Redução de dimensionalidade em dados de fMRI usando Python - Explorando o pacote BrainIAK

Paulo Cardoso

***Resumo*—Este trabalho tem por objetivo apresentar a exploração do pacote BrainIAK como primeiro contato de uma pessoa de tecnologia com foco em dados com o campo da neurociência, mais especificamente análise dados de imagem por ressonância magnética funcional, referida por sua sigla em inglês fMRI (functional magnetic resonance imaging).**

Palavras-Chave—BrainIAK, Neurociência, fMRI.

***Abstract*—This work aims to present the exploration of the BrainIAK package as the first contact of a technology person with a focus on data with the field of neuroscience, more specifically data analysis of functional magnetic resonance imaging, referred to by its acronym fMRI.**

Keywords—BrainIAK, Neuroscience , fMRI.

I. Contextualização do problema

A utilização de técnicas de análise de dados de fMRI vêm se popularizando e evoluindo nas últimas duas décadas, o que era considerado impossível, hoje não é mais. Um exemplo disso seria a pesquisa realizada por Friston et al, 1995, aplicando uma simples regressão linear para descrever as alterações na atividade cerebral voxel por voxel até os dias atuais onde temos possibilidade da utilização de computação de alta performance bibliotecas de software especializadas nesse tipo de domínio. (ELLIS et al, 2020).

Para execução do trabalho prático com os dados foi utilizado a linguagem de programação Python 3.7 por meio da distribuição Anaconda (https://anaconda.com) com o pacote de análise de imagens neurais BrainIAK (https://brainiak.org/). Os criadores do pacote BrainIAK Kumar et al, (2020a) e Kumar et al, (2020b) apresentam o mesmo como ?BrainIAK pode ser visto como um "canivete suíço" para análise avançada de fMRI?, por estender e se utilizar métodos presentes em outros pacotes como Nilearn e Scikit-learn otimizando a performance dos mesmos.

O Voxel, objeto base para a análise pode ser entendido como o volume formado pelo pixel e pela profundidade do corte, que representa uma unidade (dado pontual) de um grid retangular tridimensional, cada dado pontual destes é um pixel que representa a imagem 3D gerada pelo exame fMRI. Para cada grid retangular tridimensional resultado de uma rodada de exame fMRI contém centenas de milhares de voxels.

Cada um desses voxels pode ser entendido como uma feature e o número de features (voxels) supera em muito o número de observações (tamanho da amostra) acarretando um problema de dimensionalidade. Esse problema de dimensionalidade é conhecido como a maldição da dimensionalidade e de acordo com MWANGI expõe ao risco de overfitting do modelo, a solução é reduzir a dimensionalidade da amostra pré-selecionando as features mais relevantes, removendo ruído variáveis redundantes.

A. Covariância

Sendo a covariância um precursor para a compreensão das técnicas de redução de dimensionalidade e que pode ser entendida como a medida do grau de interdependência e inter-relação entre duas variáveis aleatórias. A covariância entre duas variáveis pode ser calculado como exemplificado pela fórmula apresentada na Figura 1.

$$Cov(X, Y) = \frac{\sum_1^N (X - \bar{X})(Y - \bar{Y})}{(N - 1)}$$

Tratando-se da aplicação da covariância em análise de dados de fMRI, X e Y podem ser dados de série temporal para dois voxels ou o padrão entre voxels para dois pontos de tempo diferentes.

B. PCA - Principal Component Analysis

Análise de componente principal - PCA é uma técnica de redução de dimensionalidade comumente utilizada para auxiliar a interpretação de grandes conjuntos de dados preservando a informação. Essa redução de dimensionalidade pode ser feita de duas formas, por eliminação ou extração. Alguns voxels podem conter informações correlacionadas, portanto, a matriz de dados voxel-dimensional original pode ser projetada em um espaço de matriz de "componente" de dimensão inferior sem perder muitas informações.

II. Metodologia

Para realização do experimento foram executadas as seguintes etapas: A) Criação e configuração do ambiente de desenvolvimento, B) Pré- Projeto e Aquisição dos dados, C) Experimento e D) Reprodutibilidade.

A. Criação e configuração do ambiente de desenvolvimento

Após análise dos pré requisitos necessários para utilização do pacote BrainIAK, optou-se por utilizar uma Máquina Virtual (VirtualBox) local com uma imagem de Ubuntu 20.04, em cima desta imagem foram instalados a distribuição de software Anaconda 3 e por fim o pacote BrainIAK com todas suas dependências.

B. Pré- Projeto e Aquisição dos dados

Com o ambiente instalado e configurado, o passo seguinte foi entender um pouco mais sobre fMRI e o básico sobre análise de imagens neurais por meio dos tutoriais disponibilizados pelo pacote em (https://brainiak.org/tutorials/). Os dados utilizados neste trabalho, foi o conjunto de dados "localizer" disponibilizado por KIM et al, conjunto de dados esse pré-processado. Em fMRI conjuntos de dados pré-processados são usualmente armazenados em formato NIfTI.

O conjunto de dados localizer consiste em 3 corridas com 5 blocos de cada categoria (rostos, cenas e objetos) por corrida. Cada bloco foi apresentado por 15s. Dentro de um bloco, um estímulo era apresentado a cada 1,5s (1 TR). Entre os blocos, houve 15s (10 TRs) de fixação. Cada corrida foi de 310 TRs. No arquivo de estímulo MATLAB, os primeiros códigos de linha para a categoria de estímulo para cada tentativa (1 = Faces, 2 = Cenas, 3 = Objetos). A 3ª linha contém o tempo (em segundos, relativo ao início da execução) quando o estímulo foi apresentado para cada tentativa.

C. Experimento

O experimento foi realizado em sessões, I) Setup e apresentação dos dados, II) Classificação e III) Redução de dimensionalidade. Para o objetivo deste texto apenas as seções I e III serão apresentadas em detalhes.

A seção I) Setup e apresentação dos dados, responsável pelo carregamento de dados, pacotes e biblioteca de apoio, bem como pela limpeza e preparação dos dados. Já a seção III) Redução de dimensionalidade, é onde ocorre a análise de covariância, PCA e por fim a seleção de componentes por meio de validação cruzada. Estas etapas como o nome descreve de forma direta são a execução dos passos listados.

D. Reprodutibilidade

Por fim, a reprodutibilidade, que é um dos pontos mais importantes quando disponibilizando uma análise ao público. No caso deste trabalho, é disponibilizado todos os códigos fonte utilizados além do dado base da análise.

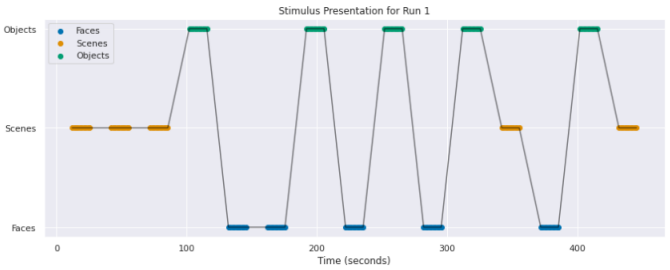
- Repositorio de codigo: https://github.com/cardosop/BrainIAK
- Jupyter Notebook da Analise: https://github.com/cardosop/BrainIAK/blob/main/PauloCardoso-234956-TrabalhoFinalFt043.ipynb
- Biblioteca suporte para analise, utils.py: https://github.com/brainiak/brainiak-tutorials/blob/master/tutorials/utils.py
- Conjunto de Dados: https://drive.google.com/file/d/1ZglrmkYw8isBAfsL53n9JgHEucmrnm4E/view

III. RESULTADOS

De forma geral pode se apresentar o resultado final da seleção componentes por validação cruzada como o principal resultado, porém existem resultados intermediários que levam aos elementos selecionados que são muito relevantes de serem apresentados e discutidos. Para a apresentação dos resultados será utilizado com referência o Jupyter Notebook utilizado na análise, exemplificando seus passos enumerados e explicando os mais importantes.

Os primeiros passos da Sessão I (A e B) do experimento são marcados pela importação de pacotes e/ou bibliotecas, seguido pela apresentação de algumas constantes extraídas do conjunto de dados por meio de métodos customizados importados da biblioteca utils.py apresentada anteriormente. O passo seguinte o C apresenta a estrutura de diretórios/arquivos de dados que compõem a amostra e é de suma importância para entender como navegar na árvore de diretórios.

Já na etapa D, o desenho do experimento é demonstrado por meio de um gráfico que apresenta o estímulo recebido pelo objeto do exame na linha do tempo das 3 "Runs". Como pode ser observado na Figura 2.



