Algorithms and Everyday Life

Ernest Davis
Department of Computer Science
New York University
`davise@cs.nyu.edu`

June 27, 2016

Review of *Algorithms to Live By: The Computer Science of Human Decisions* by Brian Christian and Tom Griffiths (Henry Holt, 2016).

Algorithms to Live By by Brian Christian and Tom Griffiths is a book written for a general audience about a number of ideas in computer science and how they are applied in practice. Despite the widespread interest in all things related to computers, very few such books have been written, and I know of no others that cover the particular subjects included here. The book gives clear, almost entirely math-free, accounts of the ideas; narrates their historical development, enlivened with many personal anecdotes;[1] and describes a variety of applications.

The applications fall into three categories. First, there are applications within computer technology itself; unsurprisingly, these form the majority of the applications discussed in the book. Second, there are applications in commercial or institutional settings; for instance, there is a fascinating discussion of techniques for testing new medical treatments which balance the need for control groups in reliable experiments against the problems involved in denying control subjects the new treatments that they need. Finally, there are applications that are presented as solutions to everyday problems that arise for the man on the street. These last seem to me the least successful aspect of the book. A few seem interesting, such as deliberately introducing random elements into projects you are working on. Most of them I find unconvincing, and a couple, which I will discuss below, seem to me completely crazy.

Finally, the authors argue that there is reason to believe that people in any case naturally use these kinds of ideas in their ordinary thinking and action. Some of this seems to me reasonable; much of it seems to me overstated. I will discuss this in section 4.

The authors' view of their own work is very different. As the book title indicates, they consider that the central point of the book is precisely the application of computer science to everyday problems. Moreover, Christian and Griffiths claim optimistically that "looking through the lens of computer science can teach us about the nature of the human mind, the meaning of rationality, and the oldest question of all: how to live." I agree that the ideas are somewhat relevant to the meaning of rationality and to the question of how to act sensibly in various situations ("how to live" is a rhetorical exaggeration). However, I am skeptical that the ideas are very helpful in understanding

---

[1] Actually, the personal anecdotes get a little wearisome. After reading about how Michael Trick used an algorithm to choose a wife, and how Laura Albert McLay organizes breakfast for her children, and how Jan Karel Lenstra helped Gene Lawler move house, and how Meghan Bellows optimized the seating plan at her wedding, and how Tom Griffiths waited a long time in line to pay for crêpes at the Cinco de Mayo festival, I started to have a craving for impersonal technical papers written in the passive voice.

the nature of the human mind; the problems that they address are a small and rather eccentric selection of the tasks faced by human cognition.

Overall the book is in fact very impressive, and well worth reading. It is extremely well-written and quite entertaining. The lay reader will have to work pretty hard in places, but he will be well rewarded for his work. Or he can skip to the next easy patch; much of the book is smooth sailing. Behind the scenes, the authors have done a remarkable amount of solid research; they carried out about a hundred interviews and their bibliography has about 1000 items, including quite a bit of pretty obscure technical work. For the reader with an appetite for more, 100 pages of endnotes give lots of mathematical details and pointers to the literature. Most of the historical content and a fair amount of the technical content of the book was new to me, and I have been teaching a graduate course on algorithms for thirty years.

Turning to specifics: The chapters on relaxation techniques for solving optimization problems, on randomized algorithms, on networking (mostly TCP and buffering issues), and on game theory seemed to me excellent, and the chapters on explore/exploit, sorting, scheduling, Bayes' law and networking seemed to me fine. However, I think that the chapters on optimal stopping, on caching, and on overfitting have some serious flaws that are worth discussing, and the explanation of these is unavoidably a little lengthy. Furthermore, I disagree substantially with their general message about human cognition; and I want to say a few words about that. So most of this review will end up being negative in tone; but as I've said, overall much of the book is really excellent.

1 The Optimal Stopping Problem

The optimal stopping problem, sometimes known as the "Secretary Problem," became well known through one of Martin Gardner's "Mathematical Games" columns in *Scientific American*. It runs as follows: Suppose that you are being shown, one by one, a sequence of N of items of varying degrees of quality. At any point you can stop and choose the one you are currently being offered; however, if you decide against an item, then it is forever lost; you are not allowed to go back. The problem gets its name from the case of an employer who wants to hire a secretary and is interviewing applicants one by one, under the constraint that a job offer can only be made to the applicant currently under consideration, and that once an applicant has been rejected, they are no longer available.

