



باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

۲۵۷۴۲ گروه ۴ - سیگنال‌ها و سیستم‌ها - بهار ۱۳۹۹ - ۱۴۰۰

## پروژه

موعد تحویل: مطابق CW

## نحوه‌ی تحویل:

- گزارش پروژه خود را در قالب یک فایل pdf. تحویل دهید. در گزارش لازم است تمامی خروجی‌ها و نتایج نهایی، پرسش‌های متن تمرین، و توضیح مختصری از فرآیند حل مسأله‌ی خود در هر قسمت را ذکر کنید.
- کد کامل تمرین را در قالب یک فایل m. تحویل دهید. لازم است بخش‌های مختلف پروژه در sectionهای مختلف تفکیک شوند و کد تحویلی منظم و دارای کامنت‌گذاری مناسب باشد. بدیهی است آپلود کردن کدی که به درستی اجرا نشود، به منزله‌ی فاقد اعتبار بودن نتایج گزارش شده نیز می‌باشد.
- توابعی را که (در صورت لزوم) نوشته‌اید، در قالب فایل‌های m. در کنار فایل‌های گزارش و کد اصلی پروژه، ضمیمه کنید.
- مجموعه‌ی تمامی فایل‌ها (گزارش، کد اصلی، توابع، و خروجی‌های دیگر در صورت لزوم) را در قالب یک فایل zip/.rar. ذخیره کرده و از طریق سامانه‌ی CW تحویل دهید.
- نام‌گذاری فایل‌های تحویلی را به صورت

Project\_StudentNumber\_StudentNumber.pdf/.m/.zip/.rar

انجام دهید.

## معیار نمره دهی:

- ساختار مرتب و حرفه‌ای گزارش
- استفاده از توابع و الگوریتم‌های مناسب
- پاسخ به سؤالات تئوری و توضیح روش‌های مطلوب سوال
- کد و گزارش خروجی کد برای خواسته‌های مسأله

## نکات تکمیلی:

- شرافت انسانی ارزشی به مراتب والاتر از تعلقات دنیوی دارد. رونویسی تمارین، زیر پا گذاشتن شرافت خویشتن است؛ به کسانی که شرافتشان را زیر پا می‌گذارند هیچ نمره‌ای تعلق نمی‌گیرد.
- در بخش‌هایی از این پروژه به ویژه بخش نهایی یادگیری ماشین قسمت‌های امتیازی وجود دارد و اگر این بخش را نتوانستید کامل و با جزئیات پیاده سازی کنید از لحاظ نمره در بخش‌های امتیازی پروژه محسوب خواهد شد.

در این پروژه ابتدا به مباحث تئوری تبدیل فوریه گسسته و نمونه‌برداری، تا حدی که در پروژه لازم است می‌پردازیم. بدیهی است که تمامی نکات ذکر نشده‌اند و به صورت کامل تر در درس و همچنین درس پردازش سیگنال‌های دیجیتال به آن‌ها پرداخته خواهد شد. همچنین توجه شود که در هر مرحله اگر در صورت پروژه از دیتا صحبتی شده است منظور تمامی دیتا‌هایی است که در اختیارتان قرار گرفته است. و بدیهی است که پردازش‌های اولیه باید روی تمامی دیتا‌ها انجام شود و سپس به بخش تشخیص کلمات پرداخته شود.

## نمونه‌برداری – Sampling

برای انجام داشتن معادل یک سیگنال پیوسته در دنیای دیجیتال کامپیوتر و انجام محاسبات بر روی آن، بایستی از سیگنال پیوسته نمونه برداری کنیم. در واقع کار تمامی سنسورها و ابزارآلات ذخیره‌سازی دیجیتال نیز همین نمونه‌برداری است.

نمونه‌برداری معمولاً با یک نرخ ثابت  $Fs$  هرگز انجام می‌پذیرد؛ یعنی در هر  $T = \frac{1}{Fs}$  ثانیه، از ورودی سنسور اطلاعات خوانده می‌شود و در حافظه ذخیره می‌شود (بایستی توجه شود که بعد از فرایند نمونه برداری، هیچ نشانی از  $Fs$  در خود داده‌ها وجود ندارد و به همین دلیل خود  $Fs$  بایستی به صورت جداگانه در گزارش‌ها ذکر شود و هر کجا که لازم باشد، به صورت دستی به عنوان ورودی سیستم داده شود).

