

Statistical Simulation for the Invertibility Test of Binary Random Matrices

Jing Yao

Department of Mathematics, Southern University of Science and Technology, Shenzhen, China

Email address:

yaoj@sustc.edu.cn

To cite this article:

Jing Yao. Statistical Simulation for the Invertibility Test of Binary Random Matrices. *Science Journal of Education*.

Vol. 5, No. 3, 2017, pp. 115-118. doi: 10.11648/j.sjedu.20170503.17

Received: March 31, 2017; **Accepted:** April 18, 2017; **Published:** April 21, 2017

Abstract: One specific mathematical problem is discussed by combining the knowledge of statistical simulation and linear algebra. Aiming to solve this easy-to-understand yet hard-to-answer problem, this paper tries in two ways to test the invertibility of large random binary matrices. By generating random entries of the matrices, and using sparse sampling strategies to get matrices, we also consider programming techniques in order to break the bottleneck of computing power. The proportion of singular matrices changes with the increase of matrix order and the trend is presented. The advantages and disadvantages of the methods are also analyzed from the aspects of result accuracy, time efficiency and applicability. This paper is an example of computer-aided teaching to assist students in enhancing their understanding and practical ability.

Keywords: Matrix Theory, Statistical Simulation, Sampling Strategy, College Mathematics Education

1. Introduction

Linear algebra, as a compulsory course for STEM students, has been focused on studying matrix theory and linear transformation [1-3]. The invertibility of a matrix plays a critical role in the qualitative and quantitative research of matrix theory. Whether a matrix is invertible or not reflects many fundamental properties of the matrix, such as the rank of the matrix, its eigenvalues, the solution of the corresponding linear equations, and so on.

Linear algebra has many applications in engineering, and has also created a subject that specializes in its numerical methods: numerical linear algebra [4]. It combines computing technology with the theoretical basis of linear algebra to solve matrix computation problems and find solutions for large-scale equations. People have realized that computers can help to solve problems in linear algebra since long time ago. One of the major categories is to explore randomness problems by utilizing statistical simulation method. In 1989 Edelman used statistical methods to study the conditional number of a random matrix and discussed the condition number distributions derived for large matrices [5].

This kind of techniques is also widely used in mathematics teaching. For example, Strang gave an example in the preface of his book [6]: "Are random triangles acute or obtuse?" and

comes to the conclusion: In reality, a typical triangle is obtuse.

In this paper, we will focus on a problem of matrix invertibility. This question is answered by considering two different statistical simulation approaches with comparisons.

2. Problem Description and Analysis

The problem is quite simple: If you put 1s and 0s at random into the entries of a n by n matrix, is it more likely to be invertible or singular [7]?

The description of the problem is easy to understand, but it is not always so easy to answer the question, especially for large n .

The object discussed here is a (0-1) random matrix. Such matrix is also called a binary matrix, logical matrix, relation matrix, or Boolean matrix. It is often used to represent a binary relation between a pair of finite sets. Typical examples include a permutation matrix in the elementary transformation of matrices and the adjacency matrix in graph theory. It may have extra characteristics according to the background of its applications.

To determine the invertibility of a matrix, we can proceed from several aspects. For instance, the equivalent condition of the singularity of a matrix includes the following: the

matrix has zero eigenvalue; the determinant of the matrix is zero; the matrix is not full rank. Any one of these conditions is sufficient to judge whether the matrix is invertible or singular.

For a (0-1) matrix, some results are given in the literature: Ryser has presented some combinatorial properties of this kind of matrices [8]. For the discussion of eigenvalues, Weisstein has counted the $n \times n$ matrices of 0's and 1's all of whose eigenvalues are real and positive, and obtained the sequence as 1, 3, 25, 543, 29281 for $n=1, 2, 3, 4, 5$ [9]. This result was subsequently proved by McKay et al [10]. These statements are very useful for the study of integer sequences and also make contributions to acyclic digraphs. However, this cannot fully answer the question here and we need to seek another way. The problem is difficult to solve theoretically, and a more feasible way is to employ computer simulation.