Two further assumptions are made. First, it is assumed that the information gained is purely ordinal; the employer can judge whether this secretary is better or worse than the other candidates he has seen, but has no other measure or judgement of quality. Second, it is assumed that the employer's goal is to maximize the probability of hiring the absolute best secretary, and he is indifferent to all other outcomes; if he can't have the best, then he doesn't care whether he gets the second-best or the worst.

Given this problem, the fairly well-known solution is that the employer should pass on the first N/e candidates and then hire the first candidate who is better than all the ones he has seen so far. It turns out that, following that rule, he has a $1/e$ probability of hiring the best of all the applicants.

Even as a model of hiring a secretary, this is pretty far-fetched. Christian and Griffiths, however, propose to apply it to a even more unsuitable problem: dating, with the goal of matrimony.

I actually can't think of any case I have seen where a mathematical model was a worse fit to a real-life situation. There are lots and lots of problems with this, but three seem to me critical.

Let me illustrate the first two problems with two hypothetical examples. First example: Suppose that our hero, Algernon, expects to have the opportunity to meet up to $N = 100$ women over his amorous career. His plan therefore is to date and dump the first 37 of these, and then propose to the first woman who is better than those 37. But the course of following algorithms never did run smooth. Date D_{10} is all that he had ever dreamed of and more, and she seems to like him. Shall he listen to his heart, or should he listen to the stern voice of Mathematics supported by Proof, which tells him that there is a 90% chance that the best woman for him of the 100 candidates is one of the 90 he hasn't met yet?

Second example: Suppose, on the contrary, that Algernon really liked dates D_5 and D_{17} but, obedient to the algorithm, he dropped them, and he has never met anyone like them in the next 80. Date D_{97}, however, is almost their equal, but not quite; she is not as witty as D_5 and her voice is less melodious than D_{17} but she'd be quite easy to love, quite easy to idolize all others above, except D_5 and D_{17}. Should he make the best of a very good deal and propose; or shall he wait to see whether D_{98}, D_{99} or D_{100} are really the absolute top?

Is Algernon being hopelessly irrational if he ignores the precepts of the optimal stopping algorithm in either of these cases? Certainly not; the assumptions of the model are not valid or close to valid. In the first example, the ordinal information assumption is violated; Algernon has presumably more or less good reason to think that D_{10} is in fact an outlier and therefore good reason to think that the probability of meeting someone even better is small rather than 90%. (This can be perfectly reasonable even if Algernon cannot or does not wish to express his feelings for D_{10} as a numerical value.) And in both examples, the "best or don't care" assumption is violated.

In fact, the "best or don't care" assumption runs strongly counter to the spirit of the maximum expected utility principle, though I suppose not to its letter. The implicit utility function here is 1 if the absolute best candidate is chosen and 0 otherwise. Mathematically, that is not an impossible function; but cognitively it is a very strange one, since the chooser almost never knows whether he has chosen the absolute best candidate, and thus never knows the utility for himself of his own outcome state. Since Griffiths is strongly committed to Bayesian theories of cognition and rationalist cognitive theories of choice, it seems very peculiar that he is willing to take seriously a model of this kind.

In fact the above two flaws apply to practically all actual cases of sequential choosing, whether you are dating, hiring a secretary, or wine-tasting. It is actually hard to think of any natural case where the constraints of the optimal stopping problem do apply or come close to applying.

A third flaw applies specifically to dating. A date involves two players, both of whom are making a choice. Suppose that Algernon and Cecily are a match made in heaven; each would prefer the other to anyone else. Suppose further that they are both following the algorithm above, The match will fail to come off, if either the date between Algernon and Cecily occurs in his first 37%, or in her first 37%, or if either accepts someone else first, and therefore they never date at all. The likelihood of that last possibility depends in turn on how well Algernon's rivals like Cecily, and how well she likes them, and where *they* are in their courtship career when they date her. Thus in fact this is an N person game, where N is the number of people currently dating, and where players constantly join and leave the game. All kinds of parameters affect the expected outcome of the game, and the choice of best strategy, including the correlation of preferences between potential partners, the correlation of preferences across rivals, and the correlation of date schedules among the players.

