Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

LUCAS MARIANI PAIVA CALDEIRA BRANT

MONOGRAFIA DE PROJETO ORIENTADO EM COMPUTAÇÃO II

GERAÇÃO DE AMOSTRAS N-DIMENSIONAIS USANDO O METROPOLIS ADJUSTED LANGEVIN ALGORITHM

Belo Horizonte

2022 / 2º semestre
Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

GERAÇÃO DE AMOSTRAS N-DIMENSIONAIS USANDO O METROPOLIS ADJUSTED LANGEVIN ALGORITHM

por

LUCAS MARIANI PAIVA CALDEIRA BRANT

Monografia de Projeto Orientado em Computação II

Apresentado como requisito da disciplina de Projeto Orientado em Computação II do Curso de Bacharelado em Ciência da Computação da UFMG

Prof. Dr. Flavio Vinicius Diniz de Figueiredo
Orientador(a)

Belo Horizonte
2022 / 2º semestre

RESUMO

Atualmente o professor Flavio Diniz está realizando um projeto de machine learning que consiste em encontrar a cópula de uma determinada distribuição de dados. Sendo breve, uma cópula é a função de distribuição cumulativa conjunta de N distribuições marginais. A cópula de uma dada distribuição multivariada pode ser usada em conjunto com as marginais para gerar amostras desta ainda que ela não tenha uma fórmula conhecida. Quando finalizada a rede neural, essas amostras serão geradas usando métodos de MCMC. Durante a POC I foi implementado um metropolis hastings capaz de gerar amostras de distribuições N-dimensionais, para o POC II esse algoritmo foi adaptado para o Metropolis Adjusted Langevin Algorithm (MALA). Apesar de alguns resultados inconclusivos, os testes com a nova implementação indicam que esta gera amostras mais próximas da distribuição desejada.

Palavras-chave: MCMC, Metropolis-Hastings, Amostragem, Distribuição de Probabilidade, Langevin.

ABSTRACT

The professor Flavio Diniz is currently working on a machine learning project that consists of finding the copula function for a given data distribution. Briefly, copulas can be described as the joint cumulative density function of n marginal distributions. The copula function of a multivariate distribution combined with its marginals can be used to generate samples of said distribution even if said distribution does not have a known formula. Once the neural network is working, the samples of these distributions shall be generated by utilizing the MCMC class of methods. During POC I a base version of the Metropolis-Hastings algorithm capable of sampling from N-dimensional distributions was implemented, and through the course of this project this implementation was adapted to follow the Metropolis Adjusted Langevin Algorithm (MALA). Despite some inconclusive results, the tests suggest that the new implementation generates samples closer to the desired distribution.

Keywords*:* MCMC, Metropolis-Hastings, Sampling, Probability Distribution, Langevin.

SUMÁRIO

1 INTRODUÇÃO

Machine learning têm-se tornado uma ferramenta fundamental em diversas áreas de atuação, mas em certas aplicações críticas como sistemas médicos e bancários a pouca disponibilidade de dados tem sido um desafio na criação e aplicação de algoritmos mais robustos [1]. Essa baixa disponibilidade se dá tanto pela ausência do dado em si como também devido a preocupações com a privacidade dos usuários. A geração de dados sintéticos seria uma forma de lidar com esse desafio.

Uma ideia que tem sido explorada em trabalhos recentes é a utilização de cópulas para geração de dados sintéticos [1]. A cópula é uma função de distribuição cumulativa multivariada que descreve a relação de dependência entre variáveis aleatórias [2]. O teorema de Sklar [Sklar, 1959] diz que qualquer distribuição multivariada pode ser representada por uma função de suas distribuições marginais e uma cópula, o que significa que encontrando a cópula e as marginais podemos gerar dados que seguem qualquer distribuição.

O professor Flávio Diniz vem conduzindo um trabalho de machine learning que explora essa ideia. O objetivo é desenvolver uma rede neural que estima a cópula de uma dada distribuição, e a partir dela gerar amostras seguindo tal distribuição. O propósito deste projeto é implementar um método que realize essa parte final, ou seja, que gere amostras seguindo a distribuição definida pela cópula e as marginais.

Para gerar essas amostras serão utilizadas a classe de métodos Markov Chain Monte Carlo (MCMC). Durante o POC I o algoritmo dessa classe implementado foi Metropolis Hastings, enquanto no POC II esta implementação foi adaptada para o Metropolis Adjusted Langevin Algorithm (MALA). Este algoritmo utiliza o conceito de dinâmicas de Langevin, um processo estocástico que define a evolução de um sistema ao longo do tempo.

Esses algoritmos geram amostras seguindo uma função densidade de probabilidade a partir de uma função proporcional. A principal diferença entre os dois é que no MALA os novos estados são propostos usando dinâmicas de Langevin. Este método foi escolhido por demonstrar bons resultados na amostragem de distribuições de alta dimensionalidade.

