

What is an Algorithm?

YAP Von Bing

National University of Singapore

Numerical procedures are classified into two types according to whether there is a well-defined goal. The distinction is illustrated with well-known examples. Quite different is a “prediction algorithm”, which resonates with the implicit meaning of “algorithm” in popular discussions on social media and artificial intelligence. Some implications of the distinctions on the pedagogy of computational thinking and the society at large are briefly discussed.

Keywords: algorithm, computation, prediction

Examples of Algorithms

The word “algorithm” was derived from the name Muhammad ibn Musa al-Khwarizmi, of a 9th-century Persian polymath. His most important work, *The Compendious Book on Calculation by Completion and Balancing* (*al-Kitab al-Mukhtasar fi Hisab al-Jabr¹ wal-Muqabalah* in Arabic), presents steps for solving all quadratic equations with real roots. Since only positive numbers were used, there are several types of equations, with corresponding instructions. Now, armed with negative and complex numbers, we just have one case:

Let $a \neq 0, b, c$ be real numbers. The roots of $ax^2 + bx + c = 0$ are given by the quadratic formula

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

It is a most fitting algorithm to pay tribute to the genius of Al-Khwarizmi.

Around 300 BC, *The Elements* was completed by Euclid. The masterpiece on geometry included an algorithm for computing the greatest common divisor of two positive integers a and b , which is still in use.

Let $r_{-2} = a, r_{-1} = b$.

Calculate integers q_0 and r_0 so that $r_{-2} = q_0 r_{-1} + r_0$ and $0 \leq r_0 < r_{-1}$.

If $r_0 > 0$, calculate integers q_1 and r_1 so that $r_{-1} = q_1 r_0 + r_1$ and $0 \leq r_1 < r_0$.

...

If $r_{k-1} > 0$, calculate integers q_k and r_k so that $r_{k-2} = q_k r_{k-1} + r_k$ and $0 \leq r_k < r_{k-1}$.

There is an $n \geq 0$ such that $r_n = 0$. Then $\gcd(a, b) = r_{n-1}$.

¹ The origin of “algebra”.

Unlike the quadratic formula, the number of calculations in the Euclidean algorithm depends on the input. However, in both algorithms, there is a well-defined goal, and the instructions are guaranteed to attain that goal. Algorithms are ubiquitous in the astounding range of computational systems that are increasingly intertwined with human life. Pedagogically, algorithms are wonderful vehicles for learning mathematical reasoning (for example, why does the Euclidean algorithm stop, and why is $\gcd(a, b) = r_{n-1}$?) and for learning a programming language. The clear goal is especially useful for debugging codes.

In the early 19th century, investigations in astronomy and physics often presented the following problem. Given known quantities $y_i, 1 \leq i \leq n$ and $x_{ij}, 1 \leq i \leq n, 1 \leq j \leq p$ related by the equations

$$y_i = b_0 + b_1x_{i1} + \cdots + b_px_{ip} + \epsilon_i, 1 \leq i \leq n,$$

the goal is to determine the constants b_j 's. This is impossible because the ϵ_i 's are unknown constants. Mathematicians such as Gauss and Legendre moved away from an exact solution to an approximate one (Stigler, 1990): find the values of b_j 's that minimise the sum of squares

$$\sum_{i=1}^n (y_i - b_0 - b_1x_{i1} - b_2x_{i2} - \cdots - b_px_{ip})^2.$$

Let $y = (y_1, \dots, y_n)'$ be a column vector, X be the $n \times (p+1)$ matrix containing the x 's, and a column of 1's on the left. X , and $b = (b_1, \dots, b_p)'$. If the rank of X is $p+1$, then the minimiser is unique, given by the least squares (LS) coefficients

$$b = (X'X)^{-1}X'y.$$

Like the two previous algorithms, there is a clear goal. Proving the algorithm works is a worthwhile intellectual journey that allows the student to use tools from calculus and algebra, and an excellent gateway to the subject of high-dimensional Euclidean spaces. The minimum sum of squares is called SSE ("Sum of Squares Error"), which is less than the SST ("Sum of Squares Total"): $\sum_{i=1}^n (y_i - \bar{y})^2$, $\bar{y}$ being the mean of $y_1, \dots, y_n$. The coefficient of determination is $R^2 = 1 - \frac{\text{SSE}}{\text{SST}}$. The approximate solution is better if R^2 is closer to 1.

