Basics of Markov Chain Monte Carlo Simulation

Minh Pham

MAT 6932 Mathematical Statistics II

I. Introduction

In Statistics, Markov Chain Monte Carlo (MCMC) is a class of algorithms for sampling from a probability distribution. The general mechanism of MCMC is that, if we can design a Markov chain that has the equilibrium distribution equal to the target distribution, one can obtain a sample of the target distribution by observing the chain after a number of steps. The more steps there are, the closer the sample matches the target distribution.

Markov Chain Monte Carlo are primarily used to approximate large hierarchical models that require integrations over hundreds of unknown parameters. When a Markov chain Monte Carlo method is used for approximating a multi-dimensional integral, an ensemble of "walkers" move around randomly. At each point where a walker steps, the integrand value at that point is counted

towards the integral. The walker then may make a number of tentative steps around the area, looking for a place with a reasonably high contribution to the integral to move into next.

The two most popular Markov Chain Monte Carlo methods are the Metropolis – Hastings algorithm and the Gibbs sampling algorithm. The remainder of this paper is organized as follows. Chapter II discusses the Gibbs sampling algorithm. Chapter III discusses the Metropolis – Hastings algorithm. Chapter IV discusses convergence criteria. Chapter V presents an example with the mixture of Gamma – Poisson model.

II. Gibbs Sampling

Gibbs sampling is a special case of the Metropolis – Hastings algorithm [1]. The idea behind Gibbs sampling is that given a multivariate distribution, it is simpler to sample from a conditional distribution than the marginal distribution.

Suppose the parameter vector θ is divided into d subvectors, $\theta = (\theta_1, \dots, \theta_d)$. At each iteration t, an ordering of the d subvectors of θ is chosen, and, in turn, each θ_j^t is sampled from the conditional distribution given all the other components of θ:

$$p(\theta_j | \theta_{-j}^{t-1}, y)$$

where θ_{-j}^{t-1} represents all the components of θ except for θ_j at their current values. More specifically, the proceed with the following steps:

- We begin with some initial value θ^t
- We want to generate the next sample θ^{t+1}. We sample each component of the vector θ^{t+1} from the distribution of that component conditioned on all other components

sampled so far. Therefore, to sample θ_j^{t+1}, we update it according to the distribution specified by $p(\theta_j^{t+1}|\theta_1^{t+1}, \ldots, \theta_{j-1}^{t+1}, \theta_{j+1}^{t}, \ldots, \theta_n^{t})$.

- Repeat the above steps *n* times

For many problems, it is possible to sample directly from most or all of the conditional posterior distributions of the parameters. We typically construct models using a sequence of conditional probability distributions.

III. Metropolis – Hastings algorithm

The Metropolis–Hastings algorithm can draw samples from any probability distribution $p(\theta)$, provided you can compute the value of a function $f(\theta)$ that is proportional to the density of $p(\theta)$ [4]. The last requirement that $f(\theta)$ should be merely proportional to the density, rather than exactly equal to it, makes the Metropolis–Hastings algorithm particularly useful, because calculating the necessary normalization factor is often extremely difficult in practice. The algorithm proceeds as follows.

1. Draw a starting point θ^0, for which $p(\theta^0|y) > 0$, from a starting distribution $p_0(\theta)$. The starting distribution might be based on an approximation.
2. For t = 1, 2, …:
    a. Sample a proposal θ^* from a proposal distribution at time t, $J_t(\theta^*|\theta^{t-1})$.
    b. Calculate the ratio of the densities,

$$r = \frac{p(\theta^*|y)J_t(\theta^{t-1}|\theta^*)}{p(\theta^{t-1}|y)J_t(\theta^*|\theta^{t-1})}$$

    c. Set $\theta = \begin{cases} \theta^* \; with \; probability \; \min(r, 1) \\ \theta^{t-1} \; otherwise \end{cases}$

Given the current value θ^{t-1}, the transition distribution $T_t(\theta^t|\theta^{t-1})$ of the Markov chain is thus a mixture of a point mass at $\theta^t = \theta^{t-1}$, and a weighted version of the jumping distribution, $J_t(\theta^*|\theta^{t-1})$, that adjust for the acceptance rate.

