Linear Regression under Partial Information

Tho M. Nguyen
Ho-Chi-Minh City Open University, tho.nm@ou.edu.vn

Saeid Tizpaz-Niari
The University of Texas at El Paso, saeid@utep.edu

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Comments:
Technical Report: UTEP-CS-23-53

Linear Regression under Partial Information

Tho M. Nguyen, Saeid Tizpaz-Niari, and Vladik Kreinovich

Abstract Often, we need to know how to estimate the value of a difficult-to-directly estimate quantity y – e.g., tomorrow's temperature – based on the known values of several quantities $x_1, \ldots, x_n$. In many practical situations, we know that the relation between y and x_i can be accurately described by a linear function. So, to find this dependence, we need to estimate the coefficients of this linear dependence based on the known cases in which we know both y and x_i; this is known as *linear regression*. In the ideal situation, when in each case, we know all the inputs x_i, the computationally efficient and well-justified least squares method provides a solution to this problem. However, in practice, some of the inputs are often missing. There are heuristic methods for dealing with such missing values, but the problem is that different methods lead to different results. This is the main problem with which we deal in this paper. To solve this problem, we propose a new well-justified method that eliminates this undesirable non-uniqueness. An auxiliary computational problem emerges if after we get a linear dependence of y on x_i, we learn the values of an additional variable x_{n+1}. In this case, in principle, we can simply re-apply the least square method "from scratch", but this idea, while feasible, is still somewhat time-consuming, so it is desirable to come up with a faster algorithm that would utilize the previous regression result. Such an algorithm is also provided in this paper.

Tho M. Nguyen
Faculty of Banking and Finance, Ho-Chi-Minh City Open University, Ho-Chi-Minh City, Vietnam
e-mail: tho.nm@ou.edu.vn

Saeid Tizpaz-Niari and Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA
e-mail: saeid@utep.edu, vladik@utep.edu

1 Formulation of the Main Problem

The main objective of this paper is to show how to solve linear regression problem under partial information. To explain why this is important, we first explain why linear regression is an important problem, how linear regression problems are solved now, and why the case of partial information is a challenge.

Why linear regression. One of the main objectives of science is to predict the future state of the world based on the information about its current state. At any given moment of time, the state of the world can be described by listing the values of the quantities $x_1, \ldots, x_n$ that characterize this state. Thus, the objective is to predict the future value y of each quantity of interest based on its current values.

In some cases – e.g., in celestial mechanics – we know how exactly the future value y depends on the available information $x_1, \ldots, x_n$, i.e., we know the function $y = f(x_1, \ldots, x_n)$ that computes y based on the inputs x_i. However, in many other cases, we do not know this function, we must determine it based on the available data, namely, based on the previous cases $k = 1, \ldots, K$ in which we knew both the values $x_i^{(k)}$ and the value $y^{(k)}$. In this situation, we need to find a function $f(x_1, \ldots, x_n)$ for which, for all k from 1 to K, we have $y^{(k)} \approx f\left(x_1^{(k)}, \ldots, x_n^{(k)}\right)$. The procedure of finding such a function has been traditionally known as *regression* – and when we use computers to find this function, this is also known as *machine learning*.

The dependence of y on x_i is often smooth; see, e.g., [4, 8], and in many practical situations, the changes are relatively small. In such cases, for each i, all the values $x_i^{(k)}$ are close to the first value $x_i^{(1)}$. We can then represent the desired dependence in terms of the differences $\Delta x_i \stackrel{\text{def}}{=} x_i - x_i^{(1)}$, as

$$y = f\left(x_1^{(1)} + \Delta x_1, \ldots, x_n^{(1)} + \Delta x_n\right). \tag{1}$$

Since the differences Δx_i are small, terms which are quadratic in Δx_i (or of higher order) can be safely ignored; see, e.g., [4, 8]. If we expand (1) in Taylor series in terms of Δx_i and ignore quadratic and higher order terms in this expansion, we get a linear dependence

$$y = y_0 + a_1 \cdot \Delta x_1 + \ldots + a_n \cdot \Delta x_n, \tag{2}$$

where $y_0 \stackrel{\text{def}}{=} f\left(x_1^{(1)}, \ldots, x_n^{(1)}\right)$ and

$$c_i \stackrel{\text{def}}{=} \frac{\partial f}{\partial x_i}_{|x_1 = x_1^{(1)}, \ldots, x_n = x_n^{(1)}}.$$

Substituting the expression $\Delta x_i = x_i - x_i^{(1)}$ into the formula (2), we get

$$y = a_0 + a_1 \cdot x_1 + \ldots + a_n \cdot x_n, \tag{3}$$

where we denoted
$$a_0 \stackrel{\text{def}}{=} y_0 - a_1 \cdot x_1^{(1)} - \ldots - a_n \cdot x_n^{(1)}.$$

In this case, regression (or machine learning) means finding the coefficients a_i of the linear dependence (3). This task is known as *linear regression.*

Why least squares: first explanation. In practice, we rarely have full information about the current state of the world. As a result, we can only make approximate predictions. In particular, for each k, instead of the exact formula (3), we only have an approximate equality

$$y^{(k)} \approx a_0 + a_1 \cdot x_1^{(k)} + \ldots + a_n \cdot x_n^{(k)}. \tag{4}$$

If we denote the difference between the left-hand side and the right-hand side of the formula (4) by $\varepsilon^{(k)}$, then this approximate equality takes the form

$$y^{(k)} = a_0 + a_1 \cdot x_1^{(k)} + \ldots + a_n \cdot x_n^{(k)} + \varepsilon^{(k)}. \tag{5}$$

We do not know the approximation errors

$$\varepsilon^{(k)} = y^{(k)} - \left(a_0 + a_1 \cdot x_1^{(k)} + \ldots + a_n \cdot x_n^{(k)}\right). \tag{6}$$

Such unknown values are usually called *random*.

Usually, there are many different factors that contribute to the approximation error. In this case, according to the Central Limit Theorem (see, e.g., [7]), the probability distribution of the approximation error is close to Gaussian (normal). Thus, it is reasonable to conclude that the approximation error (6) is normally distributed. A normal distribution is uniquely characterized by two parameters: its mean m and its standard deviation σ.

If the mean m is different from 0, then we can add this mean to a_0 and subtract it from $\varepsilon^{(k)}$, thus getting

$$y^{(k)} = a_0' + a_1 \cdot x_1^{(k)} + \ldots + a_n \cdot x_n^{(k)} + \varepsilon'^{(k)},$$

where we denoted $a_0' \stackrel{\text{def}}{=} a_0 + m$ and $\varepsilon'^{(k)} \stackrel{\text{def}}{=} \varepsilon^{(k)} - m$. Then, the mean of the values $\varepsilon' = \varepsilon - m$ is equal to 0. Thus, without loss of generality, we can always assume that $m = 0$. Under this assumption, the normal distribution is uniquely determined by a single parameter σ.

According to the formula for the normal distribution, the probability density ρ corresponding to all the measurement results is equal to

$$\rho = \prod_{k=1}^{K} \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot \exp\left(-\frac{\left(\varepsilon^{(k)}\right)^2}{2\sigma^2}\right),$$

i.e., equivalently, to

$$\rho = \frac{1}{(\sqrt{2\pi}\cdot\sigma)^K}\cdot\exp\left(-\frac{\sum\limits_{k=1}^{K}\left(\varepsilon^{(k)}\right)^2}{2\sigma^2}\right). \tag{7}$$

It is reasonable to select the most probable values a_i, i.e., the values for which the probability density ρ is the largest. This idea is known as the *Maximum Likelihood approach*. Since the function $\exp(-x)$ is strictly decreasing, maximizing the expression (7) is equivalent to minimizing the expression under the exponential function, i.e., minimizing the sum

$$\sum_{k=1}^{K}\left(\varepsilon^{(k)}\right)^2, \tag{8}$$

where $\varepsilon^{(k)}$ is defined by the formula (6). This idea of minimizing the sum of the squares is known as the *least squares* approach.