Ordinarily, if I ran across this application in a discussion of this algorithm in a book of popular math, or in Martin Gardner's column, I would say that this is just a cute, vivid example for the sake of illustrating the algorithm; and that there is no point in taking it seriously or finding flaws. But the whole point of Christian and Griffiths' book is that algorithms are useful for actual real-world problems; and this algorithm is chapter 1; and this is the flagship application for that chapter. Moreover they tell an anecdote of how Michael Trick of CMU actually used the algorithm, and ended up married to a lovely woman he met in Germany; so, they argue, it really does work for smart people, it's not just theoretical. The figure "37%", which is only valid if the assumptions hold to the letter, is mentioned seventeen times in this chapter. The book does consider variants of the problem that do not make the ordinality assumption or the "best or nothing" assumption; and the fact that there are two people involved is mentioned in an endnote; but those are distinctly second thoughts. (The book also discusses at some length the fact that, in real life, you can sometimes go back and date someone again, and how the algorithm can be modified to accommodate that.)

2 Sorting

Chapter 4, on caching, is mostly fine, but unfortunately at one point it becomes seriously misleading.

The economist Yukio Noguchi developed a personal filing system in which files were kept sequentially and then, whenever a file was returned to the cabinet, it was placed at the front. When he needed a file, he did a linear search starting from the front. Thus the files were always in order of recency of access.

Christian and Griffiths argue that Noguchi's system should be used for any kind of stuff one needs to access, contrary to the uptight, oldthink idea promoted by Martha Stewart and the like that it is worthwhile organizing things systematically. Moreover, they claim, the superiority and indeed optimality of Noguchi's system have been mathematically proved. They write:

> [T]he big pile of papers on your desk, far from being a guilt-inducing feature of chaos, is actually one of the most well-designed and efficient structures available. What might appear to others to be an unorganized mess is, in fact, a self-organizing mess. Tossing things back on top of the pile is the very best you can do, shy of knowing the future.

This seems completely impossible, and in fact it is hugely overstated. The truth is that if you have a storage device that displays all of its contents equally (a bookshelf, a file drawer, a closet, a listing of files in a computer folder etc.) then almost any organizational principle almost always beats out listing things by recency of access and doing a linear search.[2] If, for instance, there is some verbal description that you can use, such as a name, then organizing things in alphabetical order gives you a searching time of $O(\log(N))$ using binary search or $O(\log(\log(N))$ using interpolation search.

Christian and Griffiths' claim is based on two different mathematical analyses; the first is irrelevant, and they have exaggerated the scope of the second. The first theorem is due to Daniel Sleator and Robert Tarjan (1985); it states that *assuming that you always search in linear order from the front,* then Noguchi's system never requires more than twice as much time as you would need if you could foresee which files you will need in the future; and, moreover, that no algorithm that is not clairvoyant can reliably do better than that. So in a well-defined sense, it is the optimal realizable algorithm, *if you always search in linear order from the front.* The result applies if you will recognize what you are looking for when you see it, but otherwise have no systematic way of characterizing it; or if your storage system physically prevents you from searching in any way other than linear order. But it says nothing about the case where you have open access and some principle that you can use for both organizing and searching.

The second argument is based on typical access patterns and is somewhat more cogent. In many situations, access patterns follow a recency rule governed by an inverse power law distribution: If you have accessed an item at time T then the probability that you will need it again at time $T+K$ is proportional to $K^{-\alpha}$, with α usually somewhere around 2. Therefore it makes sense to keep recently

---

[2] There are exceptions, depending on the expected pattern of access. For instance , paid bills have to be saved, but in my experience are almost never accessed, except to throw out the ones that are from years ago. In that case, you may as well keep them organized by year, but with no finer organization.

accessed items somewhere easily available; that is the reason that LRU (least recently used) is the standard technique in caching and paging systems for choosing an item to delete from the cache. On that principle, you should keep recently accessed items in a particularly accessible spot, regardless of your overall system. If you are currently reading a book, by all means keep it at the left end of your bookshelf, rather than in its proper alphabetical position.[3] But that's only worthwhile for a very small number of books. Once you have more than 3 or 4 books in this stack, it is absolutely worthwhile to take the time to take the last of these and put it back where it belongs; and there is no math that says otherwise.

3 Overfitting

Chapter 7, entitled "Overfitting: When to Think Less" is very muddled in several respects.

Overfitting is the bane of machine learning (ML) algorithms that learn some method of predicting answers to some kind of problem. When overfitting occurs, the method that is learned is very good at calculating the answers for the data that has been trained on, but much less good for new problems that it encounters. In a sense, the ML algorithm has become fixated on unimportant features of the training data, and is overlooking the big picture. It tends to occur when the ML algorithm has too many options to play with in designing its question-answering methods; it then has the freedom to twiddle with knobs and add gadgets until it has exactly matched the training data rather than looking for large, systematic patterns that may correspond to the real features of the data, though not every detail.