Proof that this Markov process converges to the target distribution $p(\theta^*|y)$:

The derivation of the algorithm starts with the condition of detailed balance:

$$p(\theta^*|\theta)p(\theta) = p(\theta|\theta^*)p(\theta^*)$$

$$\frac{p(\theta^*|\theta)}{p(\theta|\theta^*)} = \frac{p(\theta^*)}{p(\theta)}$$

The approach is to separate the transition into two sub-steps: the proposal and the acceptance-rejection. The proposal distribution $J(\theta^*|\theta)$ is the conditional probability of proposing a state θ^* given θ, and the acceptance distribution $A(\theta^*|\theta)$ is the conditional probability to accept the proposed state θ^*. The transition probability can be written as the product of them:

$$p(\theta^*|\theta) = J(\theta^*|\theta)A(\theta^*|\theta)$$

Inserting this relation in the previous equation we have:

$$\frac{A(\theta^*|\theta)}{A(\theta|\theta^*)} = \frac{p(\theta^*)}{p(\theta)}\frac{J(\theta|\theta^*)}{J(\theta^*|\theta)}$$

The next step in the derivation is to choose an acceptance that fulfills the condition above. One common choice is

$$A(\theta^*|\theta) = \min(1, \frac{p(\theta^*)}{p(\theta)}\frac{J(\theta|\theta^*)}{J(\theta^*|\theta)})$$

A good proposal distribution has the following properties:

- For any θ, it is easy to sample from $J(\theta^*|\theta)$
- It is easy to compute the ratio *r*
- Each jump goes a reasonable distance in the parameter space (otherwise the random walk moves too slowly).
- The jumps are not rejected too frequently (otherwise the random walk wastes too much time standing still).

The *Metropolis* algorithm is a special case of the Metropolis – Hastings algorithm where the propose distribution is symmetrical, that is $J(\theta|\theta^*) = J(\theta^*|\theta)$. Therefore, the ratio of densities reduces to:

$$r = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)}$$

IV. Convergence Criteria

Burn-in

To diminish the influence of the starting values, we generally discard the first half of each sequence and focus attention on the second half. Our inferences will be based on the assumption that the distributions of the simulated values θ_t , for large enough t, are close to the target distribution, $p(\theta|y)$. We refer to the practice of discarding early iterations in Markov chain simulation as warm-up; depending on the context, different warm-up fractions can be appropriate. We adopt the general practice of discarding the first half as a conservative choice. For example, we might run 200 iterations and discard the first half. If approximate convergence

has not yet been reached, we might then run another 200 iterations, now discarding all of the initial 200 iterations.

Monitoring Convergence:

A good approach to assessing convergence of iterative simulation is based on comparing different simulated sequences with different initial values. We can then diagnose convergence by checking mixing and stationarity. We can also monitor convergence by the potential scale reduction.

We start with some number of simulated sequences in which the warm-up period (which by default we set to the first half of the simulations) has already been discarded. We then take each of these chains and split into the first and second half (this is all after discarding the warm-up iterations). Let m be the number of chains (after splitting) and n be the length of each chain.

For each scalar estimand ψ, we label the simulations as ψ_{ij} (i = 1, …, n; j = 1, …, m) and compute *B* and *W*, the between- and within-sequence variances:

$$B = \frac{n}{m-1}\sum_{j=1}^{m}\left(\bar{\psi}_{.j} - \bar{\psi}_{..}\right)^2$$

$$where\ \bar{\psi}_{.j} = \frac{1}{n}\sum_{i=1}^{n}\psi_{ij}\,, \bar{\psi}_{..} = \frac{1}{n}\sum_{j=1}^{m}\bar{\psi}_{.j}$$

$$W = \frac{1}{m}\sum_{j=1}^{m}s_j^2 \quad where\ s_j^2 = \frac{1}{n-1}\sum_{j=1}^{m}\left(\psi_{ij} - \bar{\psi}_{.j}\right)^2$$

Then the potential scale reduction is estimated by

$$R = \sqrt{\frac{\frac{n-1}{n}W + \frac{1}{n}B}{W}}$$

which declines to 1 as n approaches infinity. If the potential scale reduction is high, then we have reason to believe that proceeding with further simulations may improve our inference about the target distribution of the associated scalar estimand.

V. Example with mixture Gamma-Poisson model

The Gamma-Poisson Shrinkage Model (GPS) is a popular model for the problem of association study between drugs and adverse events.