Dos dados utilizados neste trabalho, parte deles representam duas regiões cerebrais de interesse conhecidas como ROI Regions Of Interest e são elas FFA e PPA. O dado que representa a região FAA é carregado no bloco de código E, para que no bloco F a intensidade do voxel fosse apresentada em série temporal. E por fim, nos blocos F, G e H ocorre uma normalização dos dados utilizando zscore e a apresentação da comparação entre o dado normalizado e não normalizado para encerrar a seção I.

Na seção III) Redução de dimensionalidade, com o dado limpo e preparado da etapa anterior um novo conjunto de variáveis é criado no bloco A, para que no bloco seguinte o B ocorra uma análise da covariância entre dois blocos. Já no bloco C é calculada a correlação entre esses mesmos blocos, como resultado é obtida a Covariância de -0.03511234506686155 e uma correlação de -0.04726873436910835.

No bloco a seguir o C, o PCA é calculado dem normalização do dado e apresentado por meio de scatter plot em 3 medidas de dimensões diferentes para propiciar comparação, Figura 3.

Na Figura 4 acima da primeira e da segunda dimensão do PCA, pode-se observar três clusters e então presumir que eles

correspondem a rostos, cenas e objetos. O bloco R apresenta a diferenciação dos elementos dos clusters por categoria tempo, demonstrando que existe uma maior similaridade nos componentes dos clusters na PCA por tempo.

O bloco G tem uma elevada importância por explorar como o PCA afeta a assertividade de uma classificação SVC e o que pode-se observar é que com 15 componentes a assertividade é de 100%, conforme apresenta a Figura 4 que é o output do n-fold.

```
Components: 5 Accuracy: [0.8666666666666667, 0.9333333333333333, 0.8]
Components: 10 Accuracy: [1.0, 0.8666666666666667, 1.0]
Components: 15 Accuracy: [1.0, 1.0, 1.0]
Components: 20 Accuracy: [0.8, 1.0, 0.9333333333333333]
Components: 25 Accuracy: [0.8666666666666667, 1.0, 1.0]
```

Para finalizar, foi realizada a validação cruzada para seleção de variáveis no bloco H, que apresenta uma média de assertividade na base de teste de 91% conforme apresentado no output do bloco e na Figura 5.

```
CV iteration: 0
Train_index:
[15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
 39 40 41 42 43 44]
Test_index:
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14]
CV iteration: 1
Train_index:
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 30 31 32 33 34 35 36 37 38
 39 40 41 42 43 44]
Test_index:
[15 16 17 18 19 20 21 22 23 24 25 26 27 28 29]
CV iteration: 2
Train_index:
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29]
Test_index:
[30 31 32 33 34 35 36 37 38 39 40 41 42 43 44]
{'fit_time': array([0.06934834, 0.01795435, 0.01649547]), 'score_time': array([0.0011282 ,
0.00102758, 0.00109005]), 'test_score': array([0.86666667, 0.93333333, 0.93333333]), 'trai
n_score': array([1., 1., 1.])}
Average Testing Accuracy: 0.91
```

IV. CONCLUSÕES

O pacote BrainIAK é uma excelente ferramenta, mesmo tendo explorado parte reduzida das suas capacidades posso afirmar que é uma ferramenta que potencializa o trabalho de análise de dados de fMRI.

REFERÊNCIAS

[1] KUMAR, Manoj et al. BrainIAK tutorials: User-friendly learning materials for advanced fMRI analysis. PLoS computational biology, v. 16, n. 1, p. e1007549, 2020.

[2] KUMAR, Manoj et al. BrainIAK: The brain imaging analysis kit. 2020.

[3] ELLIS, Cameron T. et al. Facilitating open-science with realistic fMRI simulation: validation and application. PeerJ, v. 8, p. e8564, 2020.

[4] FRISTON, Karl J. et al. Analysis of fMRI time-series revisited. Neuroimage, v. 2, n. 1, p. 45-53, 1995.

[5] MWANGI, Benson; TIAN, Tian Siva; SOARES, Jair C. A review of feature reduction techniques in neuroimaging. Neuroinformatics, v. 12, n. 2, p. 229-244, 2014.

[6] JOLLIFFE, Ian T.; CADIMA, Jorge. Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, v. 374, n. 2065, p. 20150202, 2016.

[7] KIM, Ghootae; NORMAN, Kenneth A.; TURK-BROWNE, Nicholas B. Neural differentiation of incorrectly predicted memories. Journal of Neuroscience, v. 37, n. 8, p. 2022-2031, 2017.