Thus, overfitting is essentially a category of bug in finding explanations for data. There are various techniques for fixing overfitting; some of these are fairly general, others are specific to a particular ML algorithm. Some of these techniques involve, broadly speaking, thinking less as the chapter title advocates: for example, reducing the number of features that the ML algorithm can consider or reducing the number of iterations that it runs. Other techniques involve thinking more, such as the post-pruning algorithm for decision tree construction. Some ML algorithms are highly susceptible to overfitting, some are much more robust; it depends in part on the characteristics of the data.

As far as I can see, overfitting has few close real-life analogues. The best I can think of is certain kinds of conspiracy theorizing, in which all kinds of random or irrelevant considerations are required to be explained and end up being viewed as evidence for the theory.

Chapter 7 begins with Charles Darwin making a list of the pros and cons of getting married, in order to help himself decide on a course of action, and with Ben Franklin endorsing that technique for making decisions in general. It is explained at the end of the chapter that Darwin ran the risk of overfitting, but avoided it by limiting his list to the length of the page.

However, overfitting has nothing to do with choosing an action; it is purely a feature of fitting an explanation to data and then using that for prediction. Darwin was not fitting data; so the Darwin example is completely irrelevant.

The discussion then moves onto a particular case of prediction. They use as their example some data collected by a sociological study on how life satisfaction changes after marriage; it reaches a peak in the first year, then declines, then more or less reaches stasis. Christian and Griffiths write

---

[3] I do not see eye-to-eye with Christian and Griffiths on the subject of books. They write "It's fairly rare that we find ourselves searching for a particular title" and they estimate that, if they've forgotten where a book is on their shelves, it will take ten seconds to scan for it. My own experience is that I look for specific books fairly frequently, and that if a book has been seriously misplaced, it can easily take twenty minutes or more to find it. My perspective may well be warped, however.

as follows:

> One possible formula would use just a single factor to predict life satisfaction: the time since marriage. This would create a straight line on the chart. Another possibility would be to use two factors, *time* and *time squared*; the resulting line would have a parabolic U-shape letting it capture a more complex relationship between time and happiness. And if we expand the formula to include yet more factors (time cubed and so on) the line will acquire ever more inflection points getting more and more "bendy" and flexible. By the time we get to a nine-factor formula, we can capture very complex relationships indeed.
>
> . . .
>
> The one-factor model, unsurprisingly, misses a lot of the data points though it captures the basic trends – a comedown after the honeymoon bliss. However, its straight-line prediction forecasts that this decrease will continue forever, ultimately resulting in infinite misery. . . . The two-factor model comes closer to fitting the survey data, and its curved shape makes a different long-term prediction, suggesting that after the initial decline satisfaction more or less levels out over time. The nine-factor model . . . predicts misery at the start, a giddily abrupt rise in satisfaction after several months of marriage, a bumpy roller-coaster ride there after, and a sheer drop after year ten.

There are some striking minor mistakes in this paragraph. First, as best as I can measure from the diagram in the book,[4] it looks like after 70 years of marriage, which is as much as one can expect, life satisfaction will be down around 3, not hugely negative. Second, a quadratic function does not "level out over time," except in the short term. As stated correctly in the first paragraph, a quadratic function is U-shaped; it ultimately goes off to infinity much more quickly than any linear function. Again estimating from the graph, though of course with much less precision, it looks like the parabola they draw is close to its minimum at 18 years, so it would predict that after 36 years of marriage the original honeymoon bliss is recovered, and that after 70 years life satisfaction will be up around 9 — not an enormously better prediction than the linear model.

The more important mistake, though, is that using the problems of interpolating n points with an $n-1$ degree polynomial to illustrate overfitting is somewhat like using anecdotes of fatal alcohol poisoning to illustrate the dangers of drinking; the example is certainly dramatic, but they are hardly typical and they don't actually illustrate the real problem. Polynomials are known to be a particularly terrible way to do interpolation and a worse way to do extrapolation, for reasons that are well understood in interpolation theory. You can do exact interpolations over N data points using a linear sum of any N functions that are linearly independent over those points; and there are alternative families of functions that are known to work much better than simple powers of t.

Christian and Griffiths then discuss various kinds of real-world errors as examples of overfitting.

> Overfitting, for instance, explains the irony of our palates. How can it be that the foods that taste best to us are broadly considered to be bad for our health, when the entire function of taste buds, evolutionarily speaking, is to prevent us from eating foods that are bad?
>
> The answer is that taste is our body's proxy metric for health,. Fat, sugar, and salt are important nutrients . . .
>
> But . . . [w]e can now add fat and sugar to foods beyond amounts that are good for us and then eat those foods exclusively.