The model assumes $N_{ij} \sim Poisson(\lambda_{ij} * E_{ij})$ where E_{ij} is the expected count of drug i and adverse event j and N_{ij} is the observed count, and all the $\lambda_{ij}'s$ is drawn from a common prior distribution, which is assumed to be a mixture of two Gamma distributions. The probability density function of the parameter λ_{ij} is given as:

$$\pi(\lambda; \alpha_1, \beta_1, \alpha_2, \beta_2, p) = pg(\lambda; \alpha_1, \beta_1) + (1-p)g(\lambda; \alpha_2, \beta_2), \quad \alpha_1, \beta_1, \alpha_2, \beta_2 > 0, 0 \leq p \leq 1$$

where $g(\lambda; \alpha_1, \beta_1)$ and $g(\lambda; \alpha_2, \beta_2)$ are the probability density functions of the Gamma Distribution with shape parameters α_1, α_2 and scale parameters β_1, β_2, and p is the weight of the first distribution.

We already know that the marginal distribution of N_{ij} is a mixture of 2 negative binomial distributions. We would like to draw samples of λ_{ij} and N_{ij} from this model using the Gibbs sampling and Metropolis – Hastings algorithm and compare with the true distributions. We arbitrarily pick $\alpha_1 = 1, \beta_1 = 1, \alpha_2 = 7.5, \beta_2 = \frac{2}{15}$.

For each algorithm, we generate 4 sampling sequences with initial values (λ_{ij}, N_{ij}) = (1, 1), (1, 10), (10, 1), and (10, 10).

For the Metropolis – Hastings algorithm, we choose the jumping distribution continuous Uniform(0, 30) for λ_{ij} and discrete Uniform(0, 30) for N_{ij}.

The Gibbs Sampling algorithm converged after 58 steps. Figure 1 shows the distribution function of λ_{ij}.

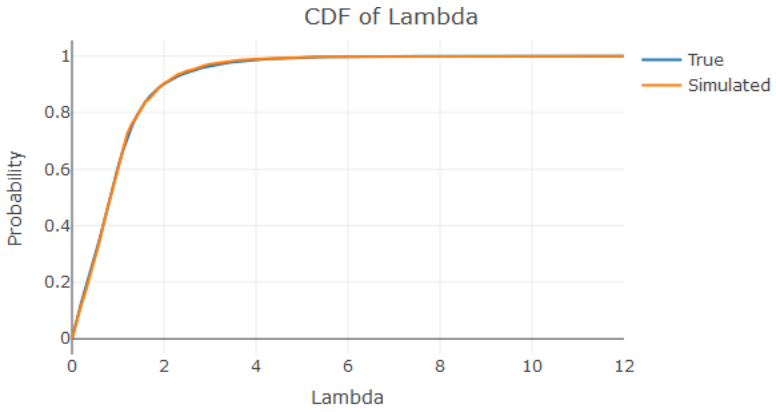

Figure 1

Figures 2 and 3 show the distribution function and histogram of N_{ij}:

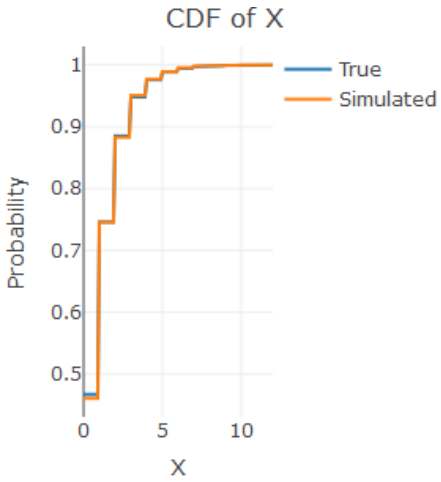

Figure 2

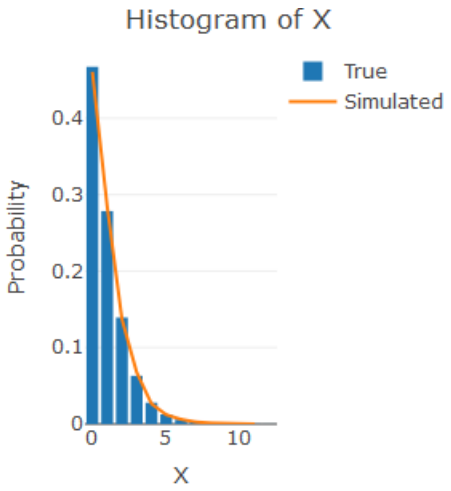

Figure 3

Figure 4 shows the random walks of the 4 sampling sequences:

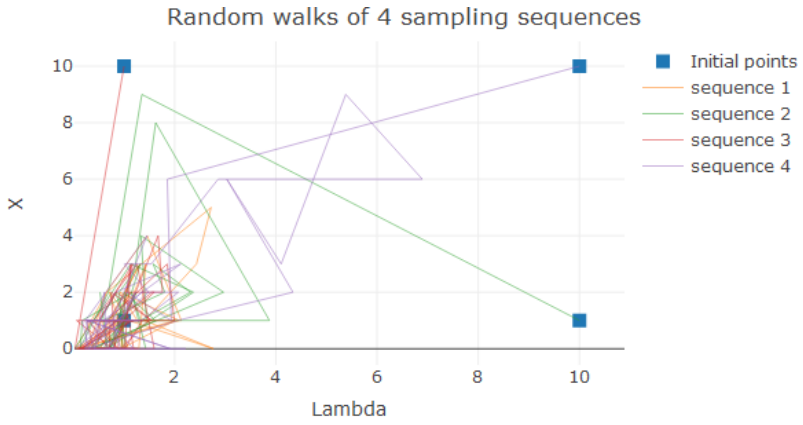

Figure 4

Figure 5 and 6 show the 4 sampling sequences of λ_{ij} and N_{ij}:

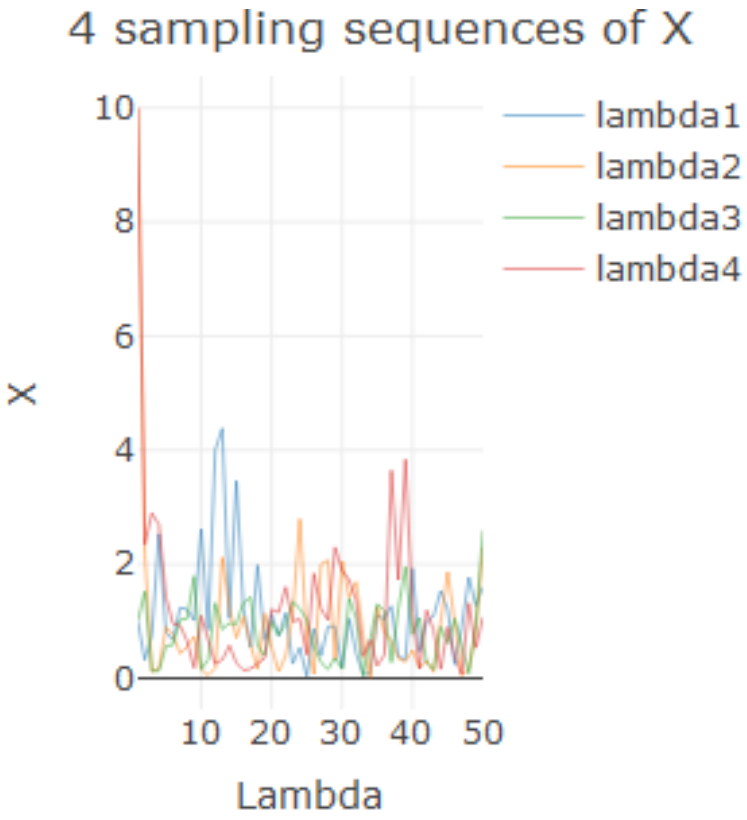

Figure 5

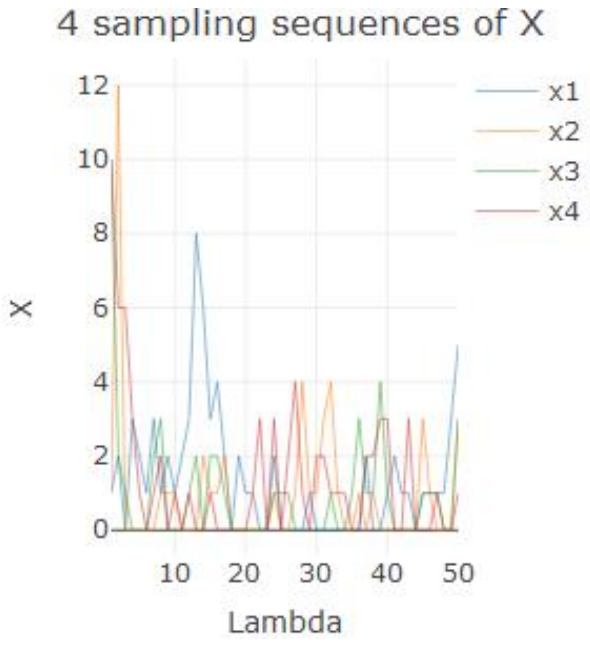

Figure 6

The Metropolis – Hastings algorithm converged after 3448 steps. Figure 7 shows the distribution function of λ_{ij}.