Why least squares: second explanation. For each k, we want the left-hand side of the formula (4) should be close to the right-hand side. In other words, the vector $\left(y^{(1)},\ldots,y^{(K)}\right)$ formed by the left-hand sides should be as close as possible to the vector formed by the right-hand sides. A natural way to describe the distance $d(u,v)$ between two K-dimensional vectors $u=(u_1,\ldots,u_K)$ and $v=(v_1,\ldots,v_K)$ – i.e., equivalently, between two points in the K-dimensional space – is to use the usual Euclidean formula

$$d^2(u,v)=(u_1-v_1)^2+\ldots+(u_K-v_K)^2.$$

In our case, for each k, the difference u_k-v_k between the left-hand side and the right-hand side is equal to $\varepsilon^{(k)}$. Thus, the square of the distance between these two vectors has the form (8). Since for all the values $x\ge 0$, x^2 is a strictly increasing function, minimizing the distance is equivalent to minimizing its square. Thus, we also arrive at the least squares approach.

How the least squares problem is solved now. When we plug in the expression (6) into the formula (8), we get the expression

$$\sum_{k=1}^{K}\left(y^{(k)}-\left(a_0+a_1\cdot x_1^{(k)}+\ldots+a_n\cdot x_n^{(k)}\right)\right)^2.$$

Differentiating this expression with respect to all the unknown a_i, equating each of the partial derivatives to 0, and dividing both sides of each equation by K, we get the following system of linear equations (see, e.g., [7]):

$$a_0+\sum_{i=1}^{n}a_i\cdot\overline{x_i}=\overline{y} \tag{9}$$

and

$$a_0 \cdot \overline{x_i} + \sum_{j=1}^{n} a_i \cdot \overline{x_i \cdot x_j} = \overline{x_i \cdot y}, \quad i = 1, \ldots, n, \tag{10}$$

where we denoted

$$\overline{x_i} \stackrel{\text{def}}{=} \frac{1}{K} \cdot \sum_{k=1}^{K} x_i^{(k)}, \overline{y} \stackrel{\text{def}}{=} \frac{1}{K} \cdot \sum_{k=1}^{K} y^{(k)}, \overline{x_i \cdot x_j} \stackrel{\text{def}}{=} \frac{1}{K} \cdot \sum_{k=1}^{K} \left(x_i^{(k)} \cdot x_j^{(k)} \right), \text{ and}$$

$$\overline{x_i \cdot y} \stackrel{\text{def}}{=} \frac{1}{K} \cdot \sum_{k=1}^{K} \left(x_i^{(k)} \cdot y^{(k)} \right). \tag{11}$$

For each i, we can multiply both sides of the equation (9) by $\overline{x}_i$ and subtract this result from the corresponding formula (10), then we get the following simplified equation not containing a_0:

$$\sum_{j=1}^{n} a_i \cdot C(x_i, x_j) = C(x_i, y), \tag{12}$$

where, for every two quantities q and q', $C(q, q')$ denotes the covariance

$$C(q, q') = \overline{q \cdot q'} - \overline{q} \cdot \overline{q'}; \tag{13}$$

(for $q = q'$, this is simply the variance $V(q) = \overline{q^2} - (\overline{q})^2$ of the quantity q).

Equations (12) form a system of n linear equations

$$Ca = b \tag{14}$$

to determine the vector $a = (a_1, \ldots, a_n)$ of n unknowns, where:

- C is the matrix formed by the coefficients $C(x_i, x_j)$, and
- $b = (C(x_1, y), \ldots, C(x_n, y))$ is the vector formed by the right-hand sides.

A general solution to this linear system has the form

$$a = C^{-1} b, \tag{15}$$

where C^{-1} is the matrix which is inverse to the matrix C. Once we know the coefficients $a_1, \ldots, a_n$, we can determine a_0 by using the formula (9):

$$a_0 = \overline{y} - \sum_{i=1}^{n} a_i \cdot \overline{x_i}. \tag{9a}$$

Case of partial information is a challenge. To use the above formulas, we need to know, for each k, the values $y^{(k)}$ and $x_i^{(k)}$ of the output y and of all the inputs x_i. In many practical situations, however, we only have partial information. For example, in medical applications, we would like to be able to predict the patient's progress based on the parameters characterizing the state of the patient and on the doses of the corresponding medicine. Some of these parameters we usually know – e.g., age, height, weight, etc. However, other parameters are determined by the laboratory tests, and different patients may take different tests. In economics, we may have different reporting schedules for different countries and/or different companies, which also leads to partial information.

In such situations, we often have very few cases k in which all the values x_i are available – not enough to make statistically significant conclusions about the dependence of the desired output y on the inputs x_i.