The answer can be easily reached with the matrix order n as 1 or 2. For $n=1$, there is one invertible and one singular matrix. For $n=2$, with the exhaustive method we can see that among the 16 matrices, only 6 matrices are invertible, and the proportion of singular matrices is 0.625. When $n=3$, we can judge the invertibility of each matrix without much difficulty. However, there are $2^9=512$ distinct matrices, and it will be time-consuming to calculate manually. With the further increase of the size of the square matrix, it is almost impossible to determine the invertibility for all without a computer. The program to fulfill the task itself is not complicated, and the difficulty mainly lies in the huge amount of computation, which easily leads to insufficient memory due to hardware limitations, and the restrictions of software can also come to inaccurate results.

3. Invertibility Test and Simulation Strategies

For situations beyond the computer load, we consider two categories of simulation strategies.

3.1. Method I (Random Entry Method: REM)

We generate each entry of the matrix by randomly choosing it to be 0 or 1. For a matrix of order n , there are n^2 entries, each of which is a Bernoulli random variable [11]. The entries take on only two values: 0 and 1, with both probabilities as 0.5. In terms of programming, this is done by generating a uniformly distributed random variable on the [0,1] interval, and then using the rounded approach to get the closest integer, which is either 0 or 1. The invertibility of the matrix is then judged according to the determinant or the rank of the matrix. The operation is repeated many times (10^4 or 10^5 times) to record the proportion of the singular matrices. In order to improve efficiency, we use matrix operation to avoid loop statement when using Matlab® software.

3.2. Method II (Sparse Sampling Strategy: SSS)

Another way is to apply sparse sampling strategy to

generate a portion of all possible matrices. The key is to ensure that not only the matrix set is complete, but also the strategy should be adequate to maintain randomness. In other words, the strategy needs to make sure that all the possible matrices have chance to be selected, and can be selected with equal probability. To meet these requirements, we have tried the following several approaches for the simulation.

3.2.1. A Convenient Way to Generate a Random Binary Matrix

In fact, the generation of all the matrices to meet the requirements needs some techniques to enhance the efficiency of the algorithm. It is necessary to achieve the completeness of all the matrices by a handy approach in the programming. The following technique is used to get all the matrices.

There are totally 2^{n^2} random matrices for a specific matrix order n .

Step 1 Generate integers from 0 to $2^{n^2} - 1$;

Step 2 Convert each of the above numbers into a binary sequence;

Step 3 Reshape each array of the binary sequences into a $n \times n$ square matrix.

3.2.2. Sparse Sampling Strategies

Sampling strategies are critical to ensure the efficiency and accuracy of the final results. The strategies are summarized as follows:

A. Sampling is not performed and all possible matrices are involved in the calculation when the computer is not overloaded (i.e., out of memory).

B. The sampling upper limit is set to be 10^5 . Sampling strategies are adopted when the number of all matrices for a certain n exceeds the upper limit.

Strategy I After randomly re-ordering of the matrices, the first 10^5 matrices are selected to be involved in the statistics;

Strategy II (Improvement to strategy I) Totally 10^5 unrepeated numbers are chosen randomly from 0 to $2^{n^2} - 1$, and corresponding matrices are tested;

Strategy III Linear sampling scheme are carried out from 0 to $2^{n^2} - 1$ with equal sampling interval. The starting points of each interval are picked and their related matrices are used in the invertibility test;

Strategy IV (improvement to strategy III) Some small increments are added to the 10^5 numbers derived from Strategy III. The disturbance is a random number chosen from 0 to the length of the interval (Figure 1). The corresponding matrices are then calculated and tested.

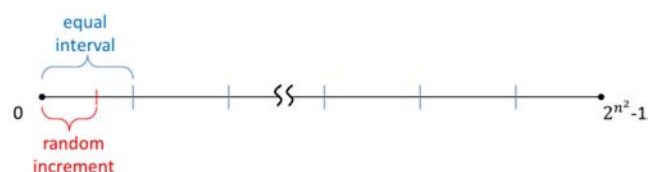


Figure 1. The scheme plot of sparse sampling strategy IV.

4. Simulation Results and Discussions

After generating the matrices according to the steps above, the determinant or rank of each matrix is computed. If the matrix is not full rank, or equivalently, with zero determinant, then it is labeled as a singular matrix; otherwise, it is an invertible matrix. Both the determinant and the rank are considered in the simulation. The two criteria lead to the same result, and do not differ much in efficiency.

We run our simulation program under the environment as follows. (i) Software: Matlab® R2016b; (ii) Hardware: Lenovo X1 Carbon laptop. The basic computer configuration is: Processor: Intel® Core™ i5-5200U CPU@2.20GHz, memory: 8G.