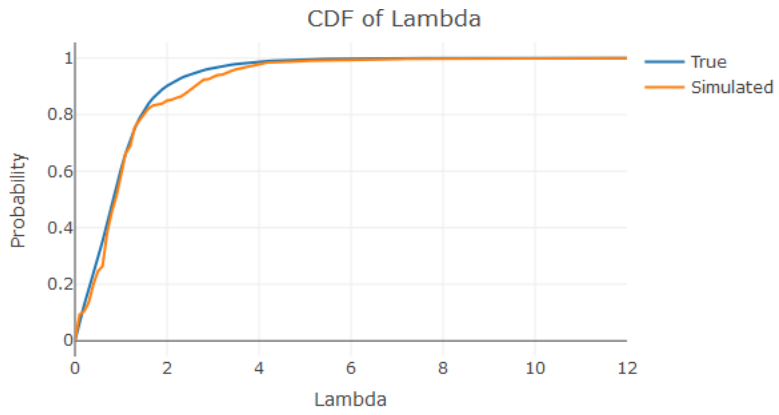

Figure 7

Figures 8 and 9 show the distribution function and histogram of N_{ij}:

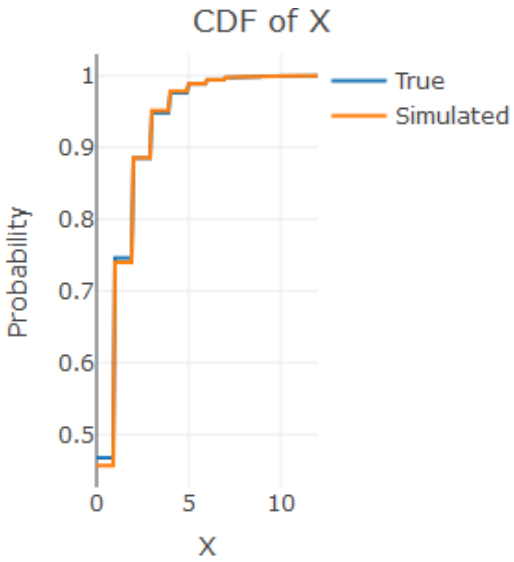

Figure 8

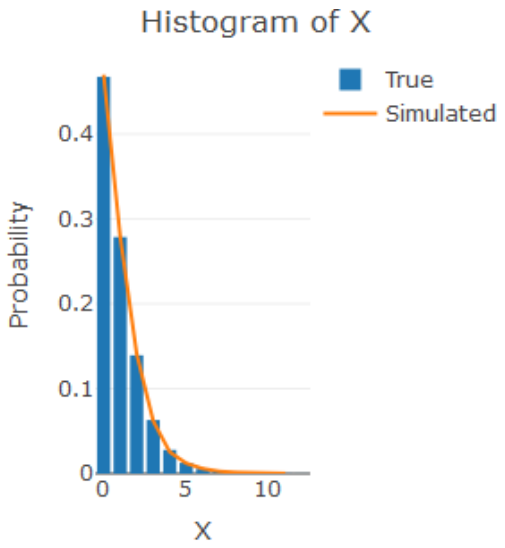

Figure 9

Figure 10 shows the random walks of the 4 sampling sequences:

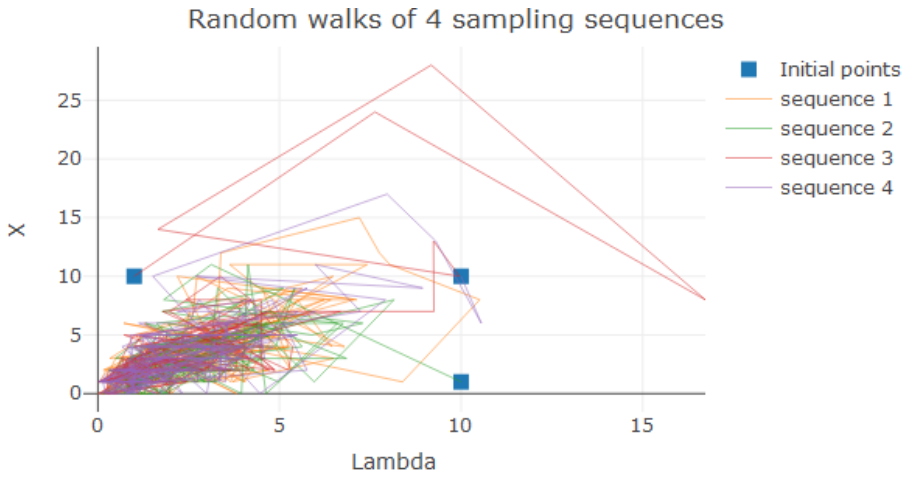

Figure 10

Figure 11 and 12 show the 4 sampling sequences of λ_{ij} and N_{ij}:

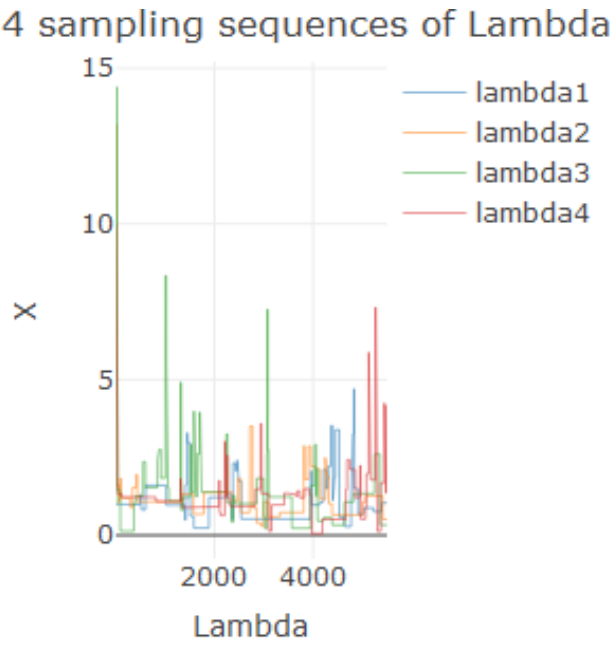

Figure 11

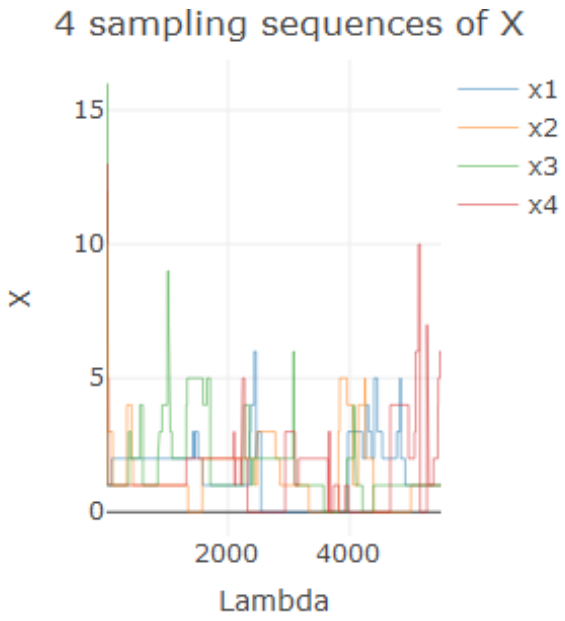

Figure 12

References

[1] Gilks, W. R., Best, N. G., & Tan, K. K. C. (1995). Adaptive rejection Metropolis sampling within Gibbs sampling. Applied Statistics, 455-472.

[2] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. Biometrika, 57(1), 97-109.

[3] Szarfman, A., Machado, S. G., & O'neill, R. T. (2002). Use of screening algorithms and computer systems to efficiently signal higher-than-expected combinations of drugs and events in the US FDA's spontaneous reports database. Drug Safety, 25(6), 381-392.

[4] Greenwood, M., & Yule, G. U. (1920). An inquiry into the nature of frequency distributions representative of multiple happenings with particular reference to the occurrence of multiple attacks of disease or of repeated accidents. Journal of the Royal statistical society, 83(2), 255-279.