مشابه تعداد نمونه‌هایی که در قسمت پیش برای نمونه‌برداری از تبدیل فوریه گسسته زمان دیدیم، برای نمونه برداری نیز شرطی وجود دارد که با برقرار بودن آن، سیگنال و تبدیل فوریه آن یکتا از روی یکدیگر قابل بازسازی هستند اما در صورتی که آن را رعایت نکنیم، حتماً اطلاعاتی را از دست خواهیم داد. رابطه‌ی حوزه‌ی زمان نمونه‌برداری به صورت زیر است:

$$x[n] = x_c(nT) \quad (1)$$

$x$  در معادله بالا، سیگنال گسسته زمان است و  $x_c$  همان سیگنال پیوسته است. می‌توان نشان داد که رابطه ۱ معادلی در حوزه فرکانس با رابطه زیر دارد:

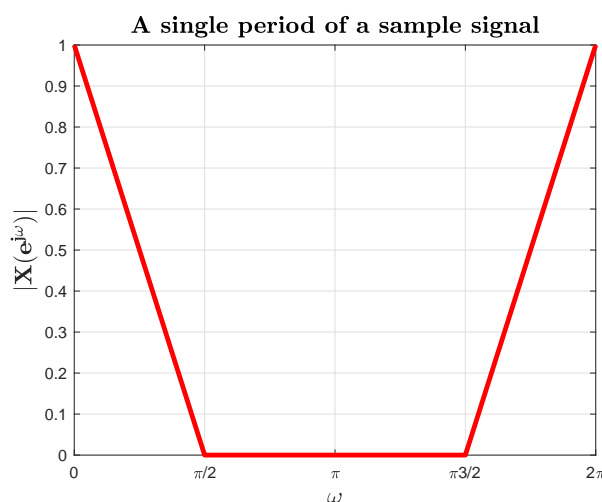
$$X(e^{j\Omega}) = \frac{1}{T} \sum_{r=-\infty}^{\infty} X_c(j(\frac{\Omega}{T} - \frac{2r\pi}{T})) \quad (2)$$

در واقع می‌توان به این دید به معادله بالا نگاه کرد که ابتدا طیف  $X_c$  را با  $T$  مقیاس می‌کنیم و سپس با  $2\pi$  متناوب می‌کنیم و نهایتاً در یک ضریب  $\frac{1}{T}$  ضرب می‌کنیم. با استفاده از این تعریف می‌خواهیم حد پایین فرکانس نمونه برداری را بیابیم. با توجه به فرمول بالا می‌توان دید که اگر  $\omega$  را فرکانس برای سیگنال پیوسته در نظر بگیریم و  $\Omega$  را برای فرکانس سیستم گسسته، رابطه زیر برقرار است:

$$\Omega = \omega T \quad (3)$$

یعنی با ضرب مقادیر فرکانس گسسته در فرکانس نمونه‌برداری، فرکانس معادل سیگنال پیوسته بدست می‌آید. همانگونه که قبلاً گفته شد، اطلاعات  $T$  در نمونه‌برداری از بین می‌روند؛ پس این‌که هر فرکانس  $\Omega$  نمایانگر چه فرکانس سیگنال پیوسته است، به فرکانس نمونه‌برداری وابسته است.

در شکل زیر یک نمونه پوش اندازه تبدیل فوریه یک سیگنال حقیقی فرضی داده شده است. همانگونه که مشاهده می‌کنید، شکل تقارن ذاتی دارد.



شکل ۱: تبدیل فوریه گسسته زمان سیگنال نمونه

- با توجه به توضیحاتی که در قسمت قبل در مورد تبدیل فوریه گسسته و تابع fft گفته شد و به کمک شکل ۱ تابعی بنویسید که با ورودی گرفتن سیگنال حقیقی، تبدیل فوریه آن را محاسبه کند و در خروجی برداری حاصل دهد که تنها فرکانس‌های  $[0, \pi]$  را داشته باشد. توجه کنید که می‌خواهیم انرژی سیگنالی که در خروجی ظاهر می‌شود، در رابطه پارسوال صدق کند. نمودار اندازه این بردار را بکشید در گزارش بیاورید. تابع شما بایستی با ورودی گرفتن فرکانس نمونه‌برداری، محور فرکانس را به درستی نمایش دهد. تابع شما بایستی به شکل زیر باشد:

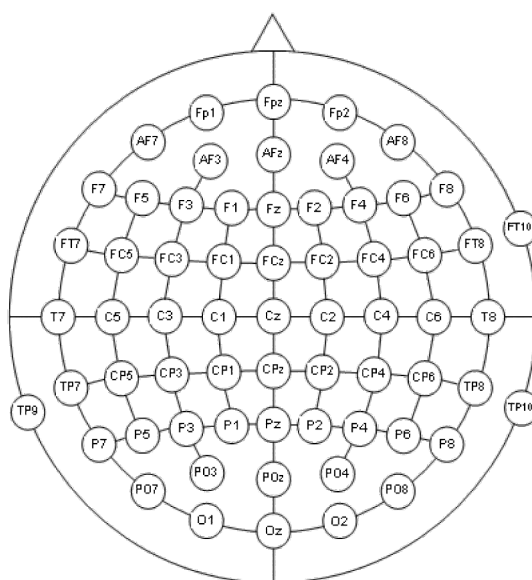
HalfBandFFT (InputSignal, Fs)

- در این سوال می‌خواهیم حد پایین فرکانس نمونه‌برداری را محاسبه کنیم. برای این کار فرض کنید که سیگنال شما حقیقی و پایین‌گذر است؛ یعنی بعد از فرکانسی مانند  $2\pi f_{max}$  محتوای فرکانسی ندارد. با توجه به نکاتی که تا کنون گفته شده از اینکه هر فرکانس گسسته نمایانگر چه فرکