Compressão de dados

Big data & Astroinformática
Professor: Clécio de Bom

Aluno: Pedro Riba Mello

Compressão de PDF’s

- DES observará 300 milhões de galáxias;
- Se cada função densidade de probabilidade (PDF) possuir ~2.8 Kb precisaríamos de reservar 843 GB para armazená-las.
- Vamos comprimir as PDF's!

https://www.darkenergysurvey.org/the-des-project/overview/

As ferramentas

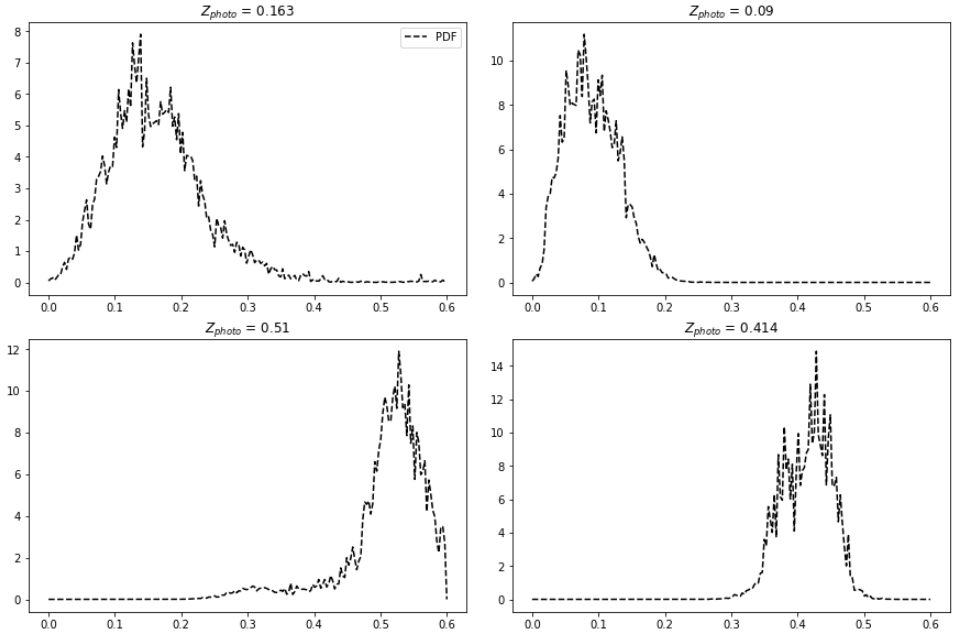
Vamos utilizar duas ferramentas para comprimir os dados:

1. Principal component analysis (PCA)
2. Auto-encoder

O PCA trata-se de uma projeção linear em espaço o de menor dimensão, já o auto-encoder é uma rede neural treinada para comprimir e descomprimir os dados.

Os dados utilizados

- Possuímos PDF's de redshift fotométrico para 20434 galáxias.
- Cada PDF possui 200 bins.
- Os dados são organizados em uma matriz (20434, 200).
- O arquivo das PDF's possui ~57 Mb, cada PDF possui ~2.8 Kb.



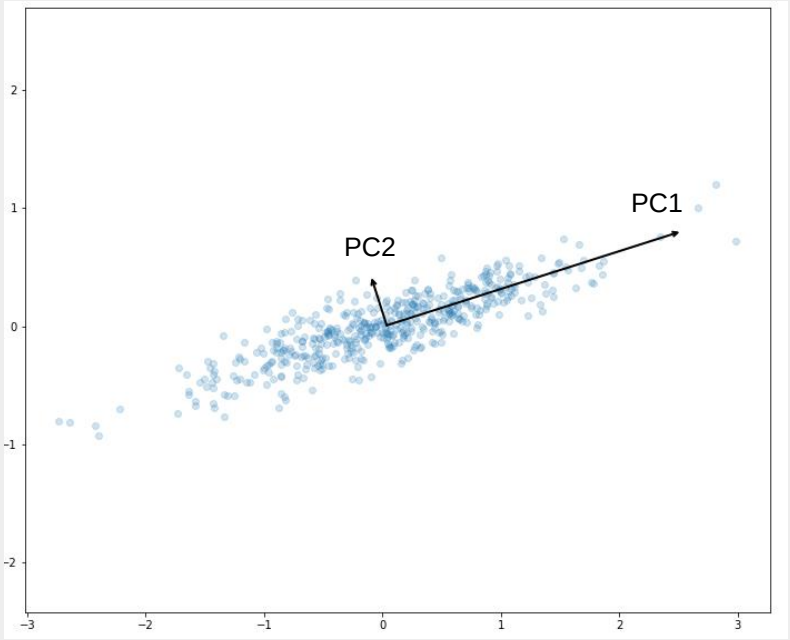
Principal component analysis

- O método PCA busca uma transformação linear R nos dados, cujos autovetores (componentes principais, ou PC) são os eixos de maior variância:

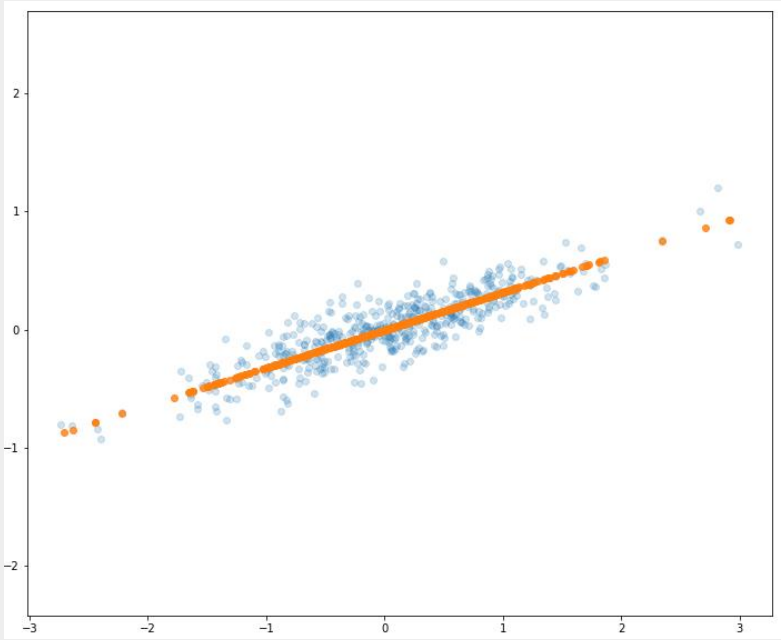
$$Y = XR$$

- Ao reduzir a dimensão dos dados para sua representação comprimida perdemos informação.
- A reconstrução não é perfeita!

Um exemplo em duas dimensões

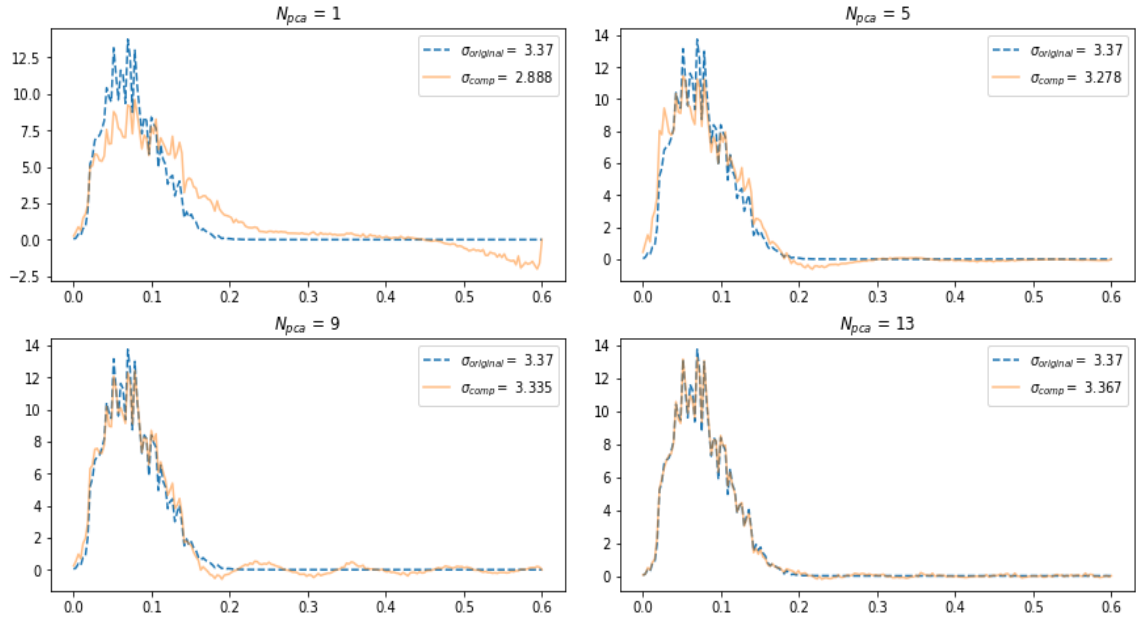


Projeção
em PC1



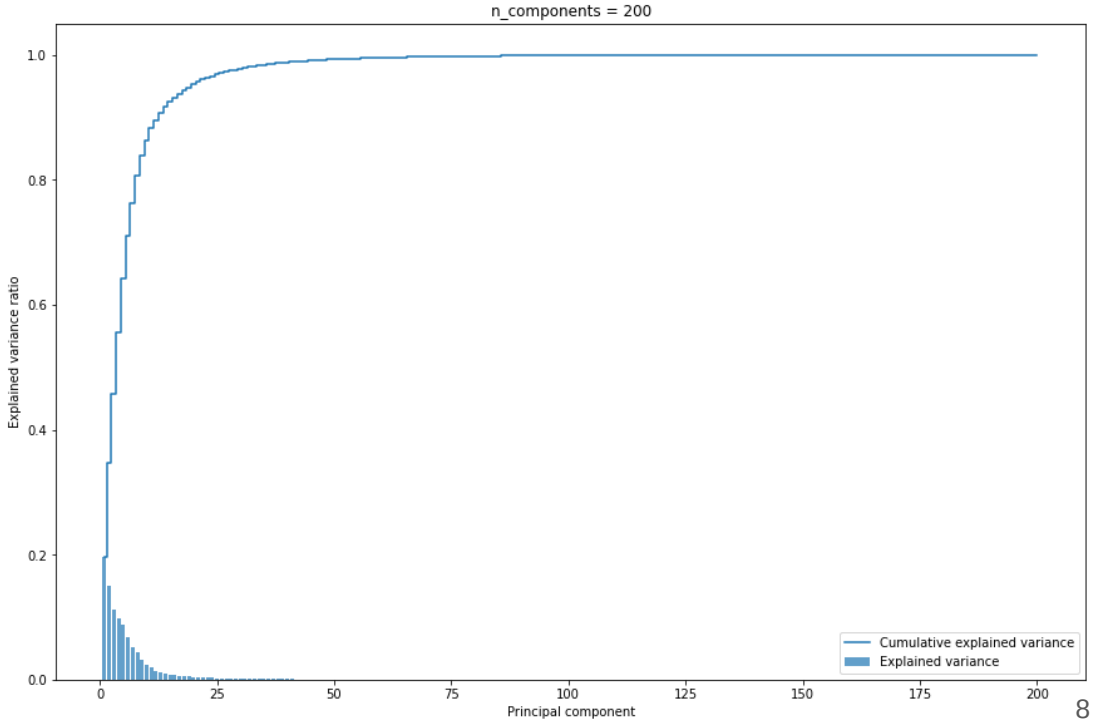
Reconstruindo os dados com PCA

- Conforme utilizamos mais componentes PCA melhor fica a nossa reconstrução.
- Para uma reconstrução perfeita precisaríamos da mesma dimensão dos dados originais.



Explained variance

- Podemos usar a soma das variâncias das componentes principais para medir a qualidade da compressão.
- Com 20 componentes principais mantemos 95.3% da variância total dos dados.
- Para atingir 99.3 % de “informação” precisamos de 50 componentes.



PCA em python

```
from sklearn.decomposition import PCA
import pandas as pd

n_components=20

pca=PCA(n_components=n_components)
pca.fit(PDF)

compressed=pca.transform(PDF)
df = pd.DataFrame(compressed)
df.to_csv('Compressed_data.csv')

decomp=pca.inverse_transform(compressed)
```

Número de componentes PCA.

Encontra a transformação R para os dados.

Comprime os dados.

Salva os dados comprimidos.

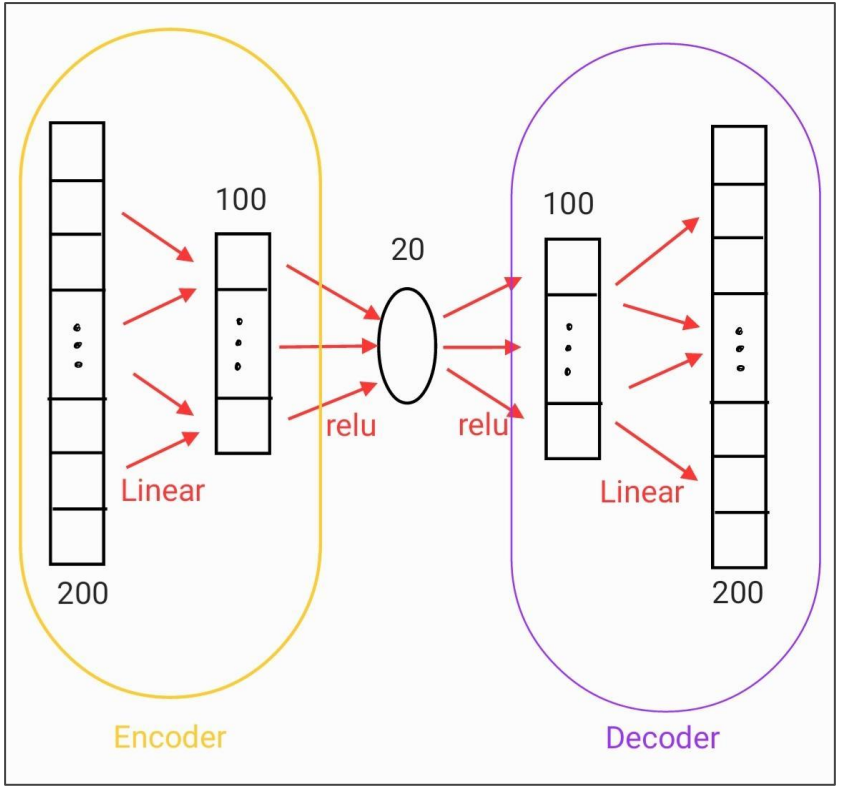
Reconstrói os dados.

Auto-encoder

- Um auto-encoder é uma rede neural constituída por três partes, um encoder, um núcleo e um decoder.
- O encoder é responsável por codificar os dados em uma representação comprimida.
- O núcleo, ou espaço latente, é a representação comprimida dos dados.
- O decoder é responsável por decodificar os dados e obter uma representação próxima dos dados originais, porém, com alguma perda.

