a manner very similar to people; in another, they differ radically. To better explain, I invite you to travel back with me to the first time I was involved in a scientific discovery.

situations showed at least one hundred or at least three hundred colonies. The results were astonishing. While both fed cultures did better than the unfed culture, as expected, those fed the concentrate of dying colonies were in fact much healthier than those fed young colonies. Up to eight times as many of the cells fed the dying colonies survived relative to the control group, compared with only three times as many for the cells fed the concentrate of growing cells.

Now that we had the data in front of us, we had to decide whether we could conclude directly on the basis of this observed difference that the concentrate of apoptotic cells had helped. It would have been ideal, of course, if nothing at all had grown on the plates with cells that weren't fed, only a small number of colonies had grown on the plates fed the young cells, and a great deal had grown on the plates fed the concentrate of old cells. Unfortunately, things in life are rarely so clear-cut that you can tell their differences apart immediately and beyond all shadow of a doubt.

Over the course of the experiment, I had set up multiple plates for both cultures that were fed a concentrate. Most of the plates fed the young cells showed fewer than a thousand colonies, while most of the colonies fed the dying cells showed many more. A number of plates in both groups, however, showed around a thousand colonies. Their distributions overlapped,



in other words; it was the *average* number of colonies fed the apoptotic cells that was significantly higher than the average for those fed young cells, roughly as shown in the illustration. Was the difference between the two averages large enough, though? Was it "statistically significant"?

The methods used to answer this type of question work in the opposite direction, by asking first whether it couldn't just as easily be coincidence that the plates fed apoptotic yeast cells did so much better. When seen from a purely statistical standpoint the number of viable cells in two different samples is bound to differ, after all, even if they are drawn from the exact same yeast culture. It's the same as if you roll a die a hundred times, then another hundred times. There is a large probability you will have rolled a different number of sixes in each round. Fortunately, statisticians know how large that sort of a difference normally is; when rolling dice, it is much more likely to be small rather than large.

It was the same with our yeast cells. If a statistician were to take a sample from two different yeast cultures, she would compare the *observed* difference in the number of viable cells to the difference that could be *expected* if the two samples were taken from the *same* culture. If the observed difference is comparable to the expected difference—that is, similar in size to what one could expect of two random samples taken from the same culture—the statistician would label that difference *insignificant*. In our case, the greater the variation between the two, the more strongly our results would support the hypothesis that one culture did in fact contain more viable cells than another.

Sadly, my degree in biochemistry didn't include a single course in scientific theory, nor for that matter any introduction on the right way to evaluate biochemical data statistically—and really, why bother?![2] Nor for that matter did all of us budding biochemists find mathematics exactly riveting. I myself had always liked math, but without any background in statistics I also lacked the knowledge necessary to demonstrate with any kind of scientific rigor that yeast cultures fed apoptotic cells did in fact show a statistically significant advantage over those fed younger cells.

I dove into the literature and soon found myself awash in a sea of statistics books. Yet nowhere did I find a patch of solid ground that would let me decide for certain which statistical method was the right one. How was I supposed to distinguish between methods for "normal distribution" and other types, for example? In the end, I opted for one of the simplest.