Prediction

The least squares algorithm is often used in prediction. For $i = 1, \dots, n$, let $y_i, x_{i1}, \dots, x_{ip}$ be values assumed by item i on a response variable, denoted y , and explanatory variables, denoted $x_1, \dots, x_p$. The data on the n individuals, called the "training data", can be used to predict y_* on a new item, which has the values $x_{*1}, \dots, x_{*p}$ on the explanatory variables. The simplest idea is to use $b_0 + b_1x_{*1} + b_2x_{*2} + \cdots + b_px_{*p}$, where $b = (X'X)^{-1}X'y$ are the LS coefficients. If $x_{*1}, \dots, x_{*p}$ are the values of a new item on the explanatory variables, then y_* is predicted as $b_0 + b_1x_{*1} + b_2x_{*2} + \cdots + b_px_{*p}$. Essentially, this is why the algorithm is the cornerstone of the vast field of regression, which is crucial in the physical and social sciences, as well as numerous other domains dealing with data analysis. In particular, the least squares algorithm is in the heart of the astounding developments in machine learning and artificial intelligence.

An algorithm contains a sequence of computations. Many are better viewed as a set of smaller algorithms. For instance, the algorithm that computes the quotient and remainder from dividing an integer by another is used many times in the Euclidean algorithm. The least squares algorithm rides on algorithms for multiplying matrices and for computing the matrix inverse. Conversely, a sequence of computations is often considered an algorithm. I want to make the

case that if there is no clear goal, then we are on to something else. An example is from the topic of variable selection.

We have seen how y can be predicted from $x_1, \dots, x_p$, given some training data. If we were to use a subset of the explanatory variables, then the corresponding R^2 will be less than before. However, experience shows that omitting some variables can lead to better performance on future predictions. How to choose such variables, given only the training data? This is the problem of variable selection, which has always been important in statistics, and is ubiquitous in the design of various tools in machine learning and artificial intelligence.

A reasonable strategy is to choose the subset of variables that maximises a quantity called the adjusted coefficient of determination. But it is impractical to search over the 2^p cases when p gets even moderately large, like 100. There are many incremental approaches which are computationally feasible. A relatively simple idea is backward elimination, which will be illustrated using a famous dataset of Woods et al. (1932). The variables are heat in calories per gram of cement (y), tricalcium aluminate (x_1), tricalcium silicate (x_2), tetracalcium aluminoferrite (x_3), and dicalcium silicate (x_4), measured in 13 cement mixtures. Following Montgomery et al. (2012), we use a threshold of 0.1 to decide whether a variable should be eliminated. In the first step, all four variables are used, giving the P values 0.071, 0.501, 0.896, 0.844, to 3 decimal places. Loosely speaking, a larger value indicates less importance, so the least important variable is x_3 , and the most important one is x_1 . Since 0.896 exceeds 0.1, we leave out x_3 , thus concluding the first step. Next, only x_1, x_2, x_4 are used, giving the P values 0.000, 0.052, 0.205. Since 0.205 exceeds 0.1, we eliminate x_4 . In the third step, only x_1, x_2 are used, and since both P values are 0.000, there is no further elimination. Hence, backward elimination (with threshold 0.1) has resulted in the recommendation to use only x_1 and x_2 to predict y .

Backward elimination consists of a sequence of computations, but there does not seem to be a simple description of the output. For some inputs, it is the subset of variables with the maximum adjusted coefficient of determination, but not for all. It is as if backward elimination conjures its goal only through the computations. This is not a denigration of its practical utility, but to point out that it is distinct from the three previous algorithms. If we are to call backward elimination an algorithm, it seems wise to qualify it, say as a “type 2 algorithm”. There are many other variable selection procedures, including forward selection and stepwise variable selection (Harrell, 2015), and the overwhelming majority of them are type 2 algorithms.

The fact that type 1 algorithms (retroactively designated) can be proved to work (for example, that the Euclidean algorithm indeed outputs $\gcd(a, b)$), means they belong to the realm of mathematics, rather than science. Type 2 algorithms are part of mathematics too, since they are analogous to a familiar situation where a conjecture is yet to be formulated out of a sea of isolated facts. One day the result of backward elimination may be shown to have certain simple characteristics. Then the procedure may be reclassified as type 1.

Prediction Method or Procedure

A major criterion of success of a scientific theory is the extent to which it predicts future observations. Although algorithms are often part of a prediction method, the method itself is different from an algorithm, because the result can be wrong, i.e., the predicted value can differ

from the actual value by an amount that cannot be explained by random fluctuations. Indeed, such discrepancies provide impetus to search for a better theory. Suppose that based on a statistical analysis of recent data, an insurance company estimates that a healthy non-smoking 40 year-old man has a 50% chance of having a heart attack later in life. If this is true of 1,000,000 new clients who are healthy non-smoking 40 year-old men, then there is a more than 0.9999 chance that between 49.8% and 50.2% of them will get a heart attack. This is because the standard deviation in the percentage is $0.5/1000 = 0.05\%$, and the standard normal variable has more than 0.9999 chance of being between -4 and 4 . Therefore, if it turns out that only 49% have no heart attack before death, the prediction of 50% can be viewed as wrong. The insurer would wish to have made a prediction somewhat higher than 50%, like 51%, which would have led to a smaller loss.

