Simulation on the “Efficient MCMC Algorithm to Sample Binary Matrices with Fixed Marginals”

May 10, 2016

**Summary**

The aim of the final project is to simulate the uniform sampling of binary matrices (0-1 matrices) with fixed margins (fixed row sum and column sum) using an Markov chain Monte Carlo (MCMC) algorithm as discussed in the literature of “AN EFFICIENT MCMC ALGORITHM TO SAMPLE BINARY MATRICES WITH FIXED MARGINALS" by Norman D. Verhelst from CITO, National Institute for Educational Measurement in 2008 (Verhelst, 2008). This new algorithm, using importance sampling with Metropolis-Hasting, had been concluded to be more efficient than previous algorithms for generating the binary matrices with fixed margins because it manages to sample with a uniform stationary distribution which is simple for implementation. Such Algorithm will be applied directly into hypothesis testing for assumption of unidimensionality for Rasch Model. Specifically, samples of binary matrices with fixed marginal will be generated by implementing the effective MCMC algorithm. The samples would then be used for computing the *p*-value of the test-statistic for the hypothesis testing.

On the other hand, Verhelst’s paper also compared the result of the *p*-value and performance of the algorithm implementation under different parameter settings, this project also try to implementation the algorithm under different parameter setting based on the mean and standard error of the *p*-value computed. One of the parameters that we will pay attention to is stepsize, which is the number of sample generated in order to collect one effective sample.

Based on the simulation, the resulting p-value fluctuated in the range of 0.27 to 0.33 with standard error range of 0.05 to 0.1. It is concluded that the null hypothesis of unidimensionality is failed to be rejected. Moreover, the standard error of the computed *p*-values has a range of 0.05 to 0.1 which decrease with increasing stepsize.

1. **Background and Objective**

**Existing Algorithms**

Sampling independently and uniformly from the family of binary matrices with fixed marginal has been a challenging problem. In the past literatures, two types of algorithms have been introduced to sample approximately uniform distribution of binary matrices with fixed margins, i.e. importance sampling (IS) and Markov chain Monte Carlo (MCMC). The issue is that these existing algorithms either converge slowly or require a long burn-in period and resulting in highly correlated samples. Chen et al. has developed another IS algorithm which is high efficient for small table but less efficient in medium size tables (300 x 30). (Verhelst, 2008)

In the literature of “An Efficient MCMC Algorithm to Sample Binary Matrices With Fixed Marginals” by Norman D. Verhelst in 2008, a new MCMC algorithm with a uniform stationary distribution has been introduced. This new algorithm is concluded to be converging faster than existing algorithms and more efficient then Chen’s algorithm. Moreover, it could perform nicely for large table. In regard to this, the aim of the report is to simulate the efficient MCMC Algorithm based on the the Norman’s article to conduct uniform sampling of binary matrices with fixed margins.

**Rasch Model**

The application of uniform sampling of binary matrices with fixed margins lies in the exact Rasch model test. The Rasch model is a modelfor analyzing categorical data. It is used to be widely applied in the field of psychometrics where binary response is expressed as a function of subject (e.g. respondent’s ability, personality traits) and item (e.g. item’s difficulty). The parameter estimates in the Rasch model depend only on the marginal totals of the data matrix. Unidimensionality, that is the items in the data sets measure only a single construct, is a fundamental requirement for the Rasch model. (Brentari, Golia ,2007)

Since the main issue presented in Verhelst’s article is to present a methodology to build nonparametric test for the Rasch model, this paper will also simulate the process of such non parametric test.

As an illustration, the algorithm will be applied using the example data set from r-Package {eRm} as starting point to generate samples of binary matrix with fixed margins. Specifically, based on the sample generated, hypothesis testing of unidimensionality of the observed binary matrix by statistics (range of inter-item correlations) will also be conducted. This will be explained in more details in later section of this report. The language used for this project is R.

1. **Description of Data Set**

**Example data sets: xmpl**

The example dataset that will be used in the simulation is a fictitious data sets created in the R-

Package {eRm} (Extended Rasch Modeling

). They are in the form of matrices with binary responses.

xmpl use an observed matrix of 300 rows and 30 columns. The row refers to the subject while the columns refer to the items. This data set will be used as the starting point for the simulation. (Appendix I)

1. **Algorithm and Hypothesis Testing**

Before descripting the MCMC algorithm for sampling the binary matrix, three related concepts, Binomial Operation, Regular Pair and -measure, would be briefly described as below.

* **Binomial Operation**

In a matrix with 2 columns, with *m* of the the rows having row sum = 1, and within these *m* rows, the number of rows with value = 1 in left column = *a*, the transformation rule on the column pairs would be described as:

Assign in the first column *a* ‘1’s to the *m* rows with row sum = 1 and ‘0’s to the other *m* - *a* rows. This defines the first column of the transformed binary matrix. The second column would be the complement of the first column. In other words, for every row, Assign ‘1’ to cells on second column if the value of the cell in the first column on the same row = 1, assign ‘0’ to cells on second column if the value of the cell in first column on the same row = 0.

* **Regular Pair**

For column pair *(i , j)* in a binary matrix with row sum = 1 :

Let be the number of “1”s in the column, and be the number of “1”s in the column, if \* > 0, the column pair is Regular Pair

* **-measure**:

The -measure of is defined as (A) = #{*(i , j)*: *i < j <= k*, *(i , j)* satisfying \* >0 (Regular Pair)}.

**Importance Sampling and Metropolis-Hasting**

To make the stationary distribution uniform, Metropolis-Hasting algorithm is used. Starting from an arbitrary transition matrix = (). The algorithm builds the transition matrix for an arbitrary stationary distribution pi by the transformation:

where

The diagonal elements are:

To get the uniform stationary distribution, set . We also have

Therefore, our algorithm could be described as follow:

Step1: Select randomly a pair from the as (A) regular column pairs of

Step2: Apply a random binomial operation to the selected pair,

(a) If = , repeat step 2

(b) If , and () (), the new state is

(c) If () (), then the new state remain with probability of 1 - ()/(), otherwise the new state is

In this algorithm, step 1 is importance sampling since all regular column pairs have an equal sampling probability despite differences in their neighborhood size. Step 2 is Metropolis-Hasting.

**Hypothesis Testing on unidimensionality of Rasch Model**

According to Norman, the parameter estimates in the Rasch model depend only on the marginal totals of the data matrix. To construct the non-parametric test for the Rasch model, random samples has been generated uniformly from the collection data matrices with the same marginal as the observed data matrix, using the new MCMC algorithm. The samples would then construct the observed distribution of the statistic of the data matrix. We can then determine the *p*-value, which is the exceedance probability of the statistics in the observed sample, and draw conclusion of the hypotheses test based on the p-value. Based on this concept, the hypothesis testing is outlined as below:

: The model (observed matrix) is unidimensional

: The model is not unidimensional

**Test Statistics:**

is the range of the inter-column correlations for a binary matrix. According to Verhelst’s literature, it could be used as a test of the unidimensionality assumption of the Rasch model. To To calculate this statistics, the *phi.range()* function in the R-package {eRm} will be used. (Patrick Mair, 2015)

**Steps for Estimation of *p*-value**

Step 1. Starting from the observed matrix , generate matrices where is the length of burn-in period.

Step 2. After burn-in period, generate \****stepsize*** matrices, and take into samples every ***stepsize*** amount of matrices. For each iteration, only effective matrices are sampled.

Step 3. Compute statistic for each of the effective samples.

Step 4. Calculate the proportion of times the value exceeded the value of the in the observed matrix, take this proportion as an estimation of *p*-value.

Step 5. Repeat step 1 to 4 up to times, calculate the mean and standard error of *p*-value.

**Parameter Setting**

Overall, there are 4 parameters in the implementation of the algorithm:

* The number of replication (): number of iteration of the MCMC sampling algorithm to generate the average *p-*value. For example, R =100 would generate 100 *p*-values to calculate the mean and standard error of the *p*-value. In our implementation, R is set to a value of 10 due to computation limit.
* Burning period (***burn /*** ): the number of sample to be discarded (burnt) before collecting the sample of effective matrix. We have set the burn-in period as 100 in our implementation.
* Number of effective matrix to generate (): Targeted number of matrix to generated for calculating the p-value. We have tried to implement the sampling with N = 100 and N = 1000 which generate very close result of *p*-value (0.29 to N = 1000 and 0.27 for N =100). Therefore, we fix =100 for rest of the implementation setting.
* Step size (***stepsize***): interval of generating one effective matrix in the MCMC sampling, i.e**.**  -1 idle matrices are generated for each effective matric with a ***stepsize*** . We have tried ***stepsize*** = 2, 4, 8, 16 and 32 and the results of using different stepsizes will be discussed in next session.

1. **Conclusion and Discussion**

The final result and conclusion of implementation is as below:

**p-value:**

|  |  |  |
| --- | --- | --- |
| **stepsize** | ***p*-value** | **standard error of *p*-value** |
| 2 | 0.2723 | 0.10501586 |
| 4 | 0.3317 | 0.08685 |
| 8 | 0.297 | 0.08545 |
| 16 | 0.3396 | 0.08044 |
| 32 | 0.303 | 0.05926 |

Table 1: Result of implementation

Since the *p*-values of all ***stepsize*** are > 0.05, the Null hypothesis of model unidimensionality is failed to be rejected.

The results for the estimated standard errors across replications are visualized as below:



Figure 1: p-value of different stepsizes



Figure 2: Standard Error of p-value of different stepsizes

From the graph above, it is clear that the p-value is quite consistent with different step sizes, while the estimated standard error decrease with increasing step sizes.

**Computation time**

During the implementation, the computation time of each iteration increase with step size. This makes sense because as the step size increase, more matrix has to be generated in order to output 1 effective matrix. In other words, for the same number of targeted effective matrix, there is larger number of idle matrix generated, which resulted in a much higher computation time.

**Limitation**

Due to the limitation of computational resources, this simulation use a much smaller parameter setting (***R, N, burn***) than that of in Verhelst’s literature. Therefore, there could be some discrepancies in the output of the test statistics and standard error of the p-value. Nonetheless, the general trend of the result basically agree with what is concluded in Verhelst’s literature.

# References

E. Brentari, S. G. (2007). UNIDIMENSIONALITY IN THE RASCH MODEL: HOW TO DETECT AND INTERPRET. *STATISTICA, anno LXVII*(3).

Patrick Mair, R. H. (2015, 11). *cran.r-project*. Retrieved from https://cran.r-project.org/web/packages/eRm/eRm.pdf

Verhelst, N. D. (2008, 12). An Efficient MCMC Algorithm To Sample Binary Matrices With Fixed Marginals. *PSYCHOMETRIKA, 73*(4).