Estatística Computacional - Relatório Final

Gabriel Lima Novais

December 4, 2019

Introdução

O tema do trabalho final escolhido para o curso de Estatística Computacional realizado em 2019, foi a análise Bayesiana de modelos de grafos aleatórios exponenciais com base em dois métodos diferentes: O Double Metropolis Hastings e o Noisy Double Metropolis Hastings. Para a elaboração do trabalho foram utilizados como referências dois artigos principais:

- Bayesian Inference in the Presence of Intractable Normalizing Functions (Jaewoo Park e Murali Haran)
- Bayesian Analysis for Exponential Random Graph Models Using the Double Metropolis-Hastings Sampler (Ick Hoon Jin e Faming Liang)

O primeiro artigo serviu como base para o aprendizado e primeira exposição de métodos mais específicos para tratar problemas de inferência Bayesiana em que há constantes de normalização cujas funções são intratáveis. O segundo artigo serviu como exemplo para a aplicação dos métodos aprendidos no primeiro artigo, e desta maneira, a escolha da aplicação dos métodos apresentados residiu na mesma aplicação apresentada neste artigo, isto é, em grafos.

Exponential Random Graphs Models (ERGM)

Recentemente, em especial após a revolução das mídias e aplicativos sociais, o tópico "Redes" vem sendo muito discutido. A tentativa de modelar tais mapeamentos constituem a busca pelo melhor entendimento dos fenômenos em questão. Desta forma, uma maneira relativamente simples de entender o processo de formação dessas redes pode ser obtido pela consideração de hipóteses sobre como tais redes se formam. Os grafos exponenciais aleatórios apontam uma função de densidade de probabilidade que se baseia na distribuição exponencial, cuja família de distribuições possuem características bem definidas. Exemplos de redes com esta suposição, podem ser as redes de midia social, redes de alunos em um colégios, redes de desenvolvimento científico entre outras.

A densidade de probabilidade de dois nós se ligarem em uma rede ERGM é dada por:

$$f(x|\theta) = \frac{exp(\sum_{i=1}^{K} \theta_i S_i(x))}{k(\theta)}$$

Na equação acima temos que a likelihood depende de uma constante de normalização e de uma soma ponderada pelos parâmetros com as estatísiticas suficientes $S_i(x)$. Algumas estatísticas suficientes bem conhecidas podem ser descritas abaixo:

- $e(x) = \frac{1}{2}\sum_{i=1}^{N-1} iD_i(x)$
- $u(x|\tau) = e^{\tau}\sum_{i=1}^{N-2}(1-(1-e^{-\tau})^i)D_i(x)$
- $v(x|\tau) = e^{\tau}\sum_{i=1}^{N-2}(1-(1-e^{-\tau})^i)EP_i(x)$

A primeira estatística listada representa o número de arestas que o grafo possui, onde a função $D_i(x)$ denomina a quantidade de nós que possuem exatamente i arestas. A segunda retrata um grau de distribuição da primeria estatística, onde o parâmetro τ indica a taxa de decrescimento dos pesos dos termos de mairo ordem. Essa estatísitica também é denominada de Geometrically Weighted Degree (GWD). Já a terceira estatística representada, possui certa similaridade com a segunda, mas modifica a função $D_i(x)$ para a função $EP_i(x)$, que representa a quantidade de pares de nós com exatamente i vizinhos em comum. Esta estatísitica também é conhecida como Geometrically Weighted Edgewise Shared Partnership (GWESP).

Os modelos discutidos no artigo "Bayesian Analysis for Exponential Random Graph Models Using the Double Metropolis-Hastings Sampler" (Ick Hoon Jin e Faming Liang) são os que seguem:

- $f(x|\theta) = \frac{exp(\theta_1 e(x)+\theta_2 u(x|\tau))}{k(\theta)}$
- $f(x|\theta) = \frac{exp(\theta_1 e(x)+\theta_2 v(x|\tau))}{k(\theta)}$
- $f(x|\theta) = \frac{exp(\theta_1 e(x)+\theta_2 u(x|\tau)+\theta_2 v(x|\tau))}{k(\theta)}$

Assim como veremos mais tarde, optou-se pelo segundo modelo pois o mesmo envolve diferentes estatísticas. Além disso, como veremos nas seções adiante, é possível verificar que como queremos estimar os parâmetros θ_i nos modelos acima, e como adota-se a modelagem Bayesiana então é preciso amostrar da lilkelihood por algum método de simulação e depois disso utilizar algum estimador para verificar se o mesmo é coerente. O estimador de Bayes utilizado para verificar se os métodos foram suficientemente bons, foi a média da distribuição a posteriori, ou seja:

$$\hat{\theta}_B = \frac{1}{N}\sum_{i=1}^{N}\theta_i$$

Métodos de Variáveis Auxiliares

Antes de entrar no cerne da replicação parcial do artigo, vale expor alguns métodos de variáveis auxiliares para aplicação do Metropolis-Hastings em contextos de modelagem Bayesiana quando na presença de constantes de normalização que não são tratáveis.