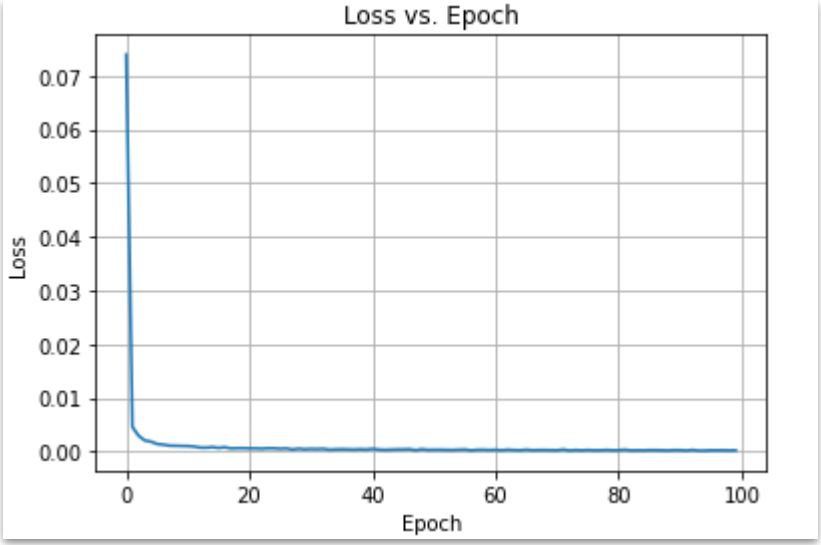
Auto-encoder

- A camada de ativação linear captura o comportamento geral dos dados.
- A camada relu captura as variações menores nos dados.
- Escolhemos 20 dimensões para os dados comprimidos, assim poderemos comparar com o PCA.



Treinando o auto-encoder

- Antes de treinar o auto-encoder, escalamos os dados de forma que eles possuam média 0 e variância 1.
- Treinamos o auto-encoder para um tipo específico de dados, ou seja, ele não será capaz de codificar outro tipo de PDF!
- Queremos comprimir um tipo específico de dado, não realizaremos nenhuma previsão. Por isso, não nos importaremos com overfitting!



Loss: mse
Optimizer: Adam

Os dados comprimidos

- Tamanho do arquivo original = 57.483 Kb
- Dimensão do arquivo comprimido = 20

	PCA	Autoencoder
Descompressor	34 Kb	558 Kb
Arquivo comprimido	7855 Kb	3064 Kb
Taxa de compressão	13.72 %	6.2 %
DES	115 GB	52 GB

- Além do tamanho, vamos comparar também a qualidade da compressão. Para isso, vamos utilizar dois critérios distintos, o BIC e a entropia relativa.

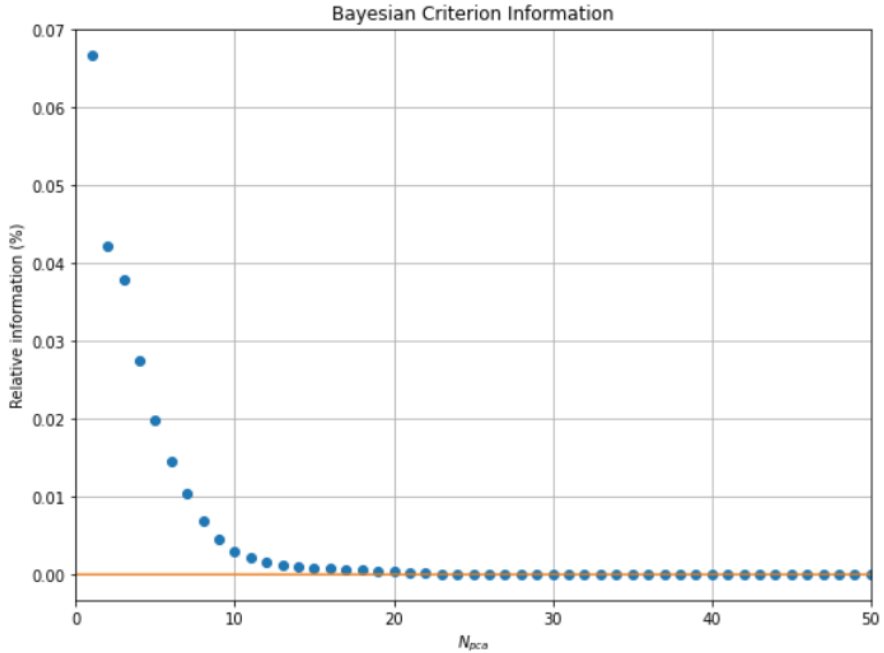
PCA ou auto-encoder?
Bayesian Criterion Information (BIC)

- O critério de informação bayesiana é definido como:

$$\mathrm{BIC} = k \cdot \ln(n) - 2\ln(\hat{L})$$

- Na equação k é o número de dimensões, n é o número de dados e L é o estimador de máxima likelihood.
- No gráfico ao lado, para o PCA:

$$\frac{BIC_{comp} - BIC_{orig}}{BIC_{orig}} \cdot 100\%$$

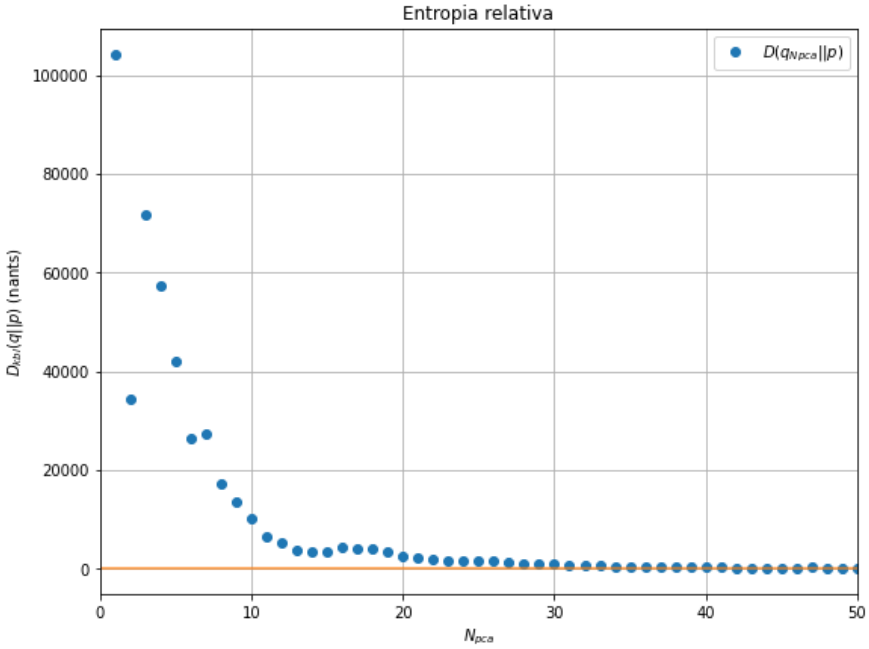


PCA ou auto-encoder?
Entropia relativa

- Podemos utilizar a divergência de Kullback-Leibler (ou entropia relativa) para medir o quão boa é a compressão:

$$D(p||q) = \int_{-\infty}^{\infty} p(x) \log[p(x)/q(x)]dx$$

- A entropia relativa é nula quando duas distribuições são iguais.
- Para a análise de PCA, a entropia relativa converge para 0 com 200 componentes, como era de se esperar!