The computing capacity is limited if all the binary random

matrices are tested. The program announces “out of memory” and breaks when the order of matrices is beyond 4. The order that can be tested has been raised while using random entry method (REM) and is basically unrestricted when the order is increased. We tested n from 1 all the way to 32, and part of the results is presented in Table 1.

The number of possible matrices increases dramatically as the matrix order grows. As can be seen from Table 1, by using REM we can get the results very close to those by the exact calculation, which can only be compared at $n=1, 2, 3, 4$. In REM, the elapsed time grows steadily with the increase of matrix order, and the number of matrices generated has a greater impact on the consumption of calculation time.

Table 1. Statistics for the singular proportion of random binary matrices (Method: REM).

Matrix order n	Number of possible matrices	Exact proportion of singularity (by exhaustive method)	Number of generated matrices (REM): 10^4		Number of generated matrices (REM): 10^5	
			Elapsed time	Proportion of singularity	Elapsed time	Proportion of singularity
1	$2^1=2$	0.5000	0.1424	0.5030	7.2601	0.5008
2	$2^4=16$	0.6250	0.1696	0.6283	7.4962	0.6276
3	$2^9=512$	0.6602	0.1708	0.6651	7.7553	0.6581
4	$2^{16}=65536$	0.6558	0.1884	0.6549	7.9057	0.6573
5	$2^{25}=33554432$	/	0.2151	0.6366	8.1266	0.6290
6	$2^{36} \approx 6.8719 \times 10^{10}$	/	0.2317	0.5852	8.3493	0.5802
7	$2^{49} \approx 5.6295 \times 10^{14}$	/	0.2625	0.5260	8.5935	0.5195
8	$2^{64} \approx 1.8447 \times 10^{19}$	/	0.2783	0.4473	8.8317	0.4479
9	$2^{81} \approx 2.4179 \times 10^{24}$	/	0.3092	0.3750	9.2047	0.3733
10	$2^{100} \approx 1.2677 \times 10^{30}$	/	0.3413	0.2931	9.4388	0.2971
11	$2^{121} \approx 2.6585 \times 10^{36}$	/	0.3749	0.2277	9.8282	0.2242
12	$2^{144} \approx 2.2301 \times 10^{43}$	/	0.4012	0.1651	10.0864	0.1648

Figure 2 shows the ratio of the matrix singularity when the order of matrices increases from 1 to 32 (the number of simulated matrices is 10^5). We can clearly see that the matrix singular proportion reaches maximum at 0.66 when $n=3$; the proportion declines as $n>3$, and when $n \geq 18$, the singular matrix ratio is less than 0.01, that is, *almost* all of the matrices are invertible.

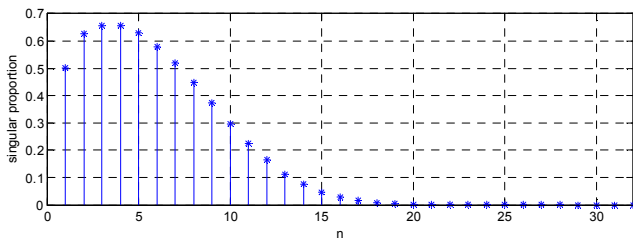


Figure 2. The singular proportion trend by REM.

If the sparse sampling strategies (SSS) are applied in the simulation, we can partially avoid the defect of memory shortage. The use of the four sampling strategies above can run the program up to the matrix order of 5, 7, 31 and 31. The calculation capacity is due to the limitation of Matlab built-in functions, and we discover that the results become unstable when $n>7$.

The proportion results by Sparse Sampling Strategy II and

IV are listed in Table 2. The number of the matrices sampled here is set to be $\min(2^{n^2}, 10^5)$.

Table 2. Statistics for the singular proportion of random binary matrices (Method: SSS).

Matrix order n	Strategy II		Strategy IV	
	Elapsed time	Proportion of singularity	Elapsed time	Proportion of singularity
1	0.0144	0.5000	0.0184	0.5000
2	0.0152	0.6250	0.0183	0.6250
3	0.1745	0.6602	0.1442	0.6602
4	23.6128	0.6558	23.0028	0.6558
5	54.8391	0.6266	54.4648	0.6418
6	75.0965	0.5803	74.9379	0.5874
7	100.6473	0.5162	100.1801	0.5206

It is obvious that for $n=1, 2, 3, 4$, the results are exactly the same as the ones using the exhaustive method, since the total number of matrices does not reach the upper limit. All the matrices are taken into account, thus also validates the correctness of the algorithm. The running of the SSS program takes much more time as the matrix order grows bigger. When n is between 5 and 7, the results obtained by SSS are compared with REM in Figure 3.