How such situations are handled now. The usual way of dealing with situations with partial information is the *missing data* approach, when we use reasonable interpolation techniques to estimate the missing values, and then apply the least squares approach to such filled-in data; see, e.g., [1, 5, 6, 9].

Limitations of the current approach. While missing data techniques work in many practical situations, they have two issues:

- first, many interpolation techniques used in the missing data approach are heuristic, and different techniques often lead to different results;
- second, the resulting coefficients a_i enable us to predict y for the case when we know all the inputs x_i; however, in practice, as we have mentioned, some of the inputs are often missing; so, to apply the linear formulas in such cases, we need to perform another interpolation – which also leads to undesirable non-uniqueness.

What we do in this paper. In this paper, we propose an alternative idea that enables us to make linear regression under partial information well-justified and thus, avoid the undesired non-uniqueness.

2 First Issue: What We Propose

Main idea. As we have mentioned, the first issue is that often, we have very few cases when we know the values of all the inputs x_i – not enough to get statistically significant estimates for the desired coefficients a_i.

Our idea comes from the fact that to find the coefficients a_i, all we need to know are the mean values $\overline{x}_i$, $\overline{y}$, and the covariances $C(x_i, x_j)$ and $C(x_i, y)$. To find these mean values and covariances, it is not necessary to have *all n* inputs in each case: it is sufficient to have, for each pair (i, j), a sufficient number of cases in which we know the values of these two quantities – which is usually the case. Thus, there is no need to interpolate – we can simply use the available data. So, we arrive at the following algorithm.

Resulting algorithm. For each case $k = 1, \dots, K$, by $A(k)$, we will denote the list of all the quantities x_i and y whose values are available in this case. Then, based on the available values, we can compute the following estimates for the means and covariances:

$$\overline{x_i} \overset{\text{def}}{=} \frac{1}{\#\{k : x_i \in A(k)\}} \cdot \sum_{k:\, x_i \in A(k)} x_i^{(k)}, \overline{y} \overset{\text{def}}{=} \frac{1}{\#\{k : y \in A(k)\}} \cdot \sum_{k:\, y \in A(k)} y^{(k)},$$

$$\overline{x_i \cdot x_j} \overset{\text{def}}{=} \frac{1}{\#\{k : x_i, x_j \in A(k)\}} \cdot \sum_{k:\, x_i, x_j \in A(k)} \left(x_i^{(k)} \cdot x_j^{(k)}\right), \text{ and}$$

$$\overline{x_i \cdot y} \overset{\text{def}}{=} \frac{1}{\#\{k : x_i, y \in A(k)\}} \cdot \sum_{k:\, x_i, y \in A(k)} \left(x_i^{(k)} \cdot y^{(k)}\right). \tag{16}$$

Then, we can find the covariances $C(x_i, x_j)$ and $C(x_i, y)$ by using the formula (13), and find the coefficients $a_1, \ldots, a_n$ by solving the linear system (14). After that, we can estimate the remaining coefficient a_0 by using the formula (9a).

3 Second Issue: What We Propose

Discussion. As we have mentioned, another case where heuristic interpolation techniques are used is when we already have estimates for the coefficients a_i of linear regression, but we cannot directly use them for prediction, since we do not know all the inputs x_i. Strictly speaking, in the case when we only know the values of some inputs $x_{i_1}, \ldots, x_{i_m}$, we need to consider a new linear regression problem: finding the coefficients $a'_0, a'_1, \ldots, a'_m$ for which

$$y \approx a'_0 + a'_1 \cdot x_{i_1} + \ldots + a'_m \cdot x_{i_m}. \tag{17}$$

In principle, we can run the general least-squares-under-partial-information procedure again, but that would be too time-consuming: we would again need to analyze all the data etc. How can we get well-justified results without starting “from scratch"?

To do that, we can use the fact that all we need to find the new coefficients are the mean values $\overline{x}_{i_j}$, $\overline{y}$, and the covariance values $C(x_{i_j}, x_{i_\ell})$ and $C(x_{i_j}, y)$ corresponding to the available variables – but these values we have already computed when we solved the original linear-regression-under-partial-information problem. Thus, we arrive at the following algorithm.