To predict is to risk being wrong. Moreover, it is fair to say that a prediction is almost always wrong, to the extent that making predictions may be strangely liberating. In contrast, an algorithm is never wrong: a type 1 is provably correct, and a type 2 cannot be proved wrong. The physicist Niels Bohr and the baseball player Yogi Berra are among many people to whom is attributed the quote: “It is difficult to make predictions, especially about the future.” Albert Einstein, keenly aware of the distinction between mathematics and science, stated in his 1921 address *Geometry and Experience*, “As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.” Even the mathematicians have a similar experience when a conjecture is falsified by a counter-example. But since there is a clear distinction between theorems and conjectures, mathematics does not need correction, unlike the sciences.

Given the sharp divide between prediction and algorithms, “prediction algorithm” is a rather strange collocation. If its goal is to make predictions about the real world, then its computation steps go beyond what algorithms do. “Prediction method” or “prediction procedure” seem better, but since the ship has long sailed, the least we can do in its wake is to be discerning.

“Algorithm” in Popular Media

In the popular media, the meaning of “algorithm” plunges beyond the slippery slope of “prediction algorithm”. Here is an attempt to interpret the word in several quotes from *BBC News*, *The Guardian*, and *The New York Times*.

Isaac and Frenkel (2023):

“The algorithms powering Facebook and Instagram, which drive what billions of people see on the social networks, have been in the cross hairs of lawmakers, activists and regulators for years. Many have called for the algorithms to be abolished to stem the spread of viral misinformation and to prevent the inflammation of political divisions.”

Rawlinson (2023), on TikTok:

“It is the home of dance tutorial videos and viral comedy sketches. But it is also host to self-harm and eating disorder content, with an algorithm that has been called the “crack cocaine of social media”. ... changing a user’s name from “Sarah” to “Sarah Lose Weight” on TikTok could result in its algorithm serving 12 times more self-harm content.”

“Algorithm” seems to refer to a recommender system, which uses past data and recently chosen items by a user to automatically generate certain new items that could interest the user. Essentially, it produces a prediction of what the user might enjoy.

Spicer (2023):

“The algorithm misinterpreted “strong women” as “massive biceps”. My book, Man-Made, is about stereotypical images and words in datasets being used to train algorithms.”

The Euclidean algorithm does not interpret and does not get trained, since it is already doing its job. By “algorithm”, the author seems to mean a classifier — a set of rules to decide whether an input represents one of several possible types of objects.

Barratt et al. (2023):

“Despite companies’ constantly increasing reliance on data-churning tech, the way most algorithms work remains a mystery to many workers — which creates a power imbalance on the job. ... In the workplace, self-learning algorithmic computer systems are being introduced by companies to assist in areas such as hiring, setting tasks, measuring productivity, evaluating performance and even terminating employment: “I’m sorry, Dave. I’m afraid you are being made redundant.””

Bria (2023):

“Hollywood’s recent battles between screenwriters and the robots threatening their jobs should have been a wake-up call for millions of European workers. Who will protect our working rights from the algorithms, which increasingly act like our bosses?... Amazon best represents a paradigm shift in which the algorithm becomes the boss. The company’s reliance on algorithms to oversee worker productivity in its warehouses is under increasing legal scrutiny, with workers complaining that the constant use of tracking technology is dehumanising. Algorithms are king at Amazon: hiring, evaluating and firing millions of people with little or no human supervision.”

Here, “algorithm” seems to mean an automated management system, which is somewhat like a so-called algorithmic trading system, where decision rules are automated.

Conclusion

According to Oxford Languages, Google’s English dictionary, an algorithm is *a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer*. Thus, an instruction for baking a cake is an algorithm. Focusing on computational problems, this article proposes that the presence of a clear goal should be a distinctive attribute of an algorithm. The condition is certainly satisfied by many well-known examples beyond the quadratic formula, the Euclidean algorithm, and the least squares algorithm. A basic objective of computational thinking as a pedagogical project must be for the student to understand that the goal is indeed accomplished by the instruction, eventually either via the implementation of a computer code or the construction of a logical proof. Given the consensus that mathematical proof is not suitable for very young children, we should also not be too eager to teach them coding. A rudimentary understanding can be achieved by patiently working through several well-chosen concrete examples, and this activity should precede the systematic learning of

coding or proving. There are many interesting and useful algorithms, which are more than sufficient for cultivating computational thinking in pre-university education.