Na aplicação do Metropolis-Hastings, precisamos calcular a razão de probabilidade ao qual a aceitação da amostra estará condicionada. Entretanto, quando não sabemos calcular algum termo é intuitivo desejar que o mesmo se cancele neste procedimento, e assim obtenhamos uma razão cujos elementos são conhecidos. É justamente neste aspecto que surge a introdução de uma variável auxiliar cujos conponentes gerem esse cancelamento desejado. Para isto, precisamos apenas escolher de maneira conveniente tal variável auxiliar. O primeiro modeo de tratar esse problema segue abaixo.

Tomemos o modelo seguinte: $h(x|\theta)$ um modelo de probabilidade não normalizado, com $x \in X$ e $\theta \in \Theta$. Então, seja $Z(\theta) = \int_X h(x|\theta)dx$ uma função de normalização e $p(\theta)$ a priori da densidade de θ. Logo a posteriori $\pi(\theta|x)$ será:

$$\pi(\theta|x) \propto p(\theta)\frac{h(x|\theta)}{Z(\theta)}$$

Seja então uma variável auxiliar $y \sim f(y|\theta, x)$ e suponha $\pi(\theta|x)$ tal como descrito acima. Assim a target aumentada será dada por $\pi(\theta, y|x) \propto f(y|\theta, x)p(\theta)h(x|\theta)/Z(\theta)$ e cuja densidade marginal será fornecida pela expressão $\int_X \pi(\theta, y|x)dy = \pi(\theta|x)$. Considere a proposal conjunta descrita por $q(\theta', y'|\theta, y) = q(y'|\theta')q(\theta'|\theta)$. Por fim tome convenientemente $q(y'|\theta')$ como $h(y'|\theta')/Z(\theta')$.

Agora o cálculo da probabilidade de aceitação do Metropolis-Hastings será tal que:

$$\alpha = min(1, \frac{\pi(\theta, y|x)q(\theta, y|\theta', y')}{\pi(\theta', y'|x)q(\theta', y'|\theta, y)}) \implies$$

$$\alpha = min(1, \frac{f(y'|\theta', x)p(\theta')h(x|\theta')Z(\theta)q(y|\theta)q(\theta|\theta')}{f(y|\theta, x)p(\theta)h(x|\theta)Z(\theta')q(y'|\theta')q(\theta'|\theta)}) \implies$$

$$\alpha = min(1, \frac{f(y'|\theta', x)p(\theta')h(x|\theta')Z(\theta)h(y|\theta)Z(\theta')q(\theta|\theta')}{f(y|\theta, x)p(\theta)h(x|\theta)Z(\theta')h(y'|\theta')Z(\theta)q(\theta'|\theta)}) \implies$$

$$\alpha = min(1, \frac{f(y'|\theta', x)p(\theta')h(x|\theta')h(y|\theta)q(\theta|\theta')}{f(y|\theta, x)p(\theta)h(x|\theta)h(y'|\theta')q(\theta'|\theta)})$$

Existe ainda o Exchange Algorithm, que modifica este último, sendo $y \sim \frac{h(y|\theta)}{Z(\theta)}$, tomando a densidade conjunta aumentada por $\pi(\theta, \theta', y|x) \propto p(\theta)\frac{h(x|\theta)}{Z(\theta)}q(\theta|\theta')\frac{h(y|\theta)}{Z(\theta)}$ e fazendo a proposal de troca $s(\theta, \theta'|\theta*, \theta*')$ como sendo simétrica. Com todas estas modificações é possível gerar uma probabilidade de aceitação do MH dada por:

$$\alpha = min(1, \frac{\pi(\theta', \theta, y|x)s(\theta, \theta'|\theta*, \theta*')}{\pi(\theta', \theta, y'|x)s(\theta*, \theta\ *'\ |\theta, \theta')}) \implies$$

$$\implies \alpha = min(1, \frac{\pi(\theta', \theta, y|x)}{\pi(\theta', \theta, y'|x)}) \implies$$

$$\implies \alpha = min(1, \frac{p(\theta')h(x|\theta')h(y|\theta)q(\theta|\theta')}{p(\theta)h(x|\theta)h(y|\theta')q(\theta'|\theta)})$$

Na próxima seção, comentaremos como é possível realizar uma modificação nesse método do Exchange Algorithm para selecionar a variável auxiliar.

