
PRIMEIROS PASSOS NA OBTENÇÃO DE PARÂMETROS COSMOLÓGICOS UTILIZANDO MATRIZES DE COVARIÂNCIA COSMOLÓGICAS SEM RUÍDO

AS ASTROCIENTISTAS

✉ **Natalí Soler Matubaro de Santi**

Instituto de Física, Universidade de São Paulo,
São Paulo, 05508-090, Brasil,
natalidesanti@gmail.com

✉ **Luis Raul Weber Abramo**

Instituto de Física, Universidade de São Paulo,
São Paulo, 05508-090, Brasil,
abramo@if.usp.br

RESUMO

Matrizes de covariância são uma das peças mais importantes na análise de dados em Cosmologia: elas não apenas representam o entendimento sobre a natureza das incertezas, mas refletem a propagação dos erros estatísticos e dependem das suposições, devido aos modelos teóricos utilizados, para reduzir os dados. Para representar os verdadeiros erros estatísticos, muitos dados são necessários para construir essas matrizes, ou seja, centenas de milhares de observações ou simulações caríssimas, algo que nem sempre possui obtenção viável. Para resolver esse problema, foi proposto o uso de técnicas de aprendizado de máquina, com uma *pipeline* completa para tal. Foi implementada uma simulação de um campo Gaussiano com três parâmetros. Então, o espectro de potências linear foi calculado para cada mapa produzido e centenas de matrizes de covariância, usando diferentes números de espectros, foram calculadas. As matrizes foram utilizadas como dados de entrada em uma rede neural convolucional para remover o ruído daquelas criadas com poucos dados. Por fim, as matrizes de covariância limpas obtidas foram utilizadas para recuperar os parâmetros da simulação, utilizando algoritmos de Monte Carlo acoplados a cadeias de Markov (MCMC). Os resultados mostraram que essa técnica é capaz de produzir boas matrizes de covariância, mesmo com poucos dados de entrada, diminuindo muito os erros dos parâmetros cosmológicos obtidos.

Palavras-chave matrizes de covariância, aprendizado de máquina, obtenção de parâmetros

1 Introdução

A Cosmologia atual passa e passará, dentro dos próximos anos, por um incrível aumento de dados disponíveis, oriundos de telescópios como o *Dark Energy Survey* (DES) [1], o *Javalambre Physics of the Accelerating Universe Astrophysical Survey* (J-PAS) [2], o *Prime Focus Spectrograph* (PFS) [3] e o *Dark Energy Spectroscopic Instrument* (DESI) [4]. Um dos desafios inerentes dessa grande quantidade de dados se dá com as matrizes de covariância, que serão cada vez maiores e mais complexas. Tais matrizes são responsáveis por conter a propagação dos erros estatísticos intrínsecos às observações e por conectar a teoria para descrevê-los. No caso dos espectros de potência dos traçadores das estruturas em largas escalas a estimativa desses observáveis inclui uma variedade de premissas, que vão desde a função de seleção dos traçadores, a profundidade das imagens, a probabilidade de distribuição dos objetos, bem como o modelo escolhido em questão.

Entretanto, o grande número de simulações necessário para testar estes efeitos se limita pela demanda intensa de recursos computacionais, principalmente para estimar parâmetros cosmológicos [5]. Guiados pelo exemplo do espectro de potências, algo de $O(n_T^2 n_k^2)$ (onde n_T é o número de traçadores, n_k é o número de k bins) simulações é suficiente para que a matriz de covariância convirja, ou seja, estime parâmetros sem vieses, com acurácia e precisão. Logo, enquanto para um único traçador esse número escala com n_k^2 , para múltiplos traçadores, esse número aumenta por n_T^2 . Por isso é de extrema importância desenvolver métodos para alcançarmos matrizes de covariância precisas rapidamente.

Trabalhos como a Referência [6] já mostraram como os erros na estimativa das covariâncias se propagam, aumentando o erro na medida dos parâmetros cosmológicos. Além disso, diferentes esforços já foram feitos na busca de matrizes computadas com um número menor de simulações, como compressão de dados, no caso da Referência [5] ou mesmo com a redução da dimensão dos observáveis cosmológicos, como no caso da Referência [7]. Ao mesmo tempo, trabalhos reforçam a fraca dependência

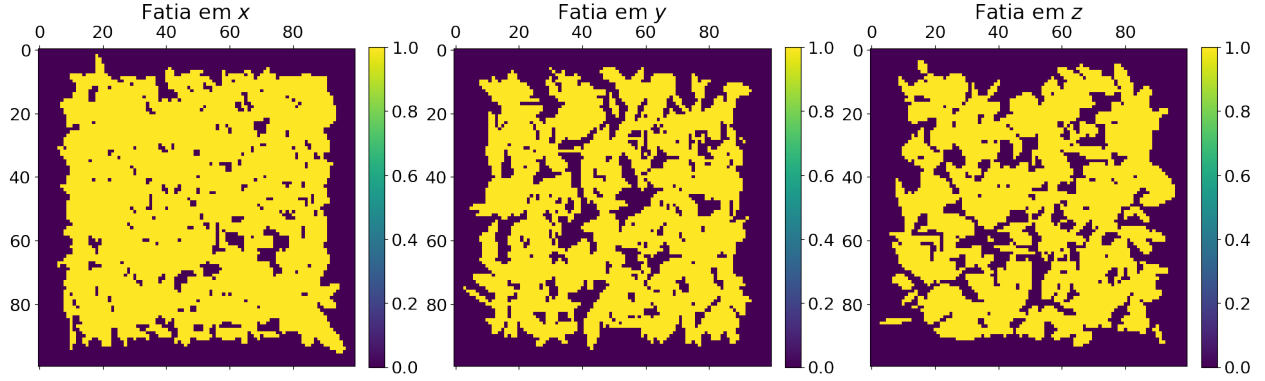


Figura 1: Visualização de fatias da máscara utilizada. A mesma é composta por elipsoides que são esticados e rodados aleatoriamente na caixa. Regiões internas a eles valem um e regiões fora da máscara, valem zero.

da cosmologia com relação as próprias matrizes de covariância, na estimativa dos parâmetros cosmológicos [8], enquanto que já existem testes de diferentes métodos para a produção rápida de catálogos de matéria escura e suas matrizes de covariância comparadas as caríssimas simulações de N-corpos [9].

Neste trabalho foi proposto o uso de aprendizado de máquina para a obtenção de matrizes de covariância tão boas quanto aquelas obtidas com um grande número de simulações, usando apenas aquelas com um baixo número destas. Primeiro, foi implementada uma simulação de um campo Gaussiano, que será vista na seção 2, bem como foram obtidos os espectros de potência para cada mapa. Em seguida foram calculadas as matrizes de covariância (ver seção 3). Então, o problema foi montado do ponto de vista da técnica de aprendizado de máquina, na qual foi utilizado um algoritmo removedor de ruído, na seção 4. Segue-se com a obtenção dos parâmetros cosmológicos na seção 5, sob a perspectiva do teorema de Bayes. Por fim, as conclusões e próximas etapas do trabalho são apresentadas na seção 6.

2 A simulação

Foi simulado um campo Gaussiano randômico $\delta(k)$ com média zero e cuja variância foi equivalente ao seguinte espectro de potências

$$P(k) = P_0 \exp \left[-\frac{(k - k_0)^2}{2\sigma_0^2} \right], \quad (1)$$

usando os parâmetros fiduciais $P_0 = 1 \text{ Mpc}/h$, $k_0 = 0.15h/\text{Mpc}$ e $\sigma_k = 0.03h/\text{Mpc}$. As simulações foram feitas em uma caixa tridimensional de lado $L = 500 \text{ Mpc}/h$ e células cúbicas de $l = 5 \text{ Mpc}/h$. A escolha do campo em questão foi feita devido a necessidade de uma simulação suficientemente barata de modo que pudessem ser obtidos centenas de milhares de mapas. O número total de espectros obtidos foi de $N = 6 \cdot 10^5$. Entretanto, porque era necessária a indução de uma maior correlação dos modos (de maneira que os espectros obtidos fossem mais próximos daqueles em um *survey* real), cada um dos mapas foi mascarado com uma máscara de elipsoides esticados e rodados aleatoriamente. Regiões fora dos elipsoides foram, portanto, desconsideradas do mapa. A máscara utilizada ocupava um volume de $\sim 50\%$ do volume da caixa e pode ser vista na Figura 1.

O espectro de potência foi calculado tomando

$$\langle \tilde{\delta}(\vec{k}) \tilde{\delta}(\vec{k}') \rangle = (2\pi)^3 \delta_D^3(\vec{k} - \vec{k}') P(k) \quad (2)$$

e os resultados do mesmo podem ser vistos na Figura 2. À esquerda, comparam-se o espectro fiducial, convoluído para incorporar o efeito da máscara e uma simulação mascarada, aleatoriamente escolhida. Já à direita, apresenta-se a mesma situação, mas para espectros sem o efeito da máscara. É evidente que a máscara atua na supressão da amplitude do espectro para baixos e altos valores de k .

3 As matrizes de covariância

Usando os espectros obtidos, as matrizes de covariância de amostragem Cov_{ij} , i.e., matrizes quadradas que medem a relação linear entre diferentes modos de k , de diferentes espectros, foram calculadas como

$$Cov[P(k_i), P(k_j)] = \frac{1}{(N-1)} \sum_{l=1}^N [P(k_i)_l - \bar{P}(k_i)] \cdot [P(k_j)_l - \bar{P}(k_j)], \quad (3)$$

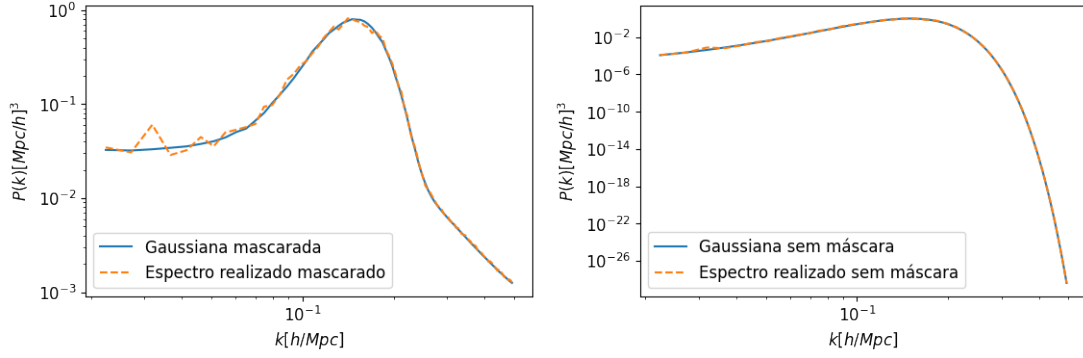


Figura 2: Comparando os espectros de potência obtidos: à esquerda, comparam-se o espectro fiducial mascarado e o espectro medido, para uma simulação escolhida aleatoriamente; à direita, comparam-se o espectro fiducial sem máscara e mais um espectro medido, também escolhido aleatoriamente, mas sem o efeito da máscara.

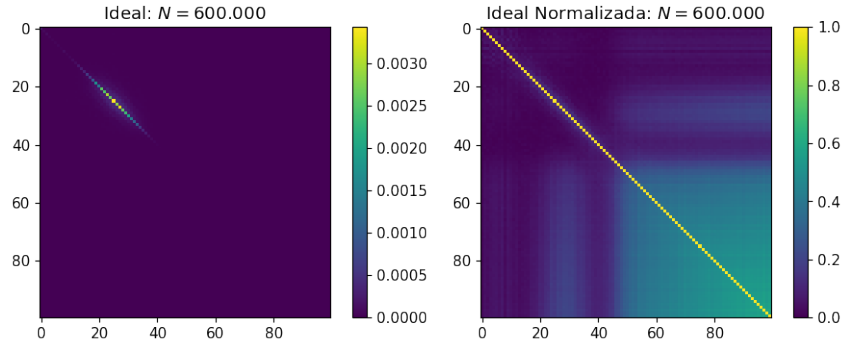


Figura 3: Matriz de covariância (à esquerda) e matriz de correlação (à direita) para $N = 6 \cdot 10^5$ espectros.

onde N é o número de espectros considerado e $\bar{P}(k_i)$ é o espectro médio para o i -ésimo bin de k . É interessante notar que a matriz de covariância possui elementos com um grande intervalo de valores em diferentes ordens de magnitude, o que faz com que sua visualização não seja tão simples. Por essa razão, a matriz de correlação $Corr_{ij}$, definida como

$$Corr_{ij} = \frac{Cov_{ij}}{\sqrt{Cov_{ii} Cov_{jj}}}, \quad norma_{ij} = \sqrt{Cov_{ii} Cov_{jj}}, \quad (4)$$

foi obtida. A quantidade denotada como $norma_{ij}$ é o fator de normalização para cada k_i bin. A matriz de correlação pode ser pensada como a matriz de covariância melhorada, de modo que não possui dimensão e sua diagonal vale exatamente um. Considerando o número máximo de espectros obtido ($N = 6 \cdot 10^5$) as matrizes podem ser visualizadas na Figura 3. Note que as mesmas são chamadas de matrizes ideais, dado que elas utilizam o número máximo de espectros obtidos.

4 Construindo o problema do ponto de vista do aprendizado de máquina

O objetivo do presente trabalho é obter uma melhor estimativa das matrizes de covariância, que seja o mais próxima possível daquela obtida com um número extremo de realizações do espectro do campo Gaussiano, mas sem ser necessário utilizar uma quantidade extrema de espectros. E.g., veja as matrizes com $N = 6 \cdot 10^5$ espectros, na Figura 3, as quais chamamos de matrizes ideais. Dessa forma, a ideia é que esse método seja capaz de obter matrizes que lembrem esse cenário ideal, partindo de matrizes ruins (obtidas usando apenas centenas de espectros), depois de submetê-las a um processo de “limpeza”, usando apenas simulações baratas. O resultado final deve então ser a reconstrução de matrizes de covariância que sejam boas o suficiente, que chamamos aqui de matrizes limpas, para serem utilizadas nas cadeias de Monte Carlo para estimar os parâmetros simulados.

4.1 Algoritmo removedor de ruídos

O método de aprendizado de máquina utilizado foi um algoritmo removedor de ruídos. Em síntese, esses algoritmos recebem imagens que contêm ruído, aprendem o que é ruído e como remover o mesmo, com base em imagens que não o possuem. Como

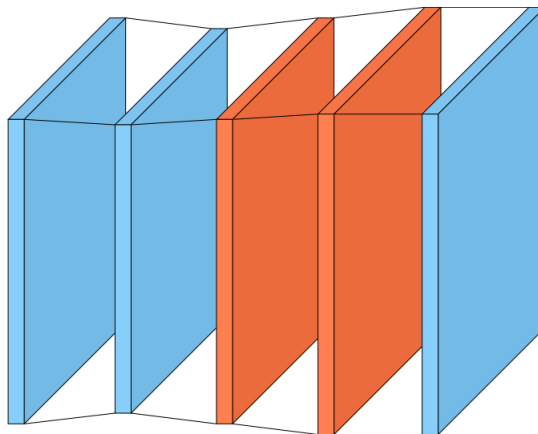


Figura 4: Visualização da arquitetura utilizada para o modelo removedor de ruídos. Foram utilizadas camadas de convolução bidimensional para o codificador (primeira e segunda camada) e as respectivas transpostas para o decodificador (duas últimas camadas). Por fim, foi utilizada uma camada convolucional bidimensional, mantendo das dimensões da imagem de entrada.

resultado, temos imagens muito próximas daquelas alvo, sem ruído. Este trabalho foi feito com o uso de auto-codificadores, um tipo de rede neural convolucional, modelos muito utilizados em visão computacional. A ideia por trás destes algoritmos é, com base em uma imagem de entrada, assimilar uma importância (pesos e *bias*, que são aprendidos pela rede) para vários aspectos/objeto na imagem e ser capaz de diferenciar um do outro, de modo a reconstruir a imagem final [10].

Mais especificamente, tem-se um processo codificador, responsável por lidar com as imagens de entrada, e um processo decodificador, capaz de reconstruir a imagem de entrada. Esse processo é composto por camadas que compõem a rede convolucional. Por sua vez, essa rede aprende padrões e invariantes translacionais através das camadas, de modo que a informação não apenas fique relacionada a cada valor específico, mas também à sua localização espacial. Além disso, cada camada é responsável por aplicar uma transformação linear e outra não-linear e propagá-la até a próxima camada. A transformação não-linear é chamada de função de ativação (por exemplo, *Leaky Rectified Linear Unit (Leaky-ReLU)* e *Hyperbolic Tangent (tanh)*). Os dados passam por esse mesmo processo um determinado número de vezes chamado de épocas e, a ideia é que, para cada uma delas, uma função chamada perda (como a *Mean Squared Error (MSE)*) seja minimizada por meio de um otimizador [11]. Esta função tem por objetivo medir a performance do resultado final. No caso de um algoritmo removedor de ruídos, a função perda mede quão próxima é a imagem limpa da imagem alvo.

Interpretando o algoritmo dessa forma, o conjunto de imagens de entrada é composto pelas matrizes ruins (construídas com poucos espectros, n), sendo as imagens com ruído e suas respectivas matrizes alvo (originalmente “sem ruído” e, portanto, com muitos espectros, N). O conjunto de imagens de saída é composto das matrizes limpas, resultado da transformação das matrizes ruins. Em cada modelo foram usados ds pares de imagem. Ainda mais, foram construídos diferentes modelos com diferentes matrizes ruins de número de espectros n para matrizes alvo com mesmo número de espectros N . A razão para esses diferentes modelos foi encontrar o primeiro e menor valor de n para o qual os erros entre as matrizes de saída limpas e as matrizes alvo fossem baixos e praticamente constantes.

Foram utilizadas $ds = 600$ matrizes e foi desenvolvido um modelo de aprendizado de máquina para cada combinação de matrizes ruins com $n \in [100, 2500]$ e alvo com mesmo $N = 6000$. Em cada modelo foram avaliadas as curvas de função de perda para o MSE. A arquitetura dos modelos pode ser vista na Figura 4. Foram utilizadas duas camadas de convolução bidimensional (respectivamente com 64 e 32 filtros) para o codificador e as respectivas transpostas para o decodificador, seguido de uma última camada convolucional, para recuperar a dimensão da imagem. Nas camadas internas, a função de ativação *Leaky-ReLU*, com parâmetro $\alpha = 0.3$, foi empregada. Na camada de saída, a *tanh* foi utilizada como ativação. Além disso, foi utilizado o otimizador de Adam e a função perda escolhida foi a MSE. Todo o código foi construído usando a biblioteca `keras` [12], em `Python` [13].

4.2 Resultados

Como já comentado, na seção 3, visualmente, as matrizes de covariância não carregam tanta informação, dado que a mesma está encapsulada nos valores de cada entrada. Entretanto, ao normalizarmos as matrizes, usando a norma (ver equação 4) da matriz

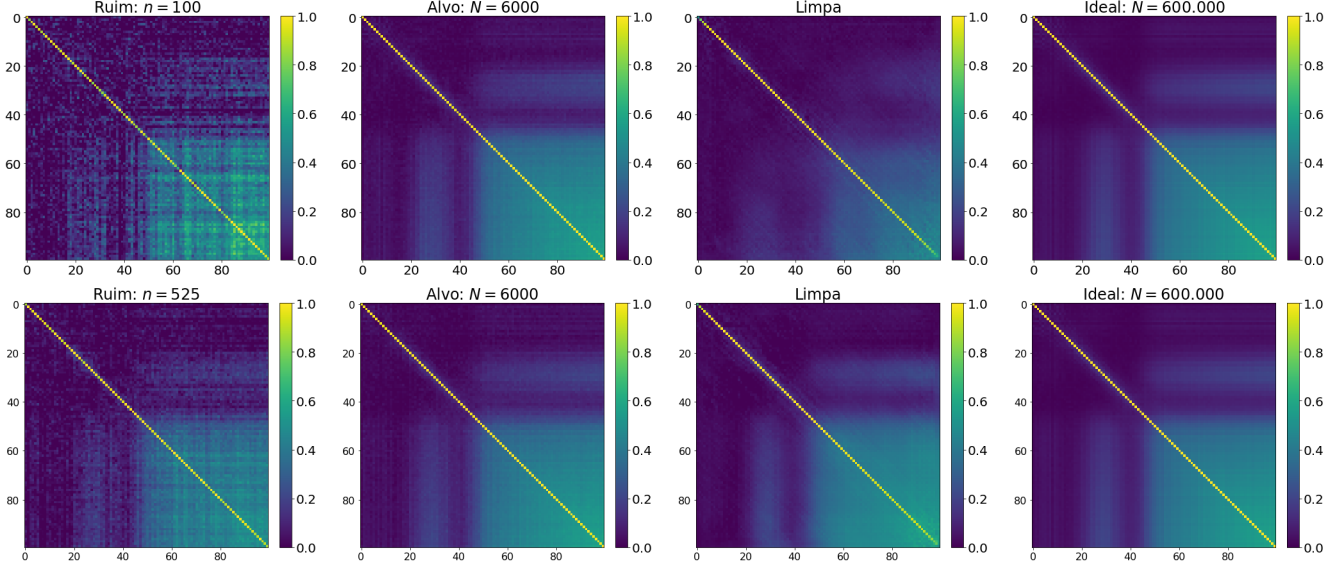


Figura 5: Comparação das matrizes de covariância normalizadas: na primeira coluna estão as matrizes ruins, com $n = [100, 525]$ espectros; na segunda coluna estão as matrizes alvo, com $N = 6000$; na terceira encontram-se as matrizes limpas, resultado do método; a quarta coluna contém as matrizes de covariância para o caso ideal ($N = 600.000$). Por questões de comparação a barra de cores entre as diferentes matrizes considera zero e um, respectivamente, como os valores mínimos e máximos.

ideal, podemos verificar toda a complexa estrutura da mesma¹. Essa estrutura foi recuperada pelo método nas matrizes limpas, como pode ser visto na Figura 5. Na mesma, temos a comparação das matrizes de covariância normalizadas para as matrizes ruins, com $n = [100, 525]$ espectros; as matrizes alvo, com $N = 6000$; as matrizes limpas e as matrizes de covariância para o caso ideal ($N = 600.000$). Para a comparação apresentada, a barra de cores entre as diferentes matrizes considera valores entre zero e um. Portanto, fica claro como as matrizes ruins perderam seu ruído, estão visualmente limpas e mais suaves. Além disso, tais matrizes estão muito mais próximas das matrizes alvo e, conseqüentemente, das matrizes ideais.

A performance dos modelos analisados foi medida utilizando a mesma métrica da função perda, a MSE, aqui escrita em função das matrizes de covariância como

$$MSE = \frac{1}{N} \sum_{l=1}^N \frac{1}{n_k^2} \sum_{i,j=1}^{n_k} (Cov_{ij}^l - \widehat{Cov}_{ij}^l)^2 \quad (5)$$

onde N representa o número de matrizes, n_k é o número de bins de k , Cov_{ij}^l as matrizes limpas e \widehat{Cov}_{ij}^l as matrizes alvo. Essa quantidade expressa o quão próximas as matrizes limpas estão das matrizes alvo. A mesma foi calculada não só para as matrizes de covariância, mas também para as matrizes de covariância normalizadas. Ainda por meio dela foi definido o que chamamos de “teste de sanidade”: que é o MSE entre as matrizes de covariância alvo e a ideal. Por isso, ao calcularmos o MSE entre as matrizes limpa e a matriz ideal, esse é o menor valor que poderemos obter para essa métrica.

Na Figura 6 encontram-se os resultados do MSE para as matrizes limpas com $n \in [100, 2500]$ espectros e matrizes alvo com $N = 6000$ espectros. O gráfico à esquerda mostra o MSE entre as matrizes de covariância normalizadas e as matrizes de covariância normalizadas alvo, enquanto o gráfico à direita contém o MSE entre as matrizes de covariância limpas e as matrizes ideais, bem como o “teste de sanidade” (a curva em preto é a MSE entre a alvo e a ideal) e o resíduo entre a primeira e a segunda quantidade, no sub-gráfico abaixo. Em ambas as análises o MSE diminui conforme aumenta-se n , praticamente estabilizando seu valor para $n \geq 525$. Notavelmente, o gráfico à direita oscila mais entre valores altos e baixos porque a normalização das matrizes foi recuperada, ou seja, temos o MSE diretamente para as matrizes de covariância e não sua versão normalizada, como à esquerda. Mesmo assim o valor médio do resíduo $\langle \Delta \rangle = 0.236$ é pequeno, mostrando que as matrizes limpas estão tão próximas da matriz ideal quanto as matrizes alvo estão.

¹É importante dizer que a normalização foi utilizada não só para efeitos visuais, mas para a convergência mais rápida e melhor do próprio método de aprendizado de máquina empregado.

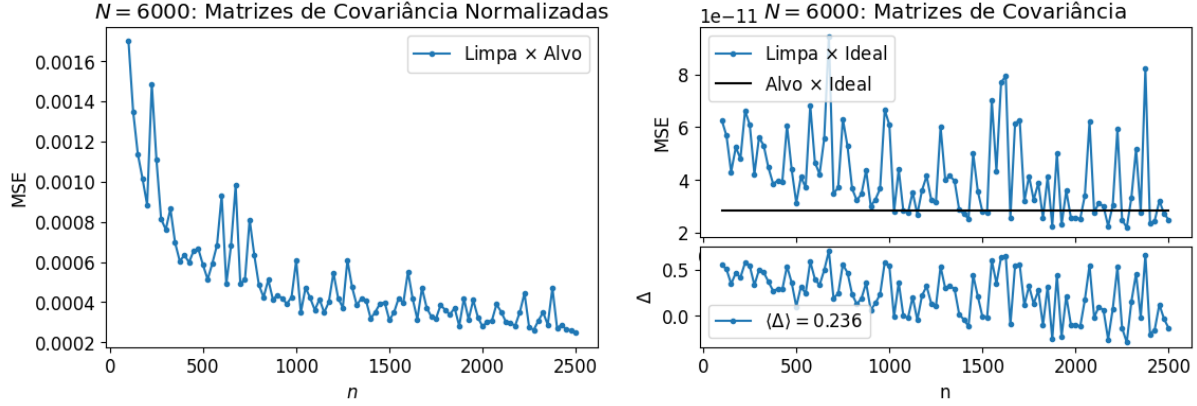


Figura 6: Comparação do MSE usando: as matrizes de covariância normalizadas (à esquerda) e as matrizes de covariância (à direita). Veja que o gráfico à direita apresenta também a curva preta, mostrando o que chamamos de teste de sanidade, i.e., o quão próximas estão as matrizes limpas da matriz ideal, tanto quanto a matriz alvo.

5 Recuperando os parâmetros

O teste final para a aplicação do aprendizado de máquina nas matrizes de covariância é dado com a sua aplicação estatística, em outras palavras, quando utilizamos-las para obter os parâmetros fiduciais da simulação em questão. Por isso, o interesse se dá no quão bem as matrizes limpas conseguem recuperar os parâmetros se comparadas as matrizes alvo, ruim e até mesmo com relação à matriz ideal. Para isso, foi feito uso do MCMC, um método numérico capaz de retornar uma amostra de pontos do espaço de parâmetros k -dimensional, com uma distribuição para cada parâmetro neste espaço, fazendo uso do teorema de Bayes [14].