Este artigo está estruturado da seguinte forma: Na seção 2 são mencionados e discutidos alguns trabalhos relacionados. A seção 3 contém uma apresentação formal do algoritmo e uma descrição de sua implementação. Na seção 4 são exibidos os resultados acompanhados de breve uma análise da performance, e por fim na última seção são apresentadas as conclusões e direções para projetos futuros.

2 CONTEXTUALIZAÇÃO E TRABALHOS RELACIONADOS

O algoritmo de Metropolis Hastings foi originalmente proposto por [Hastings, 1970], e é uma generalização do método desenvolvido por [Metropolis et al, 1953]. O algoritmo busca gerar amostras de uma distribuição com função densidade de probabilidade π, considerada a distribuição a alvo, a partir de uma distribuição proposta q, que deve ter densidade proporcional a π.

Ao longo da execução é construída uma cadeia de Markov X que é ergódica e estacionária a π, ou seja, dado tempo um tempo t suficiente a transição de um estado X^t para o estado X^{t+1} segue a distribuição de π [6]. Dessa forma, cada estado X^t pode ser considerado uma amostra da distribuição de π.

Diversos artigos foram publicados explicando em profundidade o algoritmo proposto por Hastings, como por exemplo [CHIB & GREENBERG, 1995] e [Robert, 1999]. [Robert, 1999] apresenta uma breve contextualização histórica do algoritmo, acompanhada de uma detalhada implementação em pseudocódigo. O artigo também discute algumas extensões do algoritmo de Metropolis Hastings.

O Metropolis Adjusted Langevin Algorithm (MALA) foi proposto por Julian Besang em 1994 [8] e desde então vários aprimoramentos foram feitos em relação ao algoritmo, como em [Girolami & Calderhead, 2011], onde os autores propõe a utilização de manifolds de Riemann para melhorar a amostragem de distribuições com alta dimensionalidade e correlação. Além de apresentar o novo método, este artigo também introduz o algoritmo original de maneira bastante didática, e por ser mais acessível do que artigo de origem foi usado como fonte principal na realização deste trabalho.

3 DESENVOLVIMENTO DO TRABALHO

3.1 Metropolis-Hastings

O algoritmo começa em algum estado inicial X^1, e aos poucos vai explorando o espaço local. Isso significa que os primeiros estados gerados são altamente dependentes do valor inicial de X^1, mas devido à natureza ergódica do algoritmo após uma quantia grande o suficiente de transições essa correlação deixa de ser significativa. Uma técnica utilizada para reduzir o impacto do estado inicial no resultado final é descartar as primeiras amostras geradas como *burn-in*. Neste trabalho as primeiras quinhentas amostras geradas por cada distribuição amostrada foram descartadas.

Se queremos amostrar uma distribuição alvo π a partir de uma distribuição proposta q, basta encontrar o próximo estado da cadeia de Markov X. Podemos transicionar a partir de um estado X^t para o estado X^{t+1} seguindo os seguintes passos:

1. *Gere um estado candidato* $X' \sim q(X'|X^t)$

2. *Calcule acceptance rate* $a = min\left(\frac{\pi(X')}{\pi(X^t)} * \frac{q(X^t|X')}{q(X'|X^t)}, 1\right)$

3. $X^{t+1} = \begin{cases} X' \; com \; probabilidade \; a \\ X^t \; com \; probabilidade \; 1 - a \end{cases}$

A ideia por trás do algoritmo é explorar o espaço amostral e caminhar para estados mais prováveis de serem gerados por π. O algoritmo sempre aceita mover para um estado com maior probabilidade do que o atual (calculado pela taxa de aceitação), e às vezes rejeita se mover para estados com menor probabilidade. Dessa forma a cadeia tende a permanecer em regiões de alta densidade em π, ocasionalmente explorando áreas menos densas.

Existem algumas ressalvas importantes a serem feitas, a escolha da distribuição proposta é fundamental pois ela precisa ser capaz de gerar todos os valores pertencentes a π. Além disso, uma má escolha pode levar a uma taxa de aceitação muito baixa ou até impedir que o algoritmo convirja para a distribuição alvo. A distribuição proposta utilizada foi a gaussiana multivariada.

3.2 Metropolis Adjusted Langevin Algorithm

A ideia geral permanece a mesma do algoritmo original, a principal alteração sendo feita na geração de novos estados. A partir de um estado qualquer X^t, a proposta X^{t+1} é feita utilizando o gradiente da distribuição desejada. A expectativa é que ao longo do processo as propostas geradas se aproximem do amostral com maior probabilidade de aceitação.