Noisy e Double Metropolis Hastings (NDMH e DMH)

O método de Double Metropolis Hastings é bastante similar ao do Metropolis Hastings original. A contribuição do DMH reside na introdução de uma variável auxiliar que termina por aumentar o espaço-estado da cadeia de Markov original. Isto posto, o nome do método surge simplesmente pelo fato de que a atualização da variável auxiliar é realizada mediante uma quantidade de passos pré determinadas composta por um outro Metropolis Hastings.

A matemática por trás do método é a que segue. Considere uma probabilidade de transição em m passos de atualização via MH da cadeia que atualiza x em y, definida por $P^m_{\theta'}(y|x)$, e cujo θ' possui distribuição estacionária dada por $\frac{h(y|\theta')}{Z(\theta')}$. Com isso a suposição de balanceamento detalhado nos fornece a seguinte igualdade:

$$\frac{P^m_{\theta'}(x|y)}{P^m_{\theta'}(y|x)} = \frac{h(x|\theta')}{h(y|\theta')}$$

Agora fazendo igual na seção anterior, e modificando o Exchange Algorithm, temos que para o Metropolis-Hastings, a probabilidade de aceitação se torna:

$$\alpha = min(1, \frac{p(\theta')h(x|\theta')h(y|\theta)q(\theta|\theta')}{p(\theta)h(x|\theta)h(y|\theta')q(\theta'|\theta)}) \implies$$

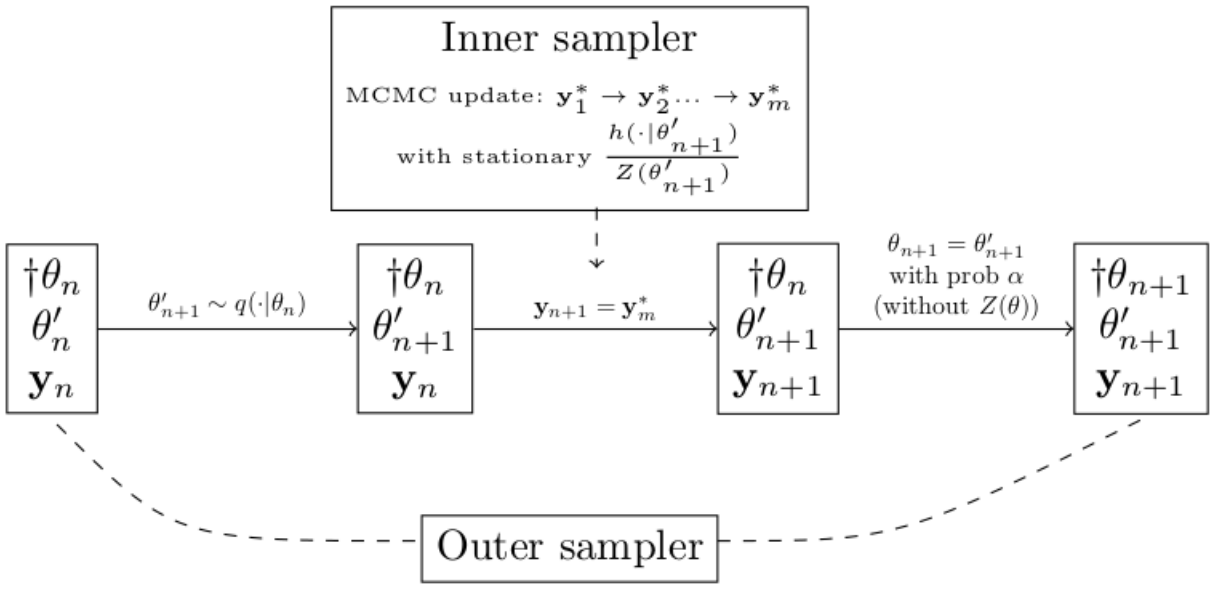
$$\implies \alpha = min(1, \frac{p(\theta')h(y|\theta)q(\theta|\theta')}{p(\theta)h(x|\theta)q(\theta'|\theta)} \frac{P^m_{\theta'}(x|y)}{P^m_{\theta'}(y|x)})$$

Caso os valores de θ venham de um Random Walk, temos pela simetria deste processo que a nossa probabilidade de aceitação se torna:

$$\alpha = min(1, \frac{p(\theta')h(y|\theta)q(\theta|\theta')}{p(\theta)h(x|\theta)q(\theta'|\theta)} \frac{P^m_{\theta'}(x|y)}{P^m_{\theta'}(y|x)}) \implies$$

$$\alpha = min(1, \frac{h(y|\theta)}{h(x|\theta)} \frac{P^m_{\theta'}(x|y)}{P^m_{\theta'}(y|x)})$$

O esquema ilustrativo a seguir consegue definir bem o método:



Os passos de implementação são os que seguem

(1) Dada uma distribuição a priori $\pi(\theta)$ tomamos uma amostra θ', e assim começamos com um θ_t.