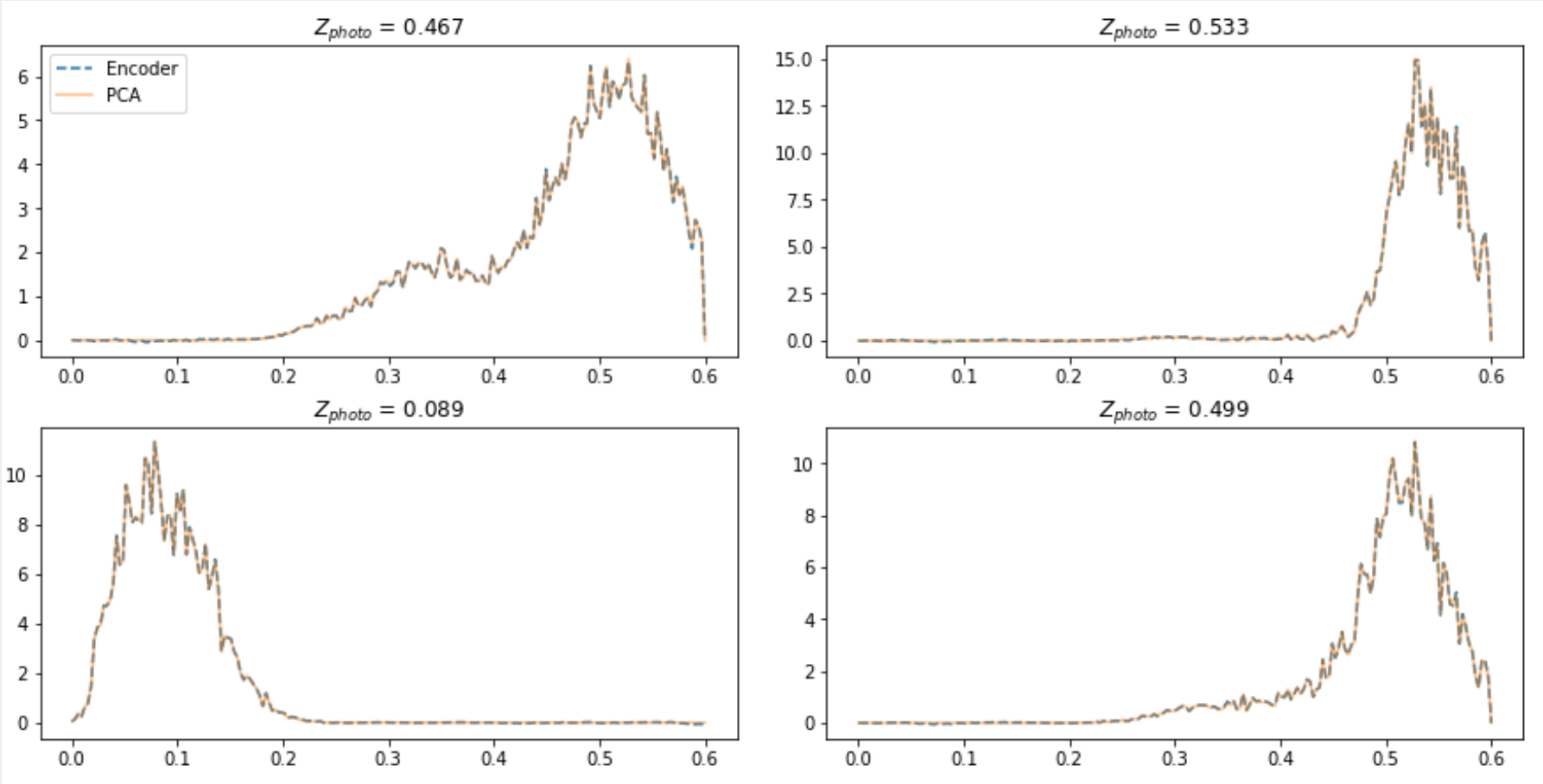


PCA ou auto-encoder? Comparando

- Abaixo vemos uma comparação direta entre o auto-encoder e o PCA. A dimensão do espaço latente utilizada é N=20.

	PCA	Auto-encoder
BIC relativo	4.1×10^{-4} %	8×10^{-5} %
Entropia relativa	1437.4 nants	1176.5 nants
Tamanho final	7889 Kb	3622 Kb

- Vemos que o auto-encoder é uma opção melhor que o PCA para compressão, pois o tamanho final do arquivo é menor e os parâmetros BIC e entropia relativa são menores.


Z_{photo} = 0.467
Encoder
PCA
Z_{photo} = 0.533
Z_{photo} = 0.089
Z_{photo} = 0.499

FIM
Obrigado!