The Theoretical and Experimental Analysis of the Maximal Information Coefficient Approximate Algorithm

Fubo SHAO

State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China; The Post-Doctor Station of the CRRC Corporation Limited, Beijing 100067, China; CRRC Academy, Beijing 100067, China
E-mail: shaofubo@163.com

Hui LIU*

School of Mathematics and Physics, Qingdao University of Science and Technology, Qingdao 266061, China
E-mail: bhjx2009@163.com

Abstract In the era of big data, correlation analysis is significant because it can quickly detect the correlation between factors. And then, it has been received much attention. Due to the good properties of generality and equitability of the maximal information coefficient (MIC), MIC is a hotspot in the research of correlation analysis. However, if the original approximate algorithm of MIC is directly applied into mining correlations in big data, the computation time is very long. Then the theoretical time complexity of the original approximate algorithm is analyzed in depth and the time complexity is $n^{2.4}$ when parameters are default. And the experiments show that the large number of candidate partitions of random relationships results in long computation time. The analysis is a good preparation for the next step work of designing new fast algorithms.

Keywords statistical correlation; the maximal information coefficient; approximate algorithm; mutual information; dynamic programming algorithm

1 Introductions

With the development of information technology and its widely applications in various industries, the total data volume explosively grows with the increasing by about 50% annually. Data fusion of different identities results in high dimensional massive data. And then, the correlation analysis of factors is very important.

The object of the correlation analysis is to identify correlations between stochastic variables. The earliest research on statistical correlation can be traced back to the statistical correlation theory between two variables proposed by British statistician Galton^[1]. After that, the first statistical coefficient, Pearson coefficient^[2], is proposed by Pearson, the student of Galton. The

Received September 5, 2019, accepted December 11, 2019

Supported by the China Postdoctoral Science Foundation (2019M650981); Shandong Provincial Natural Science Foundation, China (ZR2018MG003)

*Corresponding author

value of Pearson coefficient is $[-1, 1]$. The positive value expresses positive correlation and the negative value expresses negative correlation. However, Pearson coefficient can only be used to identify linear relationships.

Spearman coefficient^[3], which is the generalization of Pearson coefficient, is proposed by Spearman through theoretical deduction, and it is also called sequential correlation coefficient or rank correlation coefficient. This coefficient is obtained via analyzing the linear correlation of the rank of observations and its value is also $[-1, 1]$.

Kendall rank coefficient^[4] measures the correlation of grade variables via the consistency of bivariate sequence pairs. And this coefficient value is $[-1, 1]$. However, Kendall coefficient can only measure the correlation between grade variables.

Mutual information^[5-8], which is proposed by Shannon in 1948, is an excellent coefficient measuring the correlation between two variables. However, calculating mutual information is difficult because of the need of estimating its probability density function. The coefficients based on curve principle^[9, 10] obtain the principal component representation of features via decomposing the variance and covariance matrices into spectral components. There are two coefficients, including covariance along a generating curve (CovGc) and correlation along a generating curve (CorGc). These coefficients can well identify linear correlations, independence of variables and a part of non-linear relationships. When the correlation between two variables is linear, the CorGc coefficient is approximately equal to Pearson coefficient.

Distance correlation coefficient^[11, 12] measures the correlation between two variables via calculating the Euclidean distance of samples rather than the moment distance. Overcoming the weakness of Pearson coefficient, distance correlation coefficient can measure not only linear correlations but also non-linear correlations.

The above correlation coefficients have their own advantages, limitations and specific scope of applications. The maximal information coefficient (MIC)^[13], which is proposed in 2011, has two excellent properties: Generality and equitability. That is MIC can detect various relationships including linear, non-linear, functional and non-functional relationships. And the MIC values of the different type relationships are similar at the same noise level.

However, if MIC is directly applied into mining bivariate correlations in big data by employing the original approximate algorithm, the computation time is very long. Then the cause of the long computation time is analyzed from the theoretical and experimental aspects in this paper. In summary, the contributions of this paper are as follows.

Firstly, the definition of MIC is systemically introduced and the approximate algorithm calculating MIC is profoundly analyzed.

Secondly, the approximate algorithm is divided into four sub-algorithms and the computation time of different type relationships is compared by the means of experiments. The cause of long computation time is found. Compared to other relationship types, the candidate partitions of random relationships is larger and then the time of calculating MIC values of random relationships is longer. In high dimensional big data, the major relationships are random, and then the computation time is very long.

Thirdly, besides the experimental analysis, the theoretical time complexity is analyzed. If parameters are default, the time complexity of the original approximate algorithm is $O(n^{2.4})$.

The paper is organized as follows. Firstly, in Section 2, the definition and the original approximate algorithm are introduced. Then the original approximate algorithm is analyzed in Section 3. At last, conclusions are drawn in Section 4.

2 The Definition and Approximate Algorithm of MIC

Employing the excellent computing ability of modern computers, the scatter plots of two variables are divided into grids and the ratio of data points falling into grids is regarded as the approximate joint probability distribution and marginal probability distribution. Calculate and normalize the mutual information of two variables, and then obtain the MIC value of two variables.

2.1 The Definition of MIC

Given two variables X, Y , the scale of the data set $D = \{(x_i, y_i), i = 1, 2, \dots, n\}$ is n , that is $|D| = n$. The X, Y axes are divided into x, y grids, respectively. And then an $x \times y$ grid G is obtained. The ratio of data points falling into grids is the approximate probability distribution $D|_G$. Different grids will result in different probability distribution functions and then different mutual information values are obtained. For the grids with the same numbers of partitions, maximize and normalize the mutual information. The maximal element value of the matrix is the maximal information coefficient (MIC). The specific definition of MIC is as follows.

Definition 1 The data set is $D \subset R^2$ and the grid is $x \times y$ where x, y are positive integers. The probability distribution is $D|_G$ and then the maximum mutual information is as Formula (1).

$$I^*(D, x, y) = \max I(D|_G). \quad (1)$$

In the formula (1), $I(D|_G)$ is the mutual information of the data set D under the grid G and the maximal value in formula (1) is the maximal mutual information on all partitions where the two axes are divided into x, y partitions, respectively.

Definition 2 The entities of the characteristic matrix $M(D)$ of the data set D is defined as formula (2).

$$M(D)_{x,y} = \frac{I^*(D, x, y)}{\log \min\{x, y\}}. \quad (2)$$

The entity of the x th row and the y th column in the matrix $M(D)$ is the maximal mutual information of all $x \times y$ grids divided by the minimum value of x, y . That is the entity is the maximal mutual information normalized by the minimum value of x, y . After the normalization step, all entities of the matrix $M(D)$ falls into the domain $[0, 1]$ and the characteristic matrix is symmetric.

Definition 3 The MIC value of the data set D ($|D| = n$) with two variables is defined as Formula (3).

$$\text{MIC}(D) = \max_{x, y \leq B(n)} \{M(D)_{x,y}\}. \quad (3)$$

In Formula (3), the parameter $B(n) = n^{1-\epsilon}$, $\epsilon \in (0, 1)$.

Taking the computing efficiency into consideration, in order to reduce the calculation workload, only a part, not all, of grids is calculated in the search procedure of calculating MIC. According to the practice experience^[13], the default value $B(n) = n^{0.6}$.

The MIC value of two variables is also falling into the domain $[0, 1]$, because all entities of the characteristic matrix fall into the domain $[0, 1]$. With the character matrix $M(D)$, besides MIC, other criteria MINE is also summarized in [13].

2.2 The Original Approximate Algorithm Calculating MIC

According to Definitions 1 $\sim$ 3, the main work of the maximal information coefficient (MIC) is calculating the characteristic matrix $M(D)$ of the data set. That is, based on the fixed number of partitions, maximize the normalized mutual information. In order to calculate the MIC of two variables, reference [13] gives the approximate algorithm calculating MIC of two variables.

In order to improve computing efficiency, besides the definition of MIC, an approximate algorithm calculating MIC of two variables is proposed^[13]. In the approximate algorithm, firstly, one axis is equally partitioned. And then, aiming at maximizing mutual information, the partition of the other axis is obtained via employing the dynamic programming algorithm. The flow chart of the approximate algorithm is shown in Figure 1. The order of the two variable in the data set D is interchanged and then the data set $D^\perp$ is obtained. After that, the maximal mutual information of data sets $D, D^\perp$ of the given numbers of partitions is calculated by the sub algorithm $\text{ApproxMaxMI}(D, x, y, \tilde{k})$. Specifically, in Figure 1, given the numbers of partitions of grids, that is the numbers x, y and $\log \min\{x, y\}$ are given, the work of calculating the maximal mutual information of the given number of partitions is accomplished by the sub-algorithm $\text{ApproxMaxMI}(D, x, y, \tilde{k})$.

Figure 1 The approximate algorithm calculating the MIC of two variables in reference [13]

In the sub-algorithm $\text{ApproxMaxMI}(D, x, y, \tilde{k})$, one axis is equally divided into x partitions and the other axis is equally divided into cy partitions. Aiming at maximizing the mutual information, the cy partitions are fused into y partitions by employing the dynamic programming algorithm, and this is the main work of the sub-algorithm $\text{OptimizeAxis}(D, Q, x)$. The flow chart of the sub-algorithm $\text{OptimizeAxis}(D, Q, x)$ is shown in Figure 2.

Figure 2 The flow chart of the sub algorithm $\text{ApproxMaxMI}(D, x, y, k)$

3 The Original Approximate Algorithm Analysis

In the procedure of applying the approximate algorithm proposed in [13] to detecting bi-variate correlations in high dimensional big data, the computation time is very long. Then, in this paper, the approximate algorithm is thoroughly analyzed and the improvement direction of the MIC algorithm is given.

Firstly, from the theoretical aspect, the time complexity of the approximate algorithm is analyzed. Secondly, from the experimental aspect, the computation time is analyzed via analyzing nine type relationships similar to [13]. Among the nine different type relationships of this paper, the computation time of random relationship is longest and then the computation time of mining correlations in big data is very long because the majority bivariate relationships are random.

3.1 The Theoretical Time Complexity Analysis

According to Figures 1 and 2, the approximate algorithm proposed in [13] mainly applies the sub algorithm $\text{ApproxMaxM}(D, x, y, cx)$ to calculating the maximal mutual information under the given x, y partitions. Specially, there are four sub-algorithms including $\text{Equipartition}(D, y)$, $\text{GetClumpsPartition}(D, Q)$, $\text{GetSuperClumpsPartition}(D, Q, \tilde{k})$ and $\text{OptimizeXAxis}(D, Q, x)$.

The sub-algorithm Equipartition(D, y) equally partitions the data set which has been sorted by the fast sorting algorithm. Then the time complexity of the sub algorithm is equal to that of the fast sorting algorithm, that is $O(n \log_2 n)$.

The sub-algorithm GetClumpsPartition(D, Q) sorts the data set via employing the classical fast sorting algorithm and then traverses the sorted data set. Then the time complexity of the sub-algorithm GetClumpsPartition(D, Q) is also equal to that of the fast sorting algorithm, that is $O(n \log_2 n)$.

The sub algorithm GetSuperClumpsPartition($D, Q, \tilde{k}$) fuses the $\tilde{k} = cx$ partitions into x partitions, where default $c = 15$. Then the sub algorithm only traverses the $\tilde{k}$ partitions. In other words, the data set is only traversed one time. Then the time complexity of the sub algorithm GetSuperClumpsPartition($D, Q, \tilde{k}$) is $O(n)$.

The time complexity of the sub algorithm OptimizeXAxis(D, Q, x) is $O(k^2xy) = O(c^2x^3y) = O(x^2B)$, where $k = cx$, $xy \leq B = n^\alpha$. Then, if the number of the X partitions is determined, that is X is given, the time complexity of the sub algorithm is $O(k^2xy) = O(c^2x^3y) = O(x^2B)$.

According to the original approximate algorithm proposed in [13], the range of x is $[2, B/2]$. And then the time complexity of whole approximate algorithm proposed in [13] is $O(x^2B)B = O(x^2B^2) = O(B^4) = O(n^{4\alpha})$. If the parameter is the default value $\alpha = 0.6$, the time complexity of the whole algorithm is $n^{2.4}$, which is relatively high.

3.2 Experimental Analysis

In this paper, the software, developed by Albanese, et al.^[14], for calculating MIC values of two variables is the implementation of the approximate algorithm in [13]. The configuration of the computer used for computation is as follows. Win 7 operation system; CPU: Intel(R) Core(TM) i5-2450, 2.50 GHz; RAM: 4.00 GB. And similar to [13], 9 different type relationships are also adopted in this paper and the specific description is shown in Table 1.

Table 1 Bivariate relationships

No.	Name	Description
1	Linear	$y = x$
2	Parabolic	$y = 4(x - 0.5)^2$
3	Cubic	$y = 128(x - \frac{1}{3})^3 - 48(x - \frac{1}{3})^2 - 12(x - \frac{1}{3}) + 2$
4	Exponential	$y = 2^x$
5	Linear/Periodic	$y = \sin(10\pi x) + x$
6	Sinusoidal (Fourier Frequency)	$y = \sin(16\pi x)$
7	Sinusoidal (non-Fourier Frequency)	$y = \sin(13\pi x)$
8	Sinusoidal (Varying Frequency)	$y = \sin(7\pi x(1 + x))$
9	Random	random number generator

According to the 9 different type bivariate relationships in Table 1, generate the data sets with different scales, and then calculate the MIC values by employing the software of [14] which is the implementation of the original approximate algorithm proposed in [13]. The calculation time is shown in Figure 3.

Figure 3 The calculating time of the bivariate relationships of different types

From Figure 3, the following results can be obtained. For the data sets of the relationships with the same scale, the time of calculating MIC values of random relationships is longest, and with the increasing of the scale, the calculating time of random relationship increases rapidly. In contrast, the computation time of linear relationship is shortest and its growth rate is slowest.

To further study the causes of the varying of the computation time, the computation time of all sub-algorithms and the whole approximate algorithm is compared. And it is found that the majority computation time of the whole algorithm is that of the sub-algorithm $\text{OptimizeXAxis}(D, Q, x)$. The comparison of computation time is shown in Figure 4. The parameters are $n = 300$, $\alpha = 0.6$, $c = 15$.

Figure 4 The comparison of the computation time of the sub algorithm $\text{OptimizeXAxis}(D, Q, x)$ and the whole algorithm

From Figure 4, the following results can be obtained. The computation time of the random relationship is longer than that of the other type relationships and for every type relationship, the most of the computation time is spent on finding the optimal partitions (maximizing the mutual information under the fixed number of partitions). The results from the experimental aspect are consistent with that from the theoretical aspect. In the theoretical analysis, the time complex of the sub-algorithm $\text{OptimizeXAxis}(D, Q, x)$ is $O(k^2xy)$ which is relatively higher. In order to study the cause of the varying of different type relationships, the sub-algorithm

OptimizeXAxis(D, Q, x) is further analyzed.

In the sub-algorithm OptimizeXAxis(D, Q, x), the number of candidate partitions of different type relationships in the procedure of maximizing mutual information is shown in Table 2. Due to the parameters $n = 300$, $\alpha = 0.6$, then the lower integer of n^α is 30. That is the maximal value of the product of the partitions of two axes is 30. Then the product of the first and second column is not larger than 30. The values of the other columns are the number of candidate partitions of different type relationships. Due to the parameter c is equal to 15, the number of candidate partitions is not larger than the product of the value of the second column and the parameter c ($c = 15$). From Table 2, it can be obtained that the number of partitions of linear relationships is least and the number of the random relationship, which is close to the upper bound of partitions (the product of the parameter c and the number of partitions in the second column), is largest. Due to the difference of partitions, the computation time varies from different type relationships. Among these different type relationships in Table 1, the computation time of linear relationship is shortest and the computation time of random relationship is longest in Figures 3, 4.

Because there are many random bivariate relationships in high dimensional big data and the time of calculating MIC values of random relationships is long, the computation time of detecting bivariate correlations in big data is very long via employing MIC.

Table 2 The comparison of the number of candidate partitions of different type bivariate relationships

$Y(X)$	$X(Y)$	Linear	Parabolic	Cubic	Exponential	Linear/Periodic
2	15	2(2)	3(140)	4(68)	2(2)	12(100)
3	10	3(3)	5(130)	7(131)	3(3)	22(129)
4	7	4(4)	7(104)	10(93)	4(4)	30(100)
5	6	5(5)	9(90)	13(70)	5(5)	39(87)
6	5	6(6)	11(75)	16(58)	6(6)	47(73)
7	4	7(7)	13(60)	19(54)	7(7)	53(60)
8	3	8(8)	15(45)	22(41)	8(8)	45(45)
9	3	9(9)	17(45)	25(41)	9(9)	45(44)
10	3	10(10)	19(45)	26(41)	10(10)	45(44)
11	2	11(11)	21(30)	29(29)	11(11)	30(30)
12	2	12(12)	23(30)	27(28)	12(12)	30(30)
13	2	13(13)	25(30)	28(29)	13(13)	30(30)
14	2	14(14)	27(30)	28(28)	14(14)	30(30)
15	2	15(15)	29(30)	29(28)	15(15)	30(30)

Table 2 (continued)

$Y(X)$	$X(Y)$	Sinusoidal (Fourier Frequency)	Sinusoidal (non-Fourier Frequency)	Sinusoidal (Varying Frequency)	Random
2	15	16(154)	14(164)	15(171)	142(153)
3	10	33(149)	27(150)	29(149)	145(150)
4	7	48(105)	40(105)	42(105)	105(105)
5	6	64(90)	53(90)	57(90)	90(90)
6	5	70(75)	65(75)	70(75)	75(75)
7	4	60(60)	60(60)	60(60)	60(607)
8	3	45(45)	45(45)	45(45)	45(45)
9	3	45(45)	45(45)	45(45)	45(45)
10	3	45(45)	45(45)	45(45)	45(45)
11	2	30(30)	30(30)	30(30)	30(30)
12	2	30(30)	30(30)	30(30)	30(30)
13	2	30(30)	30(30)	30(30)	30(30)
14	2	30(30)	30(30)	30(30)	30(30)
15	2	30(30)	30(30)	30(30)	30(30)

4 Conclusion

MIC (the maximal information coefficient) has two excellent properties: Generality and equitability. However, if the original approximate algorithm of MIC is directly applied into detecting bivariate correlations in high dimensional big data, the computation time is very long. Aiming at this problem, this paper profoundly analyzes the original approximate algorithm calculating MIC proposed in Science in 2011. The definition of MIC and the approximate algorithm is introduced and the problem of long computation time is analyzed from two aspects, theoretical time complex analysis and experimental analysis. And it is found that the time of calculating MIC values of random relationships is longest due to the largest number of candidate partitions. However, in the high dimensional big data, the majority bivariate relationships are random. Then the computation time is very long when mining bivariate correlations in big data.

After finding the cause of long computation time, the next step work is to design the fast algorithm for calculating MIC of bivariate relationships in big data. And the next step work can be developed in two ways. Firstly, design the very fast algorithm which can only detect random relationships aiming at the majority random relationships in big data. Secondly, aiming at the cause of long computation time (large number of candidate partitions), design the efficient fast algorithm calculating MIC of bivariate relationships.

References

- [1] Galton F. Regression towards mediocrity in hereditary stature. *Journal of the Anthropological Institute*, 1885, 15: 246–263.
- [2] Pearson K. Mathematical contributions to the theory of evolution (III): Regression, heredity, and panmixia.

- Philosophical Transactions of the Royal Society of London, Series A, Containing Papers of a Mathematical or Physical Character, 1895, 187: 253–318.
- [3] Spearman C. The proof and measurement of association between two things. *The American Journal of Psychology*, 1904, 15(1): 72–101.
 - [4] Kendall M. A new measure of rank correlation. *Biometrika*, 1938, 30(1): 81–93.
 - [5] Shannon C. A mathematical theory of communication. *ACM Sigmobile: Mobile Computing and Communications Review*, 2001, 5(1): 3–55.
 - [6] Moon Y, Rajagopalan B, Lall U. Estimation of mutual information using kernel density estimators. *Physical Review E*, 1995, 52(3): 2318–2321.
 - [7] Ciganovic N, Beaudry N, Renner R. Smooth max-information as one-shot generalization for mutual information. *IEEE Transactions on Information Theory*, 2014, 60(3): 1573–1581.
 - [8] Cover T, Thomas J. *Elements of Information Theory*. John Wiley & Sons, Hoboken, 2012.
 - [9] Delicado P, Smrekar M. Measuring non-linear dependence for two random variables distributed along a curve. *Statistics and Computing*, 2009, 19: 255–269.
 - [10] Hastie T, Stuetzle W. Principal curves. *Journal of the American Statistical Association*, 1989, 84(406): 502–516.
 - [11] Székely G, Rizzo M, Bakirov N. Brownian distance covariance. *The Annals of Applied Statistics*, 2009, 3(4): 1236–1265.
 - [12] Székely G, Rizzo M, Bakirov N. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 2007, 35(6): 2769–2794.
 - [13] Reshef D, Reshef Y, Finucane H, et al. Detecting novel associations in large data sets. *Science*, 2011, 334(6062): 1518–1524.
 - [14] Albanese D, Filosi M, Visintainer R, et al. Minerva and minepy: A C engine for the MINE suite and its R, Python and MATLAB wrappers. *Bioinformatics*, 2013, 29(3): 407–408.