(2) Entramos no Inner Sampler, gerando uma variável auxiliar $y \sim f(y|\theta')$ proveniente de m atualizações do MH começando com x. A probabilidade de transição de x para y nesses m passos são fornecidas de forma específica. Então calculamos a razão de aceitação desse y e tomamos ele para o próximo ponto.

(3) Terminamos o algoritmo com o segundo Metropolis-Hastings de forma que com a probabilidade de aceitação (calculada sem o termo intratável) aceitamos $\theta' = \theta_{t+1}$ ou rejeitamos fazendo $\theta_t = \theta_{t+1}$.

Vale observar que alguns parâmetros podem ser escolhidos de maneira que o resultado seja o melhor. Um exemplo seria o próprio número de passos m, cujo valor costuma ser proporcional ao tamanho do dado (resultado heurístico). Para o caso das redes, veremos que esse número de passos será equivalente a atualização de cada elemento da matriz e assim o número de passos totais resultante deste procedimento é o próprio tamanho da matriz de adjacência do grafo em questão. Outro ponto ainda sobre o valor de m reside no fato de que como o DMH (e consequentemente o NDMH como veremos mais a frente) é um algoritmo assintóticamente inexato uma vez que a condição de balanceamento detalhado não funciona para o primeiro Metropolis-Hastings sem que o segundo possua um m suficientemente grande, e desta maneira precisamos escolher esse valor com uma magnitude elevada. Isto implica dizer que para redes pequenaas o procedimento não consegue fornecer uma aproximação tão boa quanto para redes maiores.

Com base no que foi discutido veremos como se constitui o Noisy Double Metropolis-Hastings que foi aplicado na replicação parcial do artigo em questão.

Se um kernel de transição de uma cadeia de Markov atende a condição de balanceamento detalhado com respeito a função de target, então ela é uma cadeia assintóticamente exata. Quando aproximamos esse kernel de transição por outro, as amostras geradas apenas se assemelharão a aproximação da target anterior. Esse tipo de estratégia compõem aquilo que denota-se por "Noisy MCMC". Nestes métodos procuram-se distâncias variacionais de modo que conseguimos determinar quando a técnica se qualifica como assintóticamente exata ou inexata. Esses métodos de Noisy costumam ser analisados como um híbrido de métodos de aproximação de likelihood com métodos de variáveis auxiliares. O método de Noysi Double Metropolis-Hastings utilizado é considerado um método assintóticamente inexato e suas principais modificações realizados no DMH associam-se a introdução de muitas variáveis auxiliares no procedimento de "Inner Sample".

Ou seja, ao invés de utilizarmos apenas uma única variável y que resultaria numa probabilidade de aceitação já sem a constante de normalização, procura-se acrescentar variáveis $y_1, y_2, y_3, ..., y_n$ de forma que a nova probabilidade se torna:

$$\alpha = min(1, \frac{p(\theta')h(x|\theta')q(\theta|\theta')}{p(\theta)h(x|\theta)q(\theta'|\theta)} \frac{1}{n} \sum_{i=1}^{n} \frac{h(y_i|\theta)}{h(y_i|\theta')})$$

Logo, os procedimentos deste algoritmo se assemelham e muito ao anterior, e assim os passos do esquema anterior também são efetuados, com a diferença que no Inner Sampler diversas variáveis são geradas e para depois computar a segunda probabilidade de aceitação o que se faz é avaliar as funções geradas nessas variáveis e calcular o termo adicional dentro do somatório destacado acima.

Aplicação e Resultados

Gerar uma rede exponencial aleatória não é uma tarefa fácil. Entretanto, assumimos que dado algum exemplo poderíamos estimar via pacotes já implementados no R o valor das constantes dos modelos dessas redes, com

a aplicação do Markov Chain Monte Carlos Maximum Likelihood Estimator (MCMCMLE), algoritmo este que foi utilizado no segundo artigo do trabalho final.

Para a realização da aplicação dos algoritmos acima, procuraram-se duas redes, uma com 10 nós, extraídas da base "Florentine" do R, e uma rede com 50 nós gerada no R, mas sem ser necessariamente exponencial, isto é, apenas consideramos que era, por hipótese, e comparamos o seu valor com o obtido pelo método de MCMCMLE. Desta maneira poderíamos avaliar tais algoritmos para um dado modelo de rede, modelo este que foi o segundo, pois como mencionado em seções anteriores, possui duas estatísticas suficientes diferentes.

Antes de entrar nos cálculos do algoritmo propriamente dito, foi fundamental a implementação dos algoritmos $D_i(x)$ e $EP_i(x)$ para calcular as estatísticas suficientes. Para o primeiro, bastou somar as linhas da matriz de adjacência, pegar os resultados e alocá-los em um vetor de modo que fosse possível contar a frequência deles. Para o segundo algoritimo houve um processo mais extenso, isto é, precisou-se somar cada duas linhas da matriz, depois excluir as que não representavam a soma de dois nós ligados e depois disso contar o número de 2 que aparecia em cada soma dessa. Com a anotação destas respostas em um vetor o que se fez foi contar a frequência dos resultados e pronto tínhamos o vetor de respostas. Com isso, construíamos as duas estatísticas suficientes pedidas.Em sguida foi realizado o DMH.

O procedimento para elaboração do DMH foi a seguinte:

1 Tomar θ_0 de uma normal multivariada com média zero e variância alta.

2 Considerar a evolução dos θ_t como um random walk cujo step size modifica a variância da normal e a média é atualizada pelo θ_{t-1}

3 Atualizar os elementos da matriz de adjacência por uma probailidade específica, por meio de uma rodada única de Gibbs Sampler.

4 Com a matriz atualizada calcular a razão de aceitação e verificar se o θ_t é aceito e assim finalizar o MH principal.

Para o NDMH o que foi feito foi simplesmente alterar o passo 4 de modo a repetí-lo 3 vezes e computar uma razão de aceitação diferente. Foram utilizadas no NDMH, portanto, apenas duas variáveis auxiliares a mais. Talvez o ideal fosse a implementação de mais variáveis e estabelecer uma comparção entre os métodos, porém devido a certa escassez de tempo o mesmo não foi possível.

Custos computacionais observados durante o processo: além de precisar de muitas iterações (por ser uma técnica de Monte Carlos), o algoritmo também precisa varrer uma matriz, e tanto para o DMH quanto para o NDMH verifica-se a ordem do algoritmo de n^2. Entretanto como a aplicação é voltada para redes, verifica-se que no máximo a complexidade se torna n^4, sendo que dificilmente isto poderá ocorrer, uma vez que o tamanho da rede deverá ser da magnitude do número de iterações. Grande parte dos custos relacionaram-se não somente ao MH, mas também ao cálculo das estatísticas suficentes em especial a $EP(x)$, que necessitou de muitas operações custosas.

A implementação foi incialmente feita em Python, mas pelo fato de não executar em tempo viável, optou-se por implementar em Julia, que nos forneceu um tempo de execução aproximadamente 5 vezes menor. Para a rede de 10 nós os parâmetros procurados pela estimação do modelo 2 eram de $\theta_1 = -1.66$ e $\theta_2 = 0.15$. Já para o de 50 nós o resultado deveria fornecer algo próximo de $\theta_1 = -3.19$ e $\theta_2 = 0.39$. Para efetuar a simulação procurou-se estabelecer valores de variâncias altos (200) e um número de iterações similares ao do segundo artigo (10500 para o de 10 nós e apenas 1000 para o de 50 nós). Todos os demais parâmetros foram estabelecidos segundo os valores do segundo artigo. Vale ressaltar que devido a custos computacionais difíceis de lidar e com um tempo reduzido, não foi possível simular diversas vezes (50 vezes

por exemplo como foi feito no segundo artigo) e assim verificar a média e o desvio padrão dos resultados. Entretanto para todas as poucas vezes que foi simulado, os resultados demosntraram-se satisfatórios.

O quadro comparativo entre os métodos pode ser visto a seguir:

Método	Rede	Tempo	Iteração	θ_1	θ_2
NDMH	10	26 min e 27 s	10500	-0.796426	0.405188
DMH	10	6 min e 51 s	10500	-1.74165	0.882828
DMH	50	2h 47 min 57 s	1000	-3.27527	0.159238

Os gráficos a seguir demonstram as cadeias e as distribuições que compõem a tabela anterior. As cadeias demonstram um chute inicial distante da região de estabilidade, fato que foi promovido pela variância inicial escolhida com o valor de 200. Entretanto vemos que depois do Brunin (proporcional ao tamanho da cadeia), a região de estabilidade aparece e a distribuição dos ficam bem próximas ao que deveria ser.

O Burnin de cada linha da tabela foi de (cima para baixo respectivamente) 500, 200 e 80 iterações. As distribuições que seguem após os gráficos das cadeias não consideraram este Burnin.

Cadeia de θ pela iteração (10 nós - DMH)

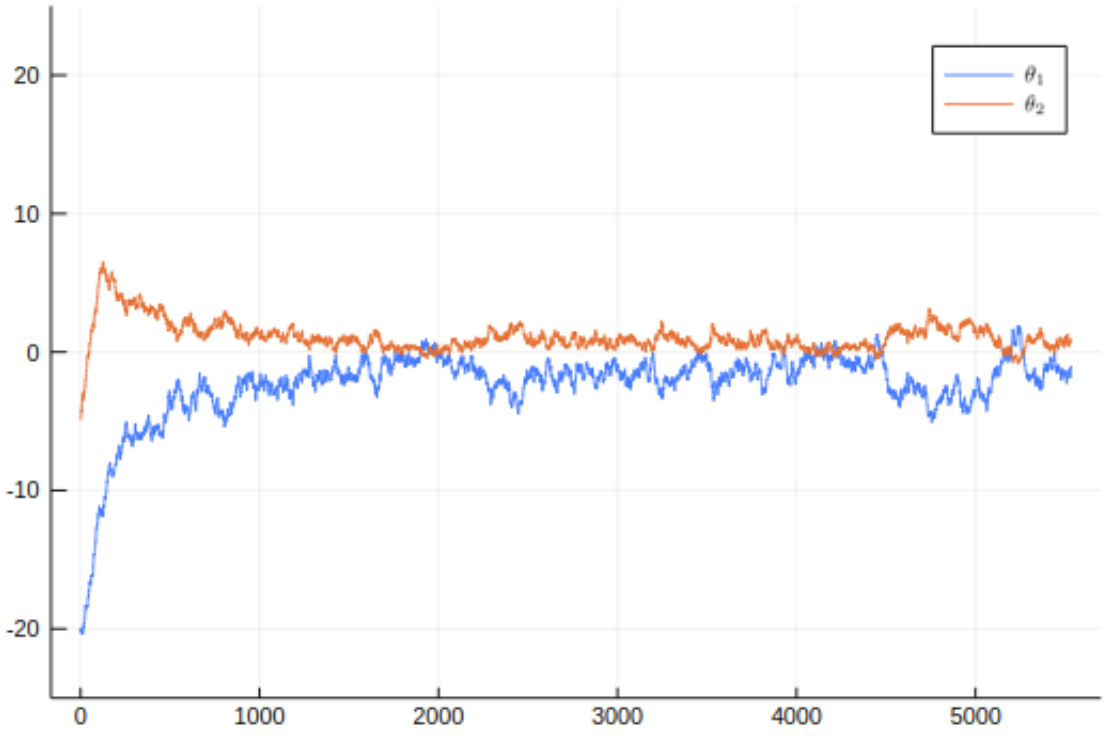
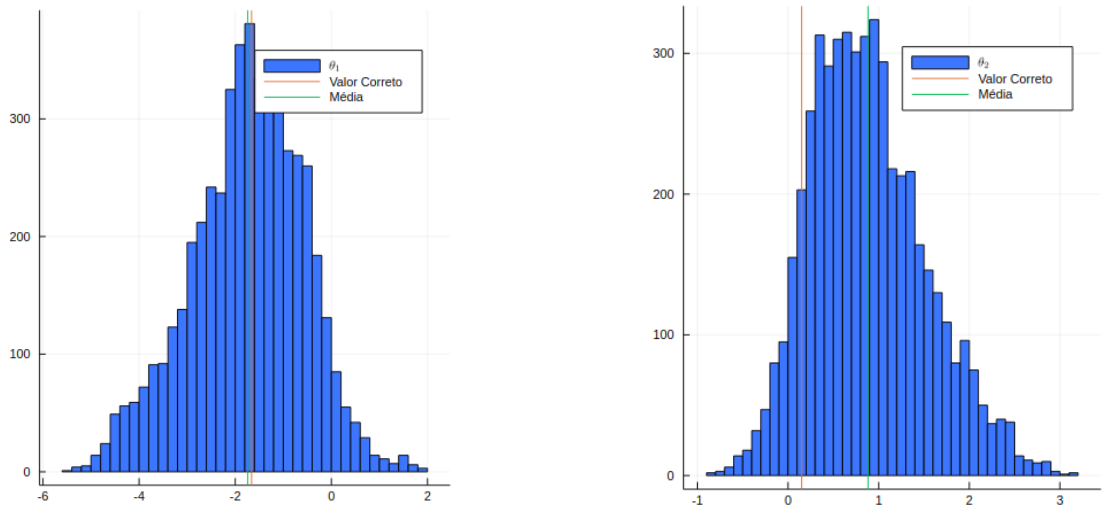


Figure 1: Depois do processo de Burnin, pela iteração 500 a região de estabilidade é atingida.

