Linear Regression under Partial Information

Tho M. Nguyen[2], Saeid Tizpaz-Niari[2], and Vladik Kreinovich[2]

[1]Faculty of Banking and Finance, Ho-Chi-Minh City Open University
Ho-Chi-Minh City, Vietnam, tho.nm@ou.edu.vn

[2]Department of Computer Science, University of Texas at El Paso
500 W. University, El Paso, Texas 79968, USA, saeid@utep.edu, vladik@utep.edu

1. Formulation of the Main Problem

- The main objective of this talk 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.

2. 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 all relevant quantities $x_1, \ldots, x_n$.
- 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.

3. Why linear regression (cont-d)

- We must determine f 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*.
- 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, 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 $x_i^{(1)}$.

4. Why linear regression (cont-d)

- We can then represent the desired dependence in terms of the differences $\Delta x_i \stackrel{\text{def}}{=} x_i - x_i^{(1)}$, as $y = f\left(x_1^{(1)} + \Delta x_1, \ldots, x_n^{(1)} + \Delta x_n\right)$.
- Since the differences Δx_i are small, terms which are quadratic in Δx_i (or of higher order) can be safely ignored.
- We can expand the above formula in Taylor series in terms of Δx_i and ignore quadratic and higher order terms in this expansion.
- So, we get a linear dependence $y = y_0 + a_1 \cdot \Delta x_1 + \ldots + a_n \cdot \Delta x_n$.
- Here $y_0 \stackrel{\text{def}}{=} f\left(x_1^{(1)}, \ldots, x_n^{(1)}\right)$ and $c_i \stackrel{\text{def}}{=} \dfrac{\partial f}{\partial x_i}_{|x_1 = x_1^{(1)}, \ldots, x_n = x_n^{(1)}}$.
- Substituting the expression $\Delta x_i = x_i - x_i^{(1)}$ into the linear formula, we get $y = a_0 + a_1 \cdot x_1 + \ldots + a_n \cdot x_n$.
- Here, we denoted $a_0 \stackrel{\text{def}}{=} y_0 - a_1 \cdot x_1^{(1)} - \ldots - a_n \cdot x_n^{(1)}$.

5. Why linear regression (cont-d)

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

6. 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, we only have an approximate equality

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

- If we denote the difference between the left- and right-hand sides of this formula by $\varepsilon^{(k)}$, we get:

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

- We do not know the approximation errors

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

- Such unknown values are usually called *random*.

7. Why least squares: first explanation (cont-d)

- Usually, there are many different factors that contribute to the approximation error.
- In this case, according to the Central Limit Theorem, the probability distribution of the approximation error is close to Gaussian (normal).
- Thus, it is reasonable to conclude that the approximation error 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

$$y^{(k)} = a'_0 + a_1 \cdot x_1^{(k)} + \ldots + a_n \cdot x_n^{(k)} + \varepsilon'^{(k)}.$$

- Here, 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.

8. Why least squares: first explanation (cont-d)

- 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

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

- 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*.
- The function $\exp(-x)$ is strictly decreasing.

9. Why least squares: first explanation (cont-d)

- So, maximizing the above expression is equivalent to minimizing the expression under the exponential function: $\sum_{k=1}^{K} \left(\varepsilon^{(k)}\right)^2$.
- Here, $\varepsilon^{(k)}$ is defined by the above formula.
- This idea of minimizing the sum of the squares is known as the *least squares* approach.

10. Why least squares: second explanation

- For each k, we want the left-hand side of the approximate equality 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.
- Vectors are points in K-dimensional space.
- A natural way to describe the distance $d(u, v)$ between them 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 least squares form.

11. Why least squares: second explanation (cont-d)

- For all values $x \geq 0$, x^2 is a strictly increasing function.
- So, minimizing the distance is equivalent to minimizing its square.
- Thus, we also arrive at the least squares approach.

12. How the least squares problem is solved now

- When we plug in the expression for $\varepsilon^{(k)}$ into the least squares formula, we get the expression

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

- To minimize this expression, we differentiating it with respect to all the unknown a_i and equate each of the partial derivatives to 0.
- For convenience, we can then divide both sides of each equation by K.
- So, we get the following system of linear equations:

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

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

13. How the least squares problem is solved now (cont-d)

- Here, we denoted

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

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

- For each i, we can multiply both sides of the equation first by $\overline{x}_i$ and subtract this result from the i-th formula.
- Then we get the following simplified equation not containing a_0:

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

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

$$C(q, q') = \overline{q \cdot q'} - \overline{q} \cdot \overline{q'}.$$

14. How the least squares problem is solved now (cont-d)

- For $q = q'$, this is simply the variance $V(q) = \overline{q^2} - (\overline{q})^2$.
- The resulting equations form a system of n linear equations $Ca = b$ 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 $a = C^{-1}b$, 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 above formula:

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

15. 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.

16. Case of partial information is a challenge (cont-d)

- 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.
- This is often not enough to make statistically significant conclusions about the dependence of the desired output y on the inputs x_i.

17. 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
  - we apply the least squares approach to such filled-in data.
- Missing data techniques work in many practical situations.
- However, they have two issues:
- First, many interpolation techniques used in the missing data approach are heuristic.
- 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.

18. How such situations are handled now (cont-d)

- So, to apply the linear formulas in such cases, we need to perform another interpolation.
- This also leads to undesirable non-uniqueness.
- In this paper, we propose an alternative idea.
- This idea enables us to make linear regression under partial information well-justified and thus, avoid the undesired non-uniqueness.