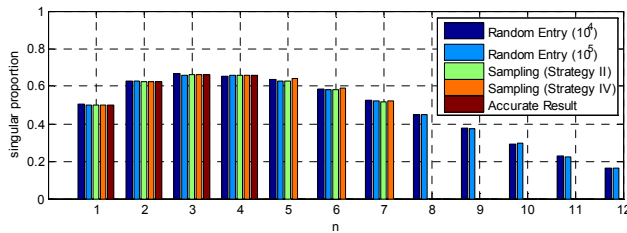


Figure 3. Comparison of matrix singular proportions of different methods.

Figure 3 illustrates that the results of the methods are consistent. The tiny differences can be understood because statistical methods are introduced. The two methods presented here have their own advantages and disadvantages, some of which are listed below.

(1) *Accuracy*: there is not much difference between the accuracy. SSS performs better when $n \leq 4$, and REM produces small amount error compared to the exact result;

(2) *Time consumption*: SSS slows down immediately when $n \geq 4$, mainly because the generation of binary numbers accounts for more time; Meanwhile, the order of matrix has insignificant and acceptable influence on the elapsed time of REM;

(3) *Applicability*: SSS can only give stable results up to $n=7$, while REM is able to be applied to larger matrix orders, which can present the trend of the singularity.

5. Conclusions

The methods of statistical simulation are used to determine the invertibility of a large number of large scale random binary matrices. The random entry generation method, and the various sparse sampling strategies, together with programming techniques are applied to estimate the singular proportion of matrices. The ratio of singularity with its trend is obtained as the matrix order increases.

There may be some improvement of the testing criterion. We used determinant and rank to judge whether the matrix is invertible. Some other method, such as whether the matrix has zero eigenvalue, can be tried to test the efficiency of the program. In addition, with the upgrade of computer hardware performance, it can also be expected to break the limit of the matrix order.

This problem is a moderate level example and can lead students to think about matrix invertibility testing, programming skills, and the relationship between theoretical analysis and actual calculation. Developing this kind of

exercises as a supplement of classroom teaching is helpful for improving students' awareness of the nature of problems and enhancing their practical ability.

Acknowledgements

This research is funded by SUSTech Teaching Reform Project ("Comparative Research and Practice on College Mathematics Teaching in Information Environment", Grant No. Y01051816).

References

- [1] Sullivan, P.; Clarke D.; Clarke B. "Perspectives on mathematics, learning, and teaching", Mathematics Teacher Education, 2013, pp. 7-12
- [2] Baturo, A.; Cooper, T; Doyle, K; Grant, E. "Using three levels in design of effective teacher-education tasks: The case of promoting conflicts with intuitive understandings in probability." Journal of Mathematics Teacher Education, 2007, 10(4), pp. 251-259
- [3] Day, J. M.; Kalman, D. "Teaching linear algebra: issues and resources." The College Mathematics Journal, 2001, 32(3), pp. 162-168
- [4] Ford W. Numerical Linear algebra with applications using MATLAB. 2015, pp. 79-101
- [5] Edelman, A. Eigenvalues and condition numbers of random matrices. MIT Dissertation, 1989, 106 pages
- [6] Strang, G. Essays in linear algebra. Wellesley-Cambridge Press, 2012, pp. iv-vi
- [7] Strang, G. Linear algebra and its applications. 4ed (international student edition), Brooks/Cole, Cengage Learning, 2006, pp. 65
- [8] Ryser, H. J. "Combinatorial properties of matrices of zeros and ones." Canad. J. Math. 1957(9), pp. 371-377
- [9] Weisstein, E. W. "Weisstein's conjecture." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/WeissteinsConjecture.html>
- [10] McKay, B. D.; Royle, G. F.; Wanless, I. M.; Oggier, F. E.; Sloane, N. J. A.; and Wilf, H. "Acyclic digraphs and eigenvalues of (0,1)-matrices." J. Integer Sequences, 2004(7), Article 04.3.3, pp. 1-5
- [11] Rice, J. A. Mathematical Statistics and Data Analysis. 3ed, Thomson, 2007, pp. 37-47