Distribuições de θ_1 e θ_2



Cadeia de θ pela iteração (10 nós - NDMH)

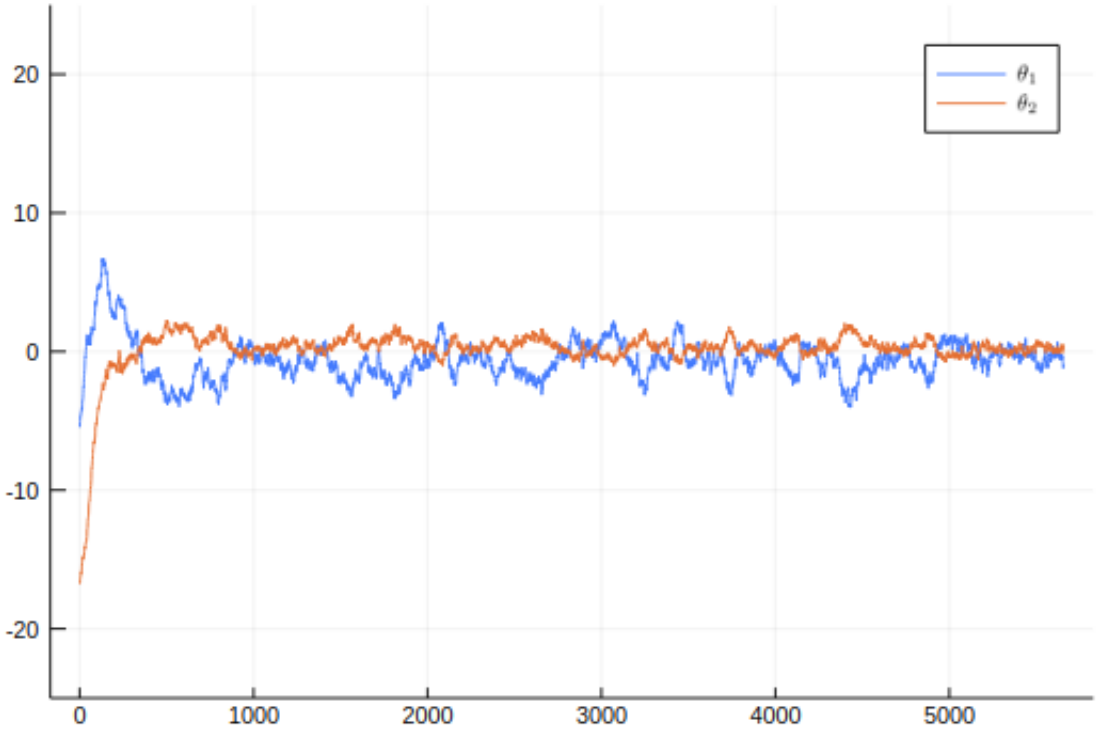
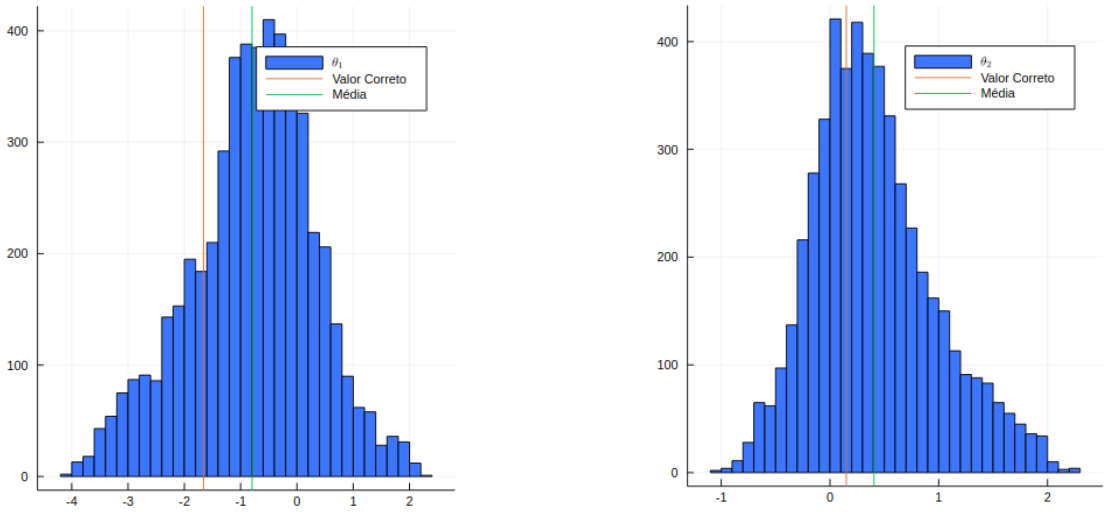


Figure 2: Depois do processo de Burnin, pela iteração 200 a região de estabilidade é atingida.