The problem of regression variable selection has been attacked with a variety of methods. For such methods, it is often not easy to describe their goals, beyond the vague “best model” which often merely emerges from the computations. These procedures are called type 2 algorithms, to distinguish from the type 1 algorithms described in the previous paragraph. Since type 2 algorithms are encountered by students only in the university, there is no confusion in referring to type 1 algorithms simply as “algorithms” in pre-university education. Both types belong to mathematics: the inputs and outputs are purely mathematical objects. Whether these objects correspond closely to real world objects is another matter.

The fact that a type 2 algorithm has no clear goals can give the impression that it is like a prediction method for some future observation. After all, one does not know what to expect about the “correct” answer beforehand, and has to wait for it to be revealed upon the completion of the computation, like when an observation is finally made. While the analogy in psychological perceptions seems understandable, it is worth repeating that type 2 algorithms are confined to the mathematical world, and that prediction methods or procedures are firmly rooted in the real world, even if they are called “prediction algorithms”.

The astoundingly remarkable advances in machine learning and artificial intelligence are destined to change the world drastically. Concerns on how lives might be adversely affected are mounting, and the struggle to confront the new reality inevitably births linguistic innovations². The word “algorithm” has been imbued with multiple new meanings (Recommender system? Classifier? Automated management system?), as demonstrated by the newspaper quotes, and numerous other online sources. Book titles are also revealing: *The Algorithm: How AI can hijack your career and steal your future* by Hilke Schellmann, and *Is the Algorithm Plotting Against Us?* by Kenneth Wenger. Years later, we might look back at the last decade as the cradle of contesting new meanings of “algorithm”. While we could insist on a narrower interpretation, it may be wise to accept an additional meaning of “algorithm”, as something like a machine that automatically makes decisions or judgments. We can be hopeful that if or when “algorithm” becomes a widely loathed term because of the new meaning, the original meaning still endures, so that algorithms continue to work where they are needed. And they are indeed indispensable in a wide array of human activities, humming away quietly. It will be a travesty if a popular backlash against artificial intelligence brings algorithms down as collateral damage.

Acknowledgements

I dedicate this paper to my father, who taught me gardening, scribing, and other things.

References

Barratt, T., Veen, A., & Goods, C. (2020, August 28). How algorithms keep workers in the dark. *BBC News*. <https://www.bbc.com/worklife/article/20200826-how-algorithms-keep-workers-in-the-dark>

² Or aberrations. The pre-eminent lexicographer Susie Dent says “English has always evolved by mistake.” Like the DNA molecule.

- Bria, F. (2023, October 12). Algorithms are hiring and firing us now – but tougher EU laws could protect workers. *The Guardian*.
<https://www.theguardian.com/commentisfree/2023/oct/12/algorithms-eu-law-workers-ai-data>
- Harrell, F. E. Jr. (2015). *Regression Modeling Strategies*. Springer.
- Isaac, M. & Frenkel, S. (2023, July 27). Facebook’s algorithm is ‘influential’ but doesn’t necessarily change beliefs, researchers say. *The New York Times*.
<https://www.nytimes.com/2023/07/27/technology/facebook-instagram-algorithms.html>
- Montgomery, D. C., Peck, E. A., & Vining G. G. (2012). *Introduction to Linear Regression Analysis*. Wiley.
- Rawlinson, K. (2023, April 4). How TikTok’s algorithm ‘exploits the vulnerability’ of children. *The Guardian*. <https://www.theguardian.com/technology/2023/apr/04/how-tiktoks-algorithm-exploits-the-vulnerability-of-children>
- Spicer, T. (2023, February 23). AI art replicates inequality at scale. We need to learn about its biases – and outsmart the algorithm. *The Guardian*. <https://www.theguardian.com/books/2023/feb/24/ai-art-replicates-inequity-at-scale-we-need-to-learn-about-its-biases-and-outsmart-the-algorithm>
- Stigler, S. M. (1990). *The History of Statistics: The Measurement of Uncertainty before 1900*. Harvard University Press.
- Woods, H., Steinour, H. H., & Starke, H. R. (1932). Effect of composition of Portland cement on heat evolved during hardening. *Industrial Engineering and Chemistry*, 24, 1207–1214.

Author:

YAP Von Bing (stayapvb@nus.edu.sg),

Department of Statistics and Data Science, National University of Singapore, 6 Science Drive 2, Singapore 117546.