---

[4] The actual data values are not in Christian and Griffiths, nor could I find them in the paper that they cite; and, as they discuss, the coefficients of the polynomials are very sensitive to the precise values.

However, this is not overfitting. From the standpoint of ML theory, the problem is here is that the test data (the foods we eat now) are systematically different from the training data (the foods our ancestors ate while taste buds were evolving). That is also a very important problem in machine learning theory, but it is a very different problem from overfitting; and techniques that work well for overfitting may be entirely unsuited to this problem.

4 Psychological theories

Griffiths is, professionally, a cognitive psychologist and the book contains a substantial discussion of the analysis of human cognition in algorithmic terms. Some of this material I found quite fascinating, even exciting. For instance they discuss a remarkable experiment by Jackson Tolins and Jean Fox Tree that showed that speakers telling a dramatic story to a listener rely on receiving a steady stream of so-called "back-channels" — "uh huh", "yup", "sure" etc. — and that if they don't get it, they may be unable to tell the story properly.

However, I disagree with a lot of the psychological discussion here. At this time, cognitive psychologists tend to be divided into two camps. The scientists in the first camp argue that most aspects of human cognition can be understood as rational, perhaps even optimal, when set in their proper context. The scientists in the second camp take the more commonsensical view that human cognitive processes are astonishing in many ways, but often suboptimal or irrational judged against mathematically normative standards. In other words, humans unlike Vulcans make lots of quite unnecessary mistakes. Griffiths is a prominent member of the first camp; I myself am largely in the second camp.

Accordingly the book contains many arguments that purport to show that seeming failings in human cognition are actually rational once everything has been taken into account or they are inevitable consequences of following algorithms that are actually provably optimal or at least reasonable.

For example: Christian and Griffiths argue that sometimes having trouble remembering things is inevitable if you have a lot of things to remember, and older people forget more things, not because their memory is failing, but because they know so much.

> We say "brain fart" when we should really say "cache miss." The disproportionate lags in information retrieval are a reminder of just how much we benefit the rest of the time by having what we need at the front of our minds.
>
> So as you age, and begin to experience these sporadic latencies, take heart: the length of a delay is partly an indicate of the extent of your experience. The effort of retrieval is a testament to how much you know.

More examples: If old people are more set in their ways than young people, it is because both groups are optimizing the explore/exploit trade-off as befits the time left to them in the world.[5] Thus babies and toddlers, who are at the extreme end of this trade-off, rationally try out all kinds of stupid things to see if they work.

As discussed earlier, if you are a slob who simply stacks the papers you are working on in big piles rather than sorting them, then Christian and Griffiths congratulate you on optimizing your

---

[5] The explore/exploit trade-off deals with the choice between experimenting with something new versus going back to the tried and true; for example trying out a new restaurant versus going to an old favorite. The optimal strategy depends on how many more opportunities for making that choice you expect to have. If you are a newcomer in a city, you should spend most of your time exploring new restaurants and only occasionally return to an old one; if you will be leaving for good in a few weeks, you do better to stick to the ones you know.

search time. If you behave irrationally in anger — for instance, if someone has injured you and you successfully sue him, but you spend more money on lawyer's bills than you collect — well, maybe that's not actually good for you, but it's good for society, because it will discourage the next guy from doing injuries in the expectation that it's not worth suing him.

No doubt there are elements of truth here, but taken as an all-purpose description of cognition, I find it Panglossian and implausible. You put down your keys somewhere and spend the next twenty minutes looking for them. You lazily put a paper on top of a pile on your desk even if you know perfectly well you will not need it again for months. People do all kinds of stupid, destructive things in anger that benefit neither them nor society, neither directly nor indirectly. Small children are notorious for insisting on the same bedtime story or playing the same song or eating only three different foods for weeks on end; this runs completely counter to the rational approach to explore/exploit that Christian and Griffiths attribute to them.

5 Conclusion

Taking the book as a whole, it seems to me that what it actually demonstrates is that the algorithmic ideas being produced in computer science are only occasionally useful in the conduct of everyday life or as models of ordinary human cognition. Not at all surprisingly, the vast majority of successful applications of the ideas of computer science are in computer technology.

Acknowledgements

Thanks to Andrew Sundstrom for helpful feedback.

References

Daniel Sleator and Robert Tarjan, "Amortized efficiency of list update and paging rules," *Communications of the ACM*, **28**:2, Feb. 1985, 202-208.