Resulting algorithm. When we estimated the coefficients a_i, we found the values of all the means $\overline{x}_i$ and all the covariances $C(x_i, x_j)$ and $C(x_i, y)$. To find the regression coefficients $a'_1, \ldots, a'_m$ for the case when we only know the values of some inputs $x_{i_1}, \ldots, x_{i_m}$, we then solve the linear system

$$C'a' = b', \tag{18}$$

where:

- C' is the square submatrix of the matrix C obtained by selecting only rows and columns $i_1, \ldots, i_m$, and
- b' is a subvector of the vector b corresponding to indices $i_1, \ldots, i_m$.

Then, we can find a'_0 as

$$a'_0 = \overline{y} - \sum_{j=1}^{m} a'_i \cdot \overline{x_{i_j}}.$$

4 Auxiliary Problem

Formulation of the problem. In the previous sections, we considered a typical situation when some values of the inputs $x_1, \ldots, x_n$ were missing. Of course, the more data we have, the more accurate will be our predictions. Thus, researchers are always trying to get more data. In particular, they are trying – and often succeeding – to find ways to measure new characteristics x_{n+1}, etc. that could be helpful for predictions.

Once we have the values of a new quantity x_{n+1}, we can hopefully get a new linear formula that uses this new quantity:

$$y \approx a'_0 + a'_1 \cdot x_1 + \ldots + a'_n \cdot x_n + a'_{n+1} \cdot x_{n+1}. \tag{19}$$

One way to find the new values a'_i is to repeat the same procedure as before. The only missing information are the mean values $\overline{x}_{n+1}$ of the new quantity and the missing correlations $C(x_i, x_{n+1})$, $C(x_{n+1}, x_{n+1})$, and $C(x_{n+1}, y)$. Once we compute these values, we will get the new – extended – matrix C', the new – extended – vector b' and we will then be able to solve the new systems of linear equations $C'a' = b'$.

The problem is that while solving a system of linear equations is feasible, it is still somewhat time-consuming. Is it possible to use the results of the original linear regression to speed up these computations?

What we propose. Actually, in this section, we do not propose any new algorithm, our proposal is to use the fact that if we know the inverse C^{-1} to a symmetric matrix C, then we can explicitly compute the inverse of a symmetric matrix obtained by adding one row and one column (see, e.g., [2]):

$$\begin{pmatrix} C & e \\ e^T & v \end{pmatrix}^{-1} = \begin{pmatrix} C^{-1} + zC^{-1}ee^TC^{-1} & -zC^{-1}e \\ -ze^TC^{-1} & z \end{pmatrix},$$

where

$$z \stackrel{\text{def}}{=} \frac{1}{v - e^TC^{-1}e}.$$

This way, we need n^2 arithmetic operations to compute the new inverse, while even the asymptotically fastest algorithm for solving systems of linear equations and for computing the inverse matrix require larger time – namely, time proportional to n^a for $a > 2$; see, e.g., [3].

Important case. An important case is when the new quantity x_{n+1} has no correlations with any of the previously available quantities $x_1, \ldots, x_n$. In this case, $e = 0$, so we have:

- $a'_i = a_i$ for $i = 1, \ldots, n$;
- $a'_{n+1} = (C(x_{n+1}, x_{n+1}))^{-1}$; and
- $a'_0 = a_0 - a'_{n+1} \cdot \overline{x}_{n+1}$.

5 Experiments

We design two sets of experiments to show the accuracy and run-time efficiency of proposed algorithms. The first set of experiments is designed to show the precision loss (predicted vs. actual parameters) as the percentage of missing values are increasing in a linear system. The second set of experiments is designed to show the run-time efficiency of the proposed algorithm to obtain the missing coefficients in comparison to the existing least square algorithm from scratch.

5.1 Experiments on Missing Values

We consider a linear system of $\mathbf{y} = a \cdot \mathbf{x} + \epsilon$ where a is a vector of coefficients, x is a vector of variables, and ϵ is a random error. We generate a and x independently randomly from a normal distribution with mean 0 and standard deviation 1 (the random error has mean 0 and standard deviation 0.1). We generate 10,000 samples from the linear equation ($y^{(1)} \ldots y^{(10{,}000)}$ and $x^{(1)} \ldots x^{(10{,}000)}$) where the number of attributes (parameters) is 50. Then, we randomly select k% of the samples and set 95% of their attributes to NA (missing values). We repeat this experiment for $k = \{5, 10, 20, 30, 40, 50, 60, 70, 80, 90\}$. Table 1 shows the results of the experiments. We report the average of the absolute error, the maximum absolute error, and the minimum absolute error for the coefficients a_i for $i = \{1, \ldots, 50\}$. We separately report the results for a_0. The results show that the error is generally increasing with the percentage of missing values. But the amounts of error is negligible for $k < 80\%$, and the error is significant when $k \geq 90\%$.