Distribuições de θ_1 e θ_2



Cadeia de θ pela iteração (50 nós - DMH)

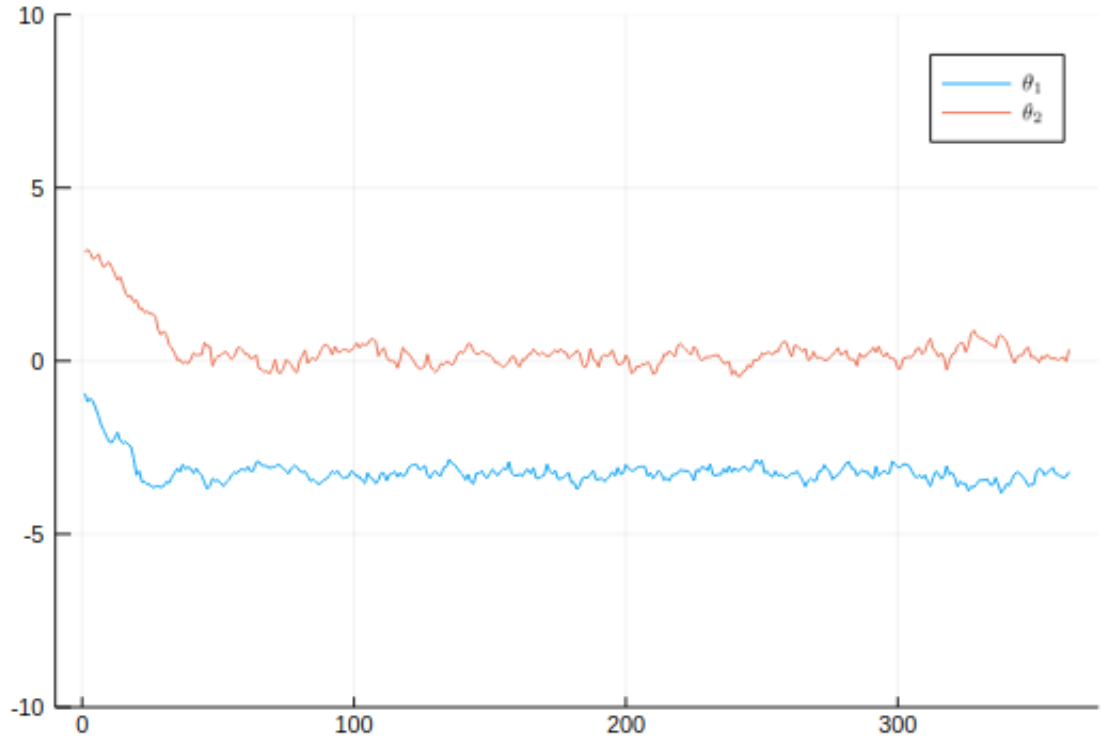
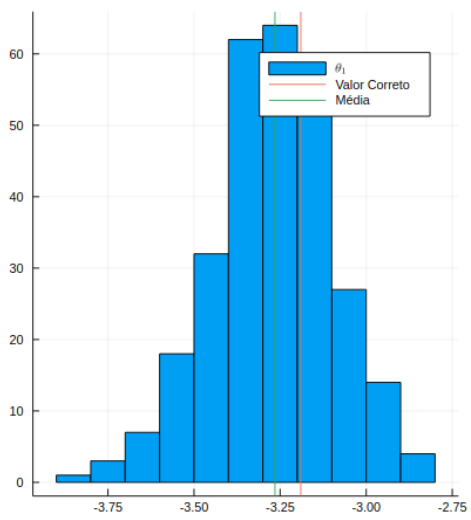
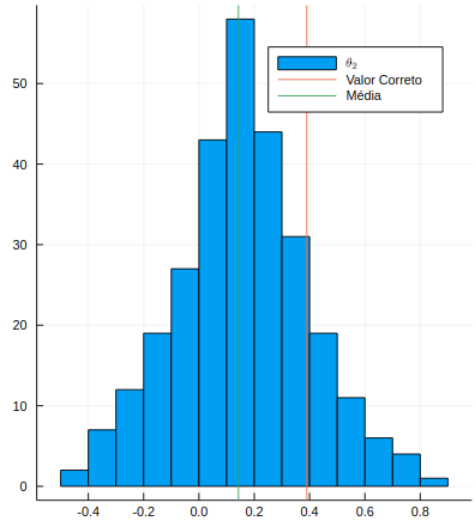


Figure 3: Depois do processo de Burnin, pela iteração 80 a região de estabilidade é atingida.

Distribuições de θ_1 e θ_2





Referências

- JIN, Ick Hoon; LIANG, Faming. Bayesian analysis for exponential random graph model using double Metropolis-Hastings sampler. Unpublished working paper, p. 1-19, 2009.
- LIANG, Faming; JIN, Ick-Hoon. A Monte Carlo Metropolis-Hastings algorithm for sampling from distributions with intractable normalizing constants. Neural computation, v. 25, n. 8, p. 2199-2234, 2013.
- PARK, Jaewoo; HARAN, Murali. Bayesian inference in the presence of intractable normalizing functions. Journal of the American Statistical Association, v. 113, n. 523, p. 1372-1390, 2018.
- STIVALA, Alex; ROBINS, Garry; LOMI, Alessandro. Exponential random graph model parameter estimation for very large directed networks. arXiv preprint arXiv:1904.08063, 2019.