Podemos gerar novos estados seguindo o Metropolis Adjusted Langevin Algorithm com a equação:

$$X' = X + \frac{\tau^2 \nabla L(X)}{2} + \tau z$$

Em que $L(X)$ é a log-densidade de π em um dado ponto X, τ um time step fixo e z é um ruído aleatório seguindo uma distribuição gaussiana com médias 0 e matriz de covariância igual à uma matriz identidade. Além disso, também são feitas alterações no cálculo da taxa de aceitação. Sendo a distribuição proposta q uma normal multivariada, a média e matriz de covariância utilizada passam a ser calculadas da seguinte forma:

$$\mu(X, \tau) = X + \frac{\tau^2}{2} \nabla L(X)$$
$$cov = \tau^2 I$$

Com I sendo uma matriz identidade com diagonal de tamanho igual à dimensão das amostras. O valor ideal para o time step τ ainda é objeto de estudo [9], nos testes realizados seu valor foi definido em 0,2. O restante do algoritmo funciona exatamente como descrito na subseção anterior.

3.3 Implementação

Durante a POC I foi implementado o algoritmo de Metropolis Hastings capaz de gerar amostras N-dimensionais, que ao longo deste projeto foi adaptado para gerar amostras seguindo o MALA. Boa parte do tempo foi dedicado ao estudo do algoritmo, pois haviam várias implementações diferentes do algoritmo e algumas não apresentavam resultados satisfatórios. O algoritmo foi implementado em python, utilizando as bibliotecas JAX, numpy e scipy.

Algumas distribuições multivariadas amostradas com o algoritmo não tinham uma definição paramétrica da função densidade de probabilidade, e portanto foi preciso estimá-las usando kernel density estimation (KDE). O KDE é uma forma não paramétrica de estimar a PDF de uma variável aleatória [8]. Sendo breve, o KDE avalia a quantidade de dados em uma determinada região do espaço amostral, a PDF estimada tem alta densidade nas áreas com muitas observações, e baixa densidade em áreas com poucas observações. As PDFs das distribuições foram estimadas com o kernel gaussiano e utilizando 100 mil pontos de cada distribuição.

Por depender do cálculo do gradiente, o MALA é altamente afetado pelo número de amostras usado para estimar o KDE. Foi observado empiricamente uma diferença significativa no gradiente em um mesmo ponto entre um KDE estimado com 100 mil pontos e outro estimado com 1 bilhão, o que certamente impacta no desempenho final do algoritmo. Dito isso, o tempo de execução para o último caso era muito longo, então os testes foram feitos usando o KDE estimado com 100 mil pontos.

4 RESULTADOS E DISCUSSÃO

Como dito antes, o propósito final deste trabalho é ser utilizado para gerar amostras com a cópula e distribuições marginais encontradas pela rede neural sendo desenvolvida no projeto do professor Flávio Diniz. Como este projeto ainda não foi concluído, a performance do algoritmo foi testada com 2 distribuições: uma distribuição beta unidimensional com parâmetros $\alpha = 2$, $\beta = 8$ e uma distribuição bidimensional que gera pontos no formato de duas "luas". Nas seções abaixo comparamos o desempenho do Metropolis-Hastings implementado durante o POC I com o MALA implementado no POC II. Para cada uma dessas distribuições foram geradas 10 mil amostras.

4.1 Beta

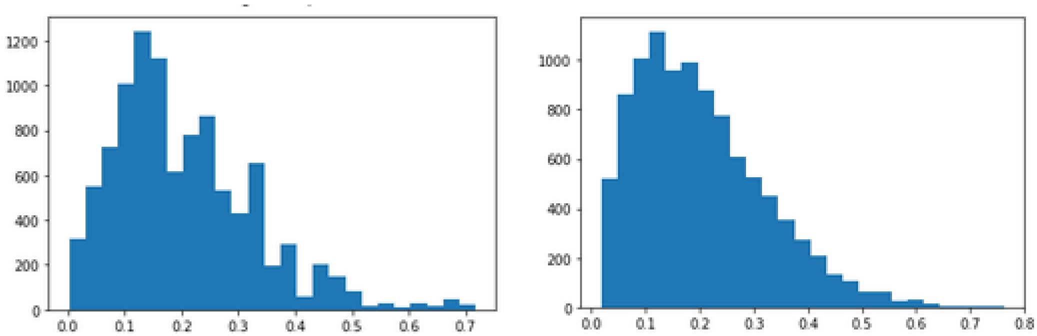

Figura 1: Comparação entre 10 mil pontos gerados pelo Metropolis-Hastings (esquerda) e pelo MALA (direita). Histograma com 25 bins.

Apesar de ambas as amostras se aproximarem da distribuição real, fica evidente que as amostras geradas pelo MALA convergem muito mais rapidamente para a beta real. Dito isso, ambos os métodos convergem para a distribuição real se gerarmos um número suficientemente grande de amostras.