5.2 Experiments on Missing Attributes

For this set of experiments, we consider the same linear system, however, we randomly remove one of the attributes (coefficients) from the system. We train the model without the parameter and then we predict the missing parameter from the set of observations considering the linear model (the proposed algorithm in Section 3). We compare the computation times of the proposed algorithm to the brute-force

Table 1 The average, maximum, and minimum absolute error of the coefficients a_i as the percentage of missing values are increasing.

Missing Percentage (k)	a_0	$a_{1\le i\le 50}$ Average	Maximum	Minimum
5%	0.02	0.00	0.01	0.00
10%	0.06	0.00	0.01	0.00
20%	0.20	0.01	0.03	0.00
30%	0.07	0.01	0.03	0.00
40%	0.00	0.01	0.04	0.00
50%	0.05	0.02	0.04	0.00
60%	0.09	0.02	0.07	0.00
70%	0.20	0.03	0.08	0.01
80%	0.12	0.04	0.14	0.01
90%	0.17	0.13	0.38	0.01

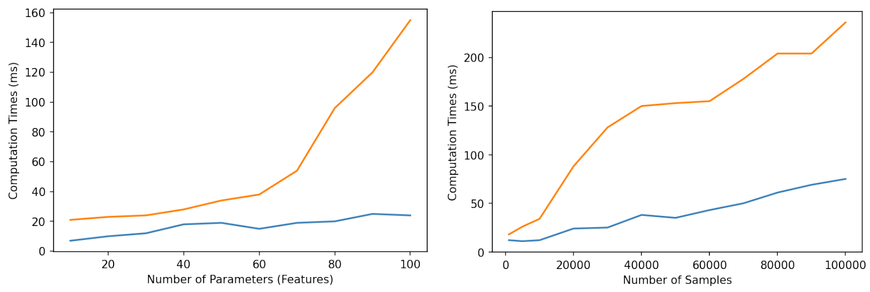

Fig. 1 The computation times (ms) for the proposed algorithm (blue curve) vs. the brute-force approach (orange curve). (a) The computation times as the number of parameters are increasing (the number of samples sets to 10,000). (b) The computation times as the number of samples are increasing (the number of parameters set to 50).

approach when we train the entire model from scratch. In doing so, we vary the number of parameters from 10 to 100 with the number of observations fixed to $10,000$. We repeat the experiment where we vary the number of observations from $1,000$ to $100,000$ with the number of parameters fixed to 50. Figure 1 shows the computation times for these two sets of experiments. The results show that our proposed algorithm (blue curve) is significantly faster than the brute-force approach (orange curve). The computation times of proposed algorithm are increasing with the number of parameters and the number of observations linearly at the worst-case. On the other hand, the brute-force approach has exponential growth in the computation times.

6 Acknowledgments

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in

Computer Science), and HRD-1834620 and HRD-2034030 (CAHSI Includes), and by the AT&T Fellowship in Information Technology.

It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

References

1. M. Al-Zoubi, C. Chang, S. Nazarian, and V. Kreinovich, “A systematic statistical approach to populate missing performance data in pavement management systems", *Journal of Infrastructure Systems*, 2015, Vol. 21, No. 4, paper 04015002.
2. D. Bernstein, *Matrix Mathematics*, Princeton University Press, Princeton, New Jersey, 2005.
3. Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, Massachusetts, 2022.
4. R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, 2005.
5. R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, Wiley, Hoboken, New Jersey, 2020.
6. G. Molenberghs, G. Fitzmaurice, M. G. Kenward, A. Tsiatis, and G. Verbeke, *Handbook of Missing Data Methodology*, Chapman & Hall/CRC, Boca Raton, Florida, 2014.
7. D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.
8. K. S. Thorne and R. D. Blandford, *Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics*, Princeton University Press, Princeton, New Jersey, 2017.
9. S. van Buuren, *Flexible Imputation of Missing Data*, Chapman & Hall/CRC, Boca Raton, Florida, 2018.