O cálculo dos parâmetros fiduciais (P_0, k_0, σ_0) foi feito para diferentes matrizes limpas obtidas com diferentes valores n . Dessa forma, a medida da fração entre os desvios das distribuições com relação ao desvio da estimativa dos mesmos para a matriz ideal σ_X/σ_{Ideal} , para cada um dos parâmetros (com $X \in [P_0, k_0, \sigma_0]$) e para cada uma das matrizes (ruim, limpa e alvo) em função de n pode ser vista na Figura 7. A curva em cinza, valendo um, representa $\sigma_{Ideal}/\sigma_{Ideal}$. Dado que a região em destaque (cinza) compreende um intervalo pequeno (5 ou 7.5 %) a partir do qual as medidas ficam estáveis, podemos dizer que não necessitamos passar de mais de $n \simeq 525$ para alcançarmos uma matriz de covariância tão precisa quanto a alvo.

Uma outra quantidade capaz de indicar o menor n é dada pelo cálculo da fração do volume das covariâncias, valendo

$$V_X = \sqrt{\det[Cov_X(P_0, k_0, \sigma_0)]}, \quad (6)$$

onde X representa a matriz a ser considerada (essa análise é similar à feita na Referência [15]). Como vimos pelos resultados visuais a matriz limpa apresenta-se bem próxima da matriz ideal, logo, a fração de interesse é dada tomando V_X/V_{Ideal} . Os resultados, então, estão na Figura 8. A conclusão da escolha de $n = 525$ se repete aqui: além de indicar, numericamente, a mesma conclusão visual de que as matrizes obtidas limpas são próximas até mesmo da matriz ideal.

Por fim, tomando a escolha de $n = 525$ para a matriz ruim, com a matriz alvo de $N = 6000$ espectros e comparando com a matriz ideal, a comparação para a obtenção dos parâmetros fiduciais pode ser vista na Figura 9. Conjuntamente, podemos calcular a quantidade

$$\frac{\Delta\mu}{\sigma_Y} = \frac{\mu_X - \mu_Y}{\sigma_Y} \quad (7)$$

com os parâmetros $X \in [P_0, k_0, \sigma_0]$ no caso das matrizes $Y \in [Alvo, Ideal]$, apresentada na Tabela 1. Portanto, concluímos que as matrizes limpas estimam os parâmetros muito mais precisamente se comparadas as matrizes ruins. Tal melhora² chega a valores da ordem de $\sim 13\times$ para P_0 (no caso da matriz ideal), $\sim 4\times$ para k_0 (no caso da matriz alvo) e $\sim 12\times$ para σ_0 (no caso da matriz alvo).

6 Discussão e conclusões

A técnica de remoção de ruídos para imagens apresentou excelentes resultados quando aplicada para remover ruídos das matrizes de covariância que chamamos como ruins. Primeiramente, mostrou-se que matrizes ruins com apenas $n = 100$ espectros já

²Por melhora indicamos o número de vezes que as frações das matrizes limpas são menores se comparadas as frações das matrizes ruins.