19. 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.
- This is 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.

20. What we propose: main idea (cont-d)

- Good news is that this 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.

21. Resulting algorithm

- For each case $k = 1, \ldots, K$, 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:

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

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

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

- Then, we can find the covariances $C(x_i, x_j)$ and $C(x_i, y)$ by using the usual formula.

22. Resulting algorithm (cont-d)

- Then, we find the coefficients $a_1, \ldots, a_n$ by solving the linear system $Ax = b$.
- After that, we can estimate the remaining coefficient a_0 by using the usual formula.

23. Second issue: 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 a new linear regression problem:
  - finding the coefficients $a'_0, a'_1, \ldots, a'_m$
  - for which $y \approx a'_0 + a'_1 \cdot x_{i_1} + \ldots + a'_m \cdot x_{i_m}$.
- In principle, we can run the general least-squares-under-partial-information procedure again.
- However, that would be too time-consuming: we would again need to analyze all the data etc.

24. Second issue: discussion (cont-d)

- 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.

25. 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 $C(x_i, y)$.
- We want to find the regression coefficients $a'_1, \ldots, a'_m$ for the case when we only know the values of some inputs $x_{i_1}, \ldots, x_{i_m}$.
- For this purpose, we solve the linear system $C'a' = b'$, 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 $a'_0 = \overline{y} - \sum_{j=1}^{m} a'_i \cdot \overline{x_{i_j}}$.

26. Auxiliary 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.
- The hope is that the new inputs 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:

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

- One way to find the new values a'_i is to repeat the same procedure as before.

27. Auxiliary problem (cont-d)

- The only missing information is:
  - 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?

28. What we propose

- Actually, in this part, 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:

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

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

29. What we propose (cont-d)

- This way, we need n^2 arithmetic operations to compute the new inverse.
- On the other hand:
  - even the asymptotically fastest algorithm for solving systems of linear equations require larger time,
  - namely, time proportional to n^a for $a > 2$.

30. 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
  - $a'_0 = a_0 - a'_{n+1} \cdot \overline{x}_{n+1}$.

31. 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
  - in comparison to the existing least square algorithm from scratch.

32. 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 ($y^{(1)} \dots y^{(10,000)}$ and $x^{(1)} \dots x^{(10,000)}$).
- 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 $k = \{5, 10, 20, 30, 40, 50, 60, 70, 80, 90\}$.
- The following table shows the results of the experiments.

33. Experiments on missing values (cont-d)

- We report:
  - the average of the absolute error,
  - the maximum absolute error, and
  - the minimum absolute error for the coefficients a_i for

$$i \in \{1, \ldots, 50\}.$$

- We separately report the results for a_0.
- The results show that the error is generally increasing with the percentage of missing values.
- However, the amounts of error is negligible for $k < 80\%$, and the error is significant when $k \geq 90\%$.

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	$a_{1\leq i\leq 50}$		
		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

34. 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.
- Then we use our algorithm to predict the missing parameter.
- We compare the computation times of the proposed algorithm to the brute-force 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.
- The figure shows the computation times for these two sets of experiments.

35. Experiments on missing attributes (cont-d)

- The results show that our proposed algorithm (blue curve) is significantly faster than the brute-force approach (orange curve).
- The worst-case computation times of proposed algorithm are linearly increasing:
  - with the number of parameters and
  - with the number of observations.
- On the other hand, the brute-force approach has exponential growth in the computation times.

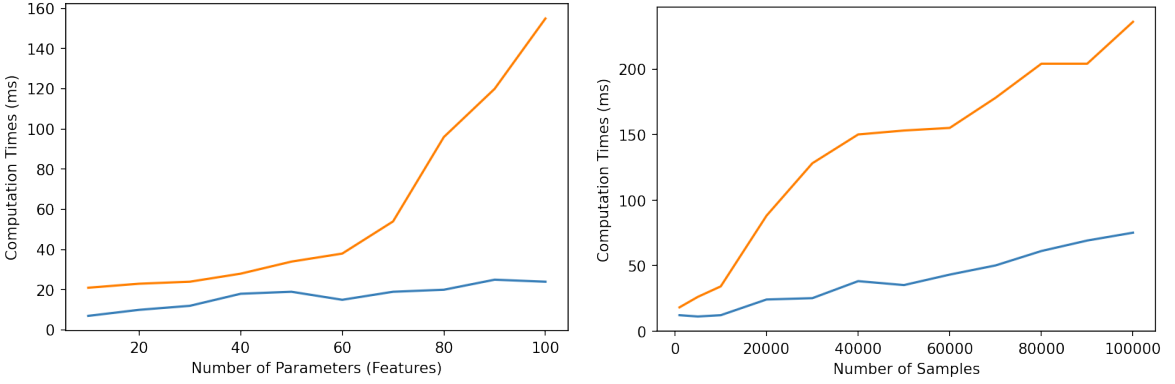

Figure 1: The computation times (ms) for the proposed algorithm (blue curve) vs. the brute-force approach (orange curve). (a) The computation times as the number of parameters are increasing (the number of samples sets to 10,000). (b) The computation times as the number of samples are increasing (the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

36. Acknowledgments

This work was supported in part by:

- National Science Foundation grants 1623190, HRD-1834620, HRD-2034030, and EAR-2225395;
- AT&T Fellowship in Information Technology;
- program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and
- a grant from the Hungarian National Research, Development and Innovation Office (NRDI).