4.2 Luas

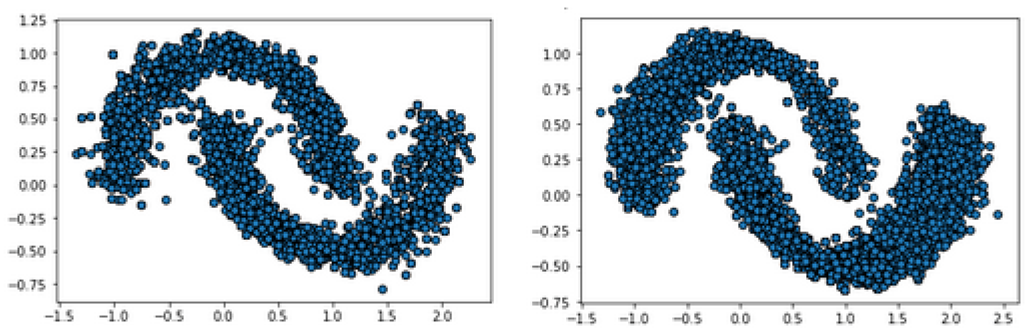

Figura 2: Comparação entre 10 mil pontos gerados pelo Metropolis-Hastings (esquerda) e pelo MALA (direita).

A distribuição das luas teve que ser estimada usando o KDE, e aqui podemos ver o problema mencionado anteriormente. Em alguns trechos a dispersão dos pontos gerados pelo MALA é mais estreita do que aquela dos pontos gerados pelo Metropolis-Hastings, mas especialmente nas bordas podemos notar que os pontos estão mais dispersos do que na implementação original. A suspeita é que esse erro seja causado devido ao cálculo impreciso do gradiente, mas a demora no tempo de execução impediu que essa hipótese fosse verificada. Ainda assim, o algoritmo foi capaz de gerar amostras seguindo a distribuição desejada.

5 CONCLUSÕES

O objetivo original de implementar o Metropolis Adjusted Langevin Algorithm capaz de gerar amostras de distribuições N-dimensionais foi concluído. O algoritmo apresentou uma melhora clara em relação ao Metropolis-Hastings quando comparados com distribuições unidimensionais, mas os testes com distribuições de maior dimensionalidade é inconclusivo.

Uma dificuldade na realização dos testes foi encontrar boas distribuições para amostrar. Distribuições acima de 3 dimensões não podem ser analisadas com plots, e as ferramentas disponíveis para plotar gráficos tridimensionais são precárias ou muito complexas. Algo a ser investigado seria testar o desempenho dos algoritmos com uma distribuição bidimensional em que sua fórmula é conhecida, de modo a não depender da estimação por KDE. Durante os testes preliminares foi comparado o desempenho amostrando de uma normal multivariada, mas não houve diferença significativa entre os algoritmos. Dito isso, ambas as implementações estão aptas para gerarem amostras N-dimensionais.

6 REFERÊNCIAS

[1] KAMTHE, Sanket; ASSEFA, Samuel; DEISENROTH, Marc. Copula Flows for Synthetic Data Generation. **arXiv preprint arXiv:2101.00598**, 2021. Disponível em: https://arxiv.org/pdf/2101.00598.pdf. Acesso em: 17 jul. 2022.

[2] SCHMIDT, Thorsten. Coping with copulas. **Copulas-From theory to application in finance**, v. 3, p. 34, 2007.

[3] SKLAR, M. Fonctions de repartition an dimensions et leurs marges. **Publ. inst. statist. univ. Paris**, v. 8, p. 229-231, 1959.

[4] HASTINGS, W. Keith. Monte Carlo sampling methods using Markov chains and their applications. 1970.

[5] METROPOLIS, Nicholas et al. Equation of state calculations by fast computing machines. **The journal of chemical physics**, v. 21, n. 6, p. 1087-1092, 1953.

[6] ROBERT, Christian P.; CASELLA, George. The metropolis—Hastings algorithm. In: **Monte Carlo statistical methods**. Springer, New York, NY, 1999. p. 231-283.

[7] CHIB, Siddhartha; GREENBERG, Edward. Understanding the metropolis-hastings algorithm. **The american statistician**, v. 49, n. 4, p. 327-335, 1995.

[8] BESAG, Julian. Comments on “Representations of knowledge in complex systems” by U. Grenander and MI Miller. **J. Roy. Statist. Soc. Ser. B**, v. 56, n. 591-592, p. 4, 1994.

[9] GIROLAMI, Mark; CALDERHEAD, Ben. Riemann manifold langevin and hamiltonian monte carlo methods. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, v. 73, n. 2, p. 123-214, 2011.

[10] PARZEN, Emanuel. On estimation of a probability density function and mode. **The annals of mathematical statistics**, v. 33, n. 3, p. 1065-1076, 1962.