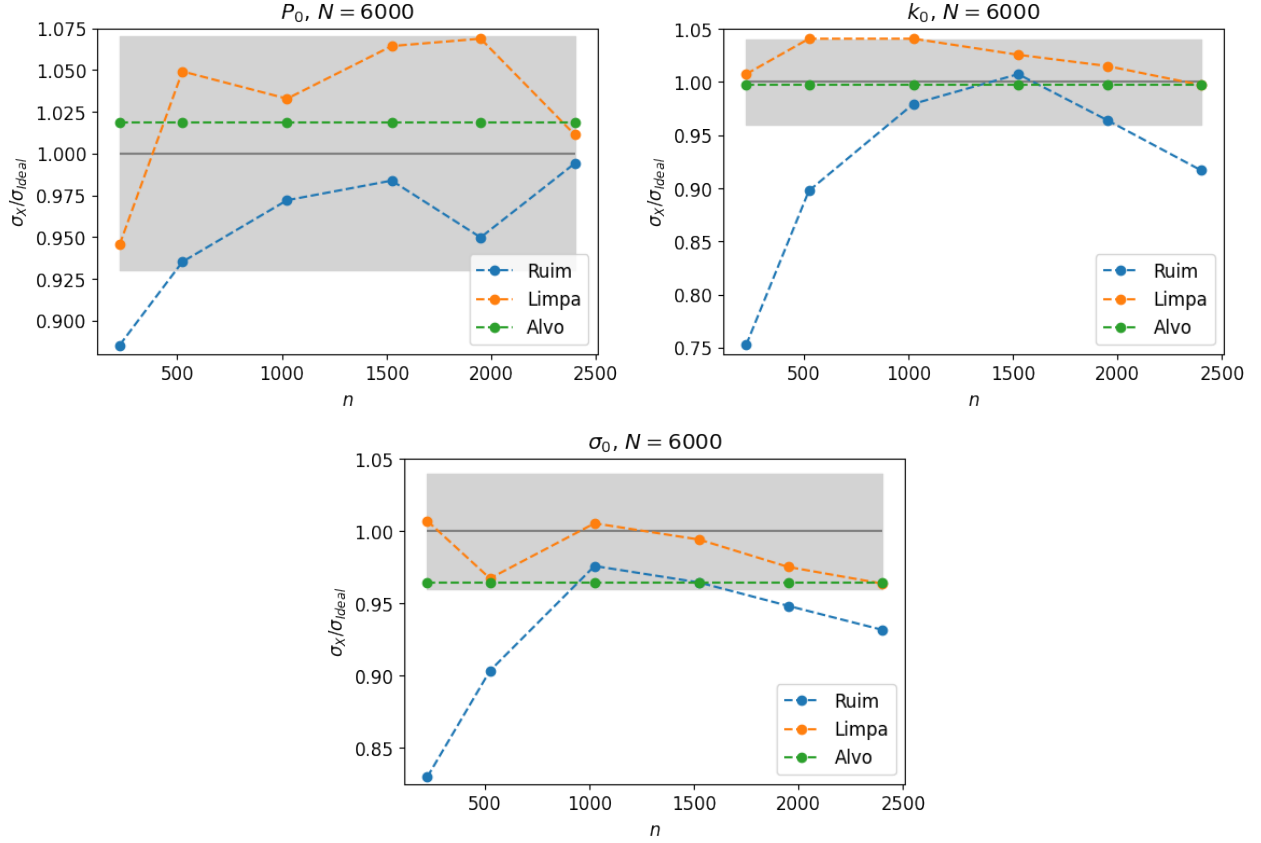


Figura 7: Fração: σ_X/σ_{Ideal} , para $X \in [P_0, k_0, \sigma_0]$ comparando os resultados obtidos utilizando as matrizes ruim, limpa e alvo para matrizes de covariância obtidas com diferentes números de espectro $n \in [100, 2500]$ e matriz alvo com $N = 6000$. A região em destaque (cinza) indica as regiões de 5 ou 7.5 % de intervalo máximo considerável para o caso da matriz limpa obtida.

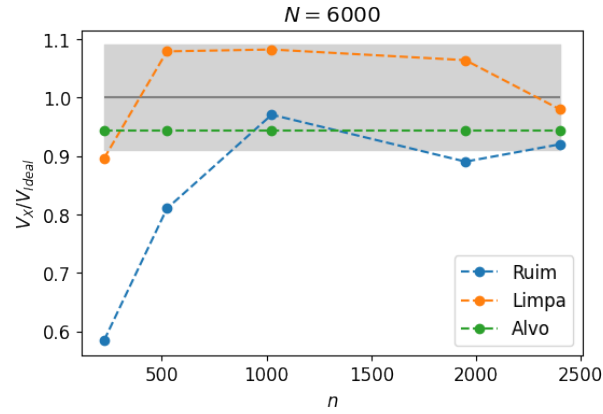


Figura 8: Fração de volume V_X/V_{Ideal} , com $X \in [Ruim, Limpa, Alvo]$ obtida para matrizes ruins com $n \in [100, 2500]$ e matriz alvo $N = 6000$.

Tabela 1: Valores absolutos da razão dos desvios da média $\Delta\mu$ e desvios padrões σ para cada uma das distribuições da matriz limpa (obtida de matrizes ruins com $n = 525$) relacionadas aos valores da matriz alvo ($N = 6000$) e ideal ($N = 600.000$).

$\Delta\mu/\sigma_{Alvo}$	Ruins	Limpa	$\Delta\mu/\sigma_{Exc}$	Ruins	Limpa
P_0	0.603	0.124	P_0	0.685	0.054
k_0	0.250	0.061	k_0	0.218	0.092
σ_0	0.227	0.019	σ_0	0.118	0.120

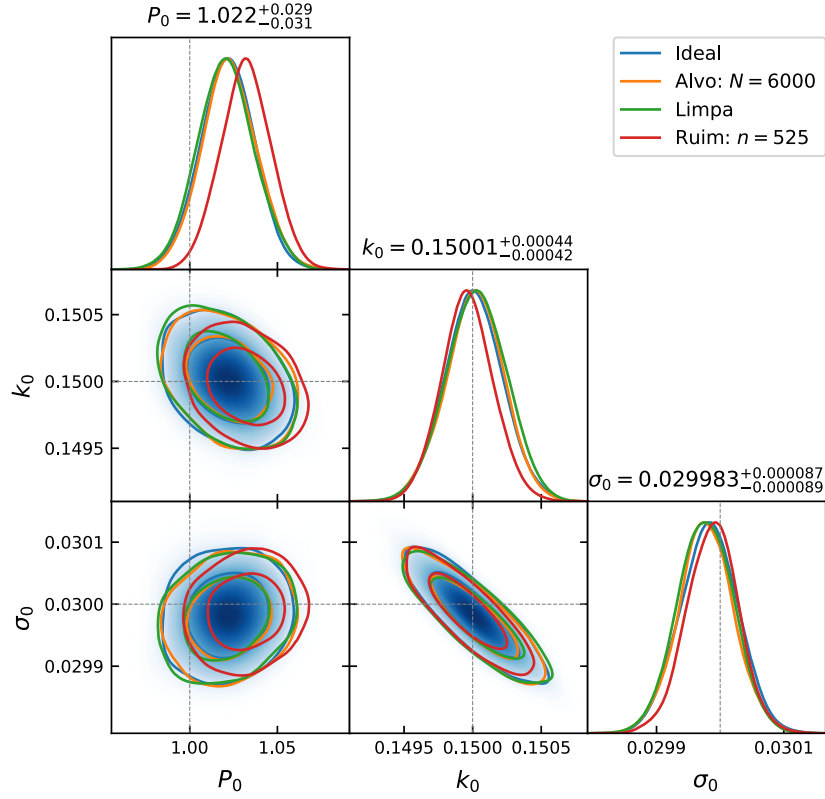


Figura 9: Gráfico de triângulos mostrando a obtenção dos parâmetros fiduciais (P_0, k_0, σ_0) para as matrizes ideal ($N = 600.000$), limpa, ruim ($n = 525$) e alvo ($N = 6000$).

ficam muito próximas das matrizes alvo com $N = 600.000$ espectros. Esses resultados ficam ainda melhores quando utilizamos matrizes ruins com $n = 525$ espectros, tornando nítido que as matrizes limpas, obtidas pelo método, são inclusive muito parecidas com as matrizes ideais, com $N = 600.000$ espectros.