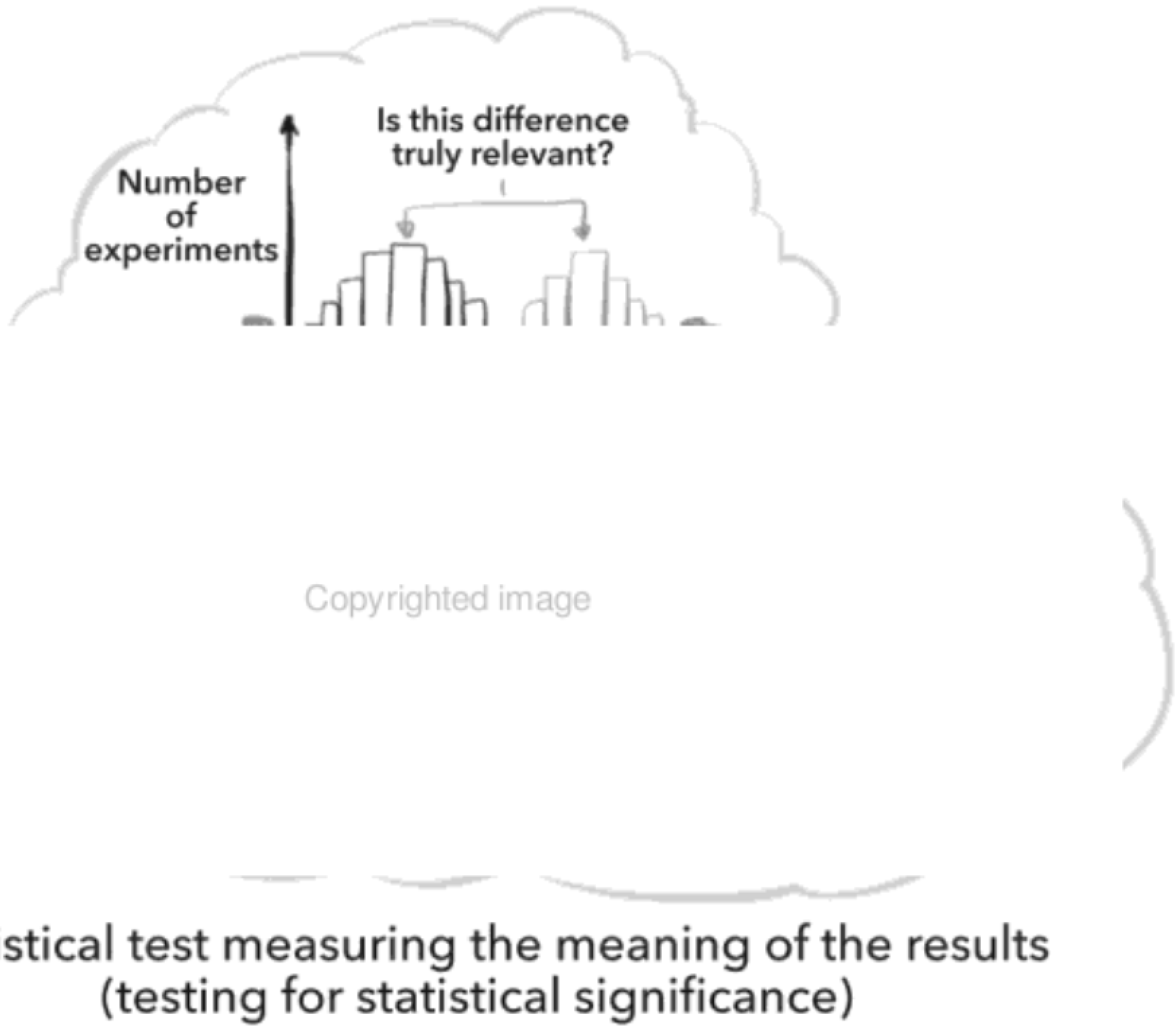


Figure 4
A test for statistical significance measures whether two observed distributions—in this case, the number of viable cells in two yeast cultures—show a conspicuous difference or not.

Armed with this knowledge, I became a one-eyed queen in the land of the blind. My team subsequently used the same method on any issue that didn't resolve itself quickly enough—always with our fingers crossed that we were actually doing it correctly.

As for my diploma thesis, it turned out that cells fed the concentrate of apoptotic cells did in fact stand a significantly greater chance of surviving. A "greater chance" is not the same as a guaranteed outcome, however, let alone sufficient evidence to conclude a direct causal relationship. It is merely a *correlation*—that is, two properties or patterns of behavior that are often observed to coincide with one another. What our observations did do was help support the hypothesis that a causal relationship *might* exist.

And so, after nine months of work, all I had managed to do was fit one small piece into a gigantic puzzle.

This explains in part why we've taken a quick detour through the laboratory on our way backstage. It's because the algorithms I discuss in this book would stop right here and simply accept the findings as such, rather than continuing on to test the correlations directly for causality. Rather, if algorithms find two things appearing alongside one another often enough it is made into a rule: "If you see the one thing, expect the other!" In this case the rule would be this: "Cultures fed by older cells will always show greater rates of survival."

Fortunately, biologists can shore up confidence in their results by running numerous similar experiments or drawing on other means of analysis and experimentation. That was exactly what my thesis advisor Frank Madeo did with the many students who came after me, and today we can rest assured that unicellular yeast cells "have good cause for apoptosis," as Frank and his coauthors phrased it.[3] As for yours truly, it was the last time I would be caught in a laboratory. I was drawn to computer science instead.

FROM DATA PRODUCER TO DATA ANALYST

To this day, the joy I find in searching for the best ways to evaluate data has never left. Yet neither has the question of when and where a given method can in fact be used to meaningfully interpret results. This sort of critical awareness of methodology is called *literacy*, a term that encompasses a great deal besides: knowing the facts, but also selecting them discerningly with an eye toward solving a problem, as well as the ability to solve problems in and of itself.[4] These also happen to be the very skills needed in the field of artificial intelligence, where sometimes it is about as clear as mud which particular method will bring forth the best conclusions from the data.

As my work in biochemistry drew to a close, it was my time spent toiling in the laboratory I was happiest to leave behind. Generating data was *laborious* in every sense of the word, while the part that actually brought me joy—analyzing data—always seemed to get the short end of the stick. I found it maddening just how many individual experiments and observations it took to piece together a single causal chain. By *causal chain*, I mean setting facts in a sequence that explains how a given observation came to be. And that is exactly what machine learning promises us today: that *correlating* data with observed behavior is on its own enough to make decisions about new data.

To do so, however, would be to jump to conclusions. Tyler Vigen captured this in a particularly memorable way on his website, Spurious

Correlations (and in the eponymous book).[5] A visitor to the website encounters various sets of public government data, from which she then picks a set of her choosing—the number of divorces in Alabama, say—to see how the numbers have shifted over the years. Once a set of data has been selected, all other available data sets are sorted by their correlation with the chosen set. If two sets behave in the same way—that is, both sets of values rise and fall at the same time—they are said to be *highly correlated*. The exact degree of correlation can be measured using mathematical formulas. And lo and behold, the share of women with a degree in engineering shows a strong correlation with the divorce rate in Alabama![6] Figure 5 shows the change in divorce rate over time compared to the percentage of female engineers. They rise and fall nearly at the same time; in this case, there is a visible correlation. Was this undeniable evidence that women working in male professions destroys marriages? Or alternatively that women who had been left by their husbands were going on to pursue engineering degrees?

The answer in both cases is a resounding *no*. What is going on here is a case of *spurious* correlation, or coincidence that appears statistically.

5
5
5
5
5
4
4
4
4
4
00
50
00
50
00
50
00
50
Copyrighted image
Divorce rate in Alabama (number of divorces per 1,000 couples)
Percentage of engineering degrees held by women

Figure 5
The annual change in the divorce rate in Alabama and the share of engineers who are female. The two curves show a strong correlation, which means they rise and fall nearly at the same time with only slight deviations.

nondeterministic infinite automatons into deterministic finite automatons, or something equally abstract. I sat there, spellbound—though I couldn't say the same for many of my friends. Inspired by the great mathematician and computer scientist Alan Turing, theoretical computer science asks the philosophical questions raised by the discipline. What does computability actually mean? Do problems exist whose solutions only computers can compute or questions only humans can answer? Are there questions that neither people nor machines can solve by a general schema?

As it turns out, the first generation of computer scientists managed to come up with quite a surprising answer in response to these downright ethereal questions: *Based on everything we know to date, humans and computers are essentially capable of answering the exact same questions. Both are capable of solving—and fail to solve—the same problems.*

Such is the gist of the Church-Turing thesis.[8] Humans and machines can both calculate the root of one million or the shortest path from A to B, for example, or arrange a pile of books by the last and first names of their authors. Neither, by contrast, is capable of coming up with a general method for determining whether a given piece of software code will ever enter an infinite loop. That's a shame, by the way, for a tremendous number of computer crashes could be avoided if such a method did exist. In this case, however, we've run up against the limits of computability.

Something about Professor Lange's lecture set me thinking. *People actually get paid to figure out philosophical and mathematical puzzles like these? That's what I want to do!* I found the work that went into designing algorithms especially appealing. Discovering, then analyzing and evaluating patterns in data was the piece of the puzzle that joined my various passions: my love of the natural sciences, but also my curiosity about what certain observations might herald for our lives and societies.

But does the Church-Turing thesis actually hold water? Don't we all share the sneaking suspicion that computers are much better at calculating than us humans, we who are constantly making mistakes? Only rarely do people get the same answer twice when asked to calculate even a small handful of numbers. We make subjective, not objective, decisions and often fail to see the forest for the trees. Fortunately, making calculations is child's play for a computer; adding together long strings of numbers, generating statistics, or searching for patterns within large sets of data—none of these pose a problem. Computers don't slip up; when given the same input, they will always give the same result. That's because the way a computer calculates the desired result has been prescribed by an algorithm that sets out in great detail how the computer should arrive there based on the input (more on that in chapter 3). No hormone fluctuations, no bad days, no surprise prejudices—they are *lifeless decisions*, in the best sense of the term.

Yet it is the very same spark of life that computers seem to be lacking when it comes to our deepest emotions and judgements as humans. Say you wanted to commission a poem or a piece of art: it's difficult to imagine a computer fabricating something that another person would enjoy. The same holds true for questions of justice and fairness in court, for example, or when educating our children or caring for the sick and elderly. Isn't a computer bound to fail in these instances if it "lacks soul"?

But these days, an entirely new breed of algorithm has emerged that would seem to overtake us in these matters as well. I'm talking about algorithms that make use of *machine learning* and that form the basis of artificial intelligence. With their help, texts that have foiled other methods for years are now suddenly translatable; the famed Babel fish from Douglas Adams's *The Hitchhiker's Guide to the Galaxy* seemingly draws nigh. Machine learning is capable of identifying the most important objects in a photo and transcribing spoken language more quickly and reliably than humans are able to. AI has even composed poems and painted pictures that humans regard aesthetically.

So why dally in the natural sciences when artificial intelligence seems like such a safe bet? It is because machine learning—an essential component of artificial intelligence—turns fact-finding completely on its head, a process that may have advanced very slowly over the centuries, but with great success. Instead of searching for reasons (a causal chain), machine learning identifies modes of behavior or properties that often appear alongside (correlate with) a significant event: the age of a driver in the case of an accident, for example, or personal characteristics often associated with criminal recidivism. Yet in contrast to classical algorithms, where a model is constructed (i.e., a mathematical problem is defined) *before* the algorithm is designed and used, it is now the algorithm itself that *constructs a model of the world from the data*; more on that later too.

The automatically discovered correlations I discuss in the following chapters are rarely reviewed and never examined for causal connections, yet they are still used to stick people in different risk categories. As we will see, this regularly leads to mistakes. To my mind, that means we can only take meaningful advantage of the efficiency gains machine learning offers if we examine these correlations for causal connections, as is normally done in the natural sciences.

With these initial considerations in mind, we are now ready to step backstage. As the first step in the long chain of responsibility, I pick up with the ABCs of computer science: algorithms, big data, and computer intelligence.

II

THE ABCs OF COMPUTER SCIENCE

Flip through the pages of just about any newspaper these days and there's a good chance you will come across at least one article featuring the terms *algorithm*, *big data*, or *artificial intelligence*. One reads in the *Guardian* of "Franken-algorithms," while the *New York Times* warns us that people may wind up "wrongfully accused by an algorithm." We'll get to that, but before we do, what exactly is an algorithm? How is it related to digitalization, and what does it have to do with big data? Part II explores these questions through the ABCs of computer science, beginning with A, of course, which in this case stands for *algorithm*.

solves what's known as the shortest path problem using an algorithm that finds the shortest route from A to B.

An algorithm, then, is a detailed set of instructions for how to actually arrive at the desired solution once all the information needed to do so is available. It's essentially what you might explain to any rookie the first time she has to solve a common problem in her profession independently. In order truly to count as an algorithm, however, those instructions must be rigorous enough that they can be translated into programming languages. In computer science, this step is called *implementation*.

Math puzzles often bear a resemblance to the sort of mathematical problems I am talking about. I'll give you an example: Which four numbers add up to 45 and are also all the same number if you add 2 to the first, subtract 2 from the second, divide the third by 2, and multiply the fourth by 2?

In this case, the input is 45, and the solution—the four numbers—must meet certain requirements to count as solutions.[2]

And how would you go about solving the problem? Well, we can start by whittling down the list of candidates with a couple of initial considerations, and then proceed by trial and error. The difference between the first two numbers must be 4, as adding 2 to the first gives the same number as if you subtracted 2 from the second. It follows from this that both numbers are either even or odd. As for the third and fourth numbers, the fourth number equals one-half of the third number when doubled, which means it must be one-quarter as large as the third number. Any number multiplied by 4 comes out even, so for all four numbers to add up to an odd number (45), the fourth number must itself be odd. That means it could be 1, 3, 5, 7, and so on; once we've figured out which, the other numbers will follow. If the fourth number is 1, that makes the third number 4, and the first and second 0 and 4, respectively. Yet that would make for a sum total of 9. Proceeding by this kind of *guesswork*, we eventually arrive at 5 for the fourth number, making the third number 20, the first 8, and the second 12, for a grand total of 45.

Now, this kind of guesswork isn't an algorithm but what is called a *heuristic*, a term which comes from the Ancient Greek *heurískein*, "to find" or "to discover." Heuristics are strategies developed for finding solutions to a problem that have proven themselves so far, but offer no guarantee of ultimately finding a solution that satisfies all the set conditions.

One interesting example of a heuristic comes from ants in their own attempt to solve the shortest path problem. When off in search of food, an ant initially wanders about quite at random, leaving a scented trail behind it. Should it happen upon something delicious, the ant then uses the same trail to find its way back to the nest. On the way back, the ant lays out another scented trail—the more delicious the food and the more of it there is, the stronger the scent. This in turn allows the food to be found by other ants, who then further strengthen the trail. Since the scent is evenly distributed equally in all directions, it is more concentrated within a curve than directly at its edge. That's because all of the little scent molecules radiate out and meet somewhere inside the curve. It's similar to perfume; if you spray some on your neck, chest and wrist, the greatest concentration will be found somewhere between those sources. This means that other ants near the center of the curve will be somewhat more attracted than those at the edge, so that the loops initially traced by the first ant become shorter and shorter. Ultimately this leads to a relatively short path, albeit not necessarily the shortest.

We will come across the concept of a heuristic again in chapter 5 on computer intelligence, as most of the methods used in those cases aren't algorithms at all. How about that for some specialized knowledge to boost your score on trivia night! Still, it's an important point to remember: Only algorithms are certain to find the best solution; heuristics can't make the same promise.

Nor is it worth taking the trouble to develop an algorithm that applies generally anyway, so long as we are dealing with individual cases like the numbers puzzle given previously. Rather, it is regularly recurring mathematical problems *of a general nature* that deserve attention. Examples might include questions about the root of any given number or the product of any given set of numbers—but also consulting a database for all the purchases a customer made last year.

We've now assembled all the pieces we need to explain the term *algorithm*:

> An algorithm for a specific mathematical problem is any description of instructions sufficiently detailed and systematic such that if transferred into code correctly, its implementation will calculate the correct output for any correct input.

The computer scientist in me would like to add that the algorithm must calculate its solution in finite time, thank you very much. It seems to defeat

Figure 6
An *algorithm* has a plan for finding the solution and guarantees that it is actually a solution. A *heuristic* is a method that attempts to find a solution.

the purpose, after all, if we have to wait until the end of the universe to get the answer! But enough small talk. Would you like to see one for yourself, a real live algorithm? Step right up then, folks, as I present to you: the sorting algorithm!

THE ALL-PRESENT SORTING PROBLEM

Some of my favorite childhood memories are the afternoons spent at my father's side, helping him sort through his collection of advertising stamps. From 1880 to 1940, these stamps were used similarly to today's trading cards, as a reminder to customers of whatever product they had just brought home. The stamps had no postal value but were often used to decorate letters or simply assembled in large albums; it was these albums that my father collected.

Unfortunately, the stamps' previous owners had often glued them into the albums instead of simply laying them down on the pages. This meant that before we could get down to resorting them, we usually had to detach the stamps from the albums by rinsing them with soap suds in a little tub then drying them off with blotting paper. The sorting itself was more of an interactive process. In the case of the stamp shown in figure 9, the conversation might have gone something like this: "Papa, how do I sort this stamp? By the product, Sachsenglanz, or the company, W. Stephan? And if it's by the company, should I use the W, or the S for Stephan?"

Our sorting rules were constantly being refined, as each stamp looked different and I had no idea which keyword my father might use the next time he looked one up. So I sat there and sorted through hundreds upon hundreds of stamps, initially by the first letter, and then within that pile by the second letter, and so on. Finally, there was a small enough number of stamps left over that I could simply arrange them as you might a hand of cards and sort them in the correct order.

Figure 9
An advertising stamp from the early twentieth century.