As matrizes limpas obtidas foram comparadas em valores numéricos às matrizes alvo e a matriz ideal em sua versão normalizada e com suas unidades utilizando a quantidade MSE. Ambos os resultados apresentaram-se em perfeita acórdância, mostrando a não necessidade, devido a diminuição dessa quantidade (e a baixa variação de seu valor), de obtermos matrizes com n maior que 525 para alcançarmos uma matriz de covariância precisa.

Os principais resultados foram dados com a recuperação dos parâmetros fiduciais, fazendo uso do método MCMC. Os resultados das frações da variância dos parâmetros, bem como do volume dos mesmos foram capazes de confirmar as premissas dos resultados anteriores, mostrando o quão bem estimamos os parâmetros com as matrizes limpas. Finalmente, fizemos uso da matriz ruim contendo apenas $n = 525$ espectros, comparando a mesma às matrizes ruim, alvo e ideal, no gráfico de triângulos para os três parâmetros, apresentado na Figura 9. Ou seja, as matrizes limpas mostraram-se tão boas quanto as matrizes alvo e, até mesmo a matriz ideal.

Os resultados se mostraram consistentes em todas as análises. Portanto, conclui-se que com apenas um pequeno número de espectros o método removedor de ruídos apresentado é capaz de fornecer matrizes comparáveis a matrizes geradas com um grande número de espectros.

Os próximos passos deste trabalho incluem a aplicação desse método para matrizes de covariância realísticas, i.e., oriundas de simulações de N-corpos e simulações mais baratas como mapas de halos gerados por técnicas perturbativas e fazem parte do projeto de doutorado da primeira autora deste trabalho.

Agradecimentos

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), em especial processo [n° 2019/13108-0](#), pelo suporte financeiro.

Referências

- [1] The Dark Energy Survey Collaboration. The Dark Energy Survey. *arXiv e-prints*, pages astro-ph/0510346, October 2005.
- [2] N. Benitez, R. Dupke, M. Moles, and et. al. J-PAS: The Javalambre-Physics of the Accelerated Universe Astrophysical Survey. *arXiv e-prints*, page arXiv:1403.5237, March 2014.
- [3] Masahiro Takada, Richard S. Ellis, and Masashi et. al. Chiba. Extragalactic science, cosmology, and Galactic archaeology with the Subaru Prime Focus Spectrograph. *pasj*, (1):R1, February 2014.
- [4] DESI Collaboration, Amir Aghamousa, Jessica Aguilar, Steve Ahlen, and et. al. The DESI Experiment Part I: Science, Targeting, and Survey Design. *arXiv e-prints*, page arXiv:1611.00036, October 2016.
- [5] A. F. Heavens, E. Sellentin, D. de Mijolla, and A. Vianello. Massive data compression for parameter-dependent covariance matrices. *mnras*, 472(4):4244–4250, December 2017.
- [6] Scott Dodelson and Michael D. Schneider. The effect of covariance estimator error on cosmological parameter constraints. *Phys. Rev. D*, 88:063537, Sep 2013.
- [7] Oliver H. E. Philcox, Mikhail M. Ivanov, Matias Zaldarriaga, Marko Simonovic, and Marcel Schmittfull. Fewer mocks and less noise: Reducing the dimensionality of cosmological observables with subspace projections. *Phys. Rev. D*, 103:043508, Feb 2021.
- [8] Darsh Kodwani, David Alonso, and Pedro Ferreira. The effect on cosmological parameter estimation of a parameter dependent covariance matrix. *The Open Journal of Astrophysics*, 2(1):3, March 2019.
- [9] Linda Blot, Martin Crocce, Emiliano Sefusatti, Martha Lippich, Ariel G. Sánchez, Manuel Colavincenzo, Pierluigi Monaco, Marcelo A. Alvarez, and et. al. Comparing approximate methods for mock catalogues and covariance matrices II: power spectrum multipoles. *MNRAS*, 485(2):2806–2824, May 2019.
- [10] S. Saha. A comprehensive guide to convolutional neural networks - the eli5 way, 2020.
- [11] F. Chollet. *Deep Learning with Python*. Manning Publications Company, 2017.
- [12] François Chollet. keras. <https://github.com/fchollet/keras>, 2015.
- [13] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.
- [14] Ž. Ivezić, A.J. Connolly, J.T. VanderPlas, and A. Gray. *Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data*. Princeton Series in Modern Observational Astronomy. Princeton University Press, 2014.
- [15] Linda Blot, Martin Crocce, Emiliano Sefusatti, Martha Lippich, Ariel G. Sánchez, Manuel Colavincenzo, Pierluigi Monaco, Marcelo A. Alvarez, Aniket Agrawal, Santiago Avila, Andrés Balaguera-Antolínez, Richard Bond, Sandrine Codis, Claudio Dalla Vecchia, Antonio Dorta, Pablo Fosalba, Albert Izard, Francisco-Shu Kitaura, Marcos Pellejero-Ibanez, George Stein, Mohammadjavad Vakili, and Gustavo Yepes. Comparing approximate methods for mock catalogues and covariance matrices II: power spectrum multipoles. *MNRAS*, 485(2):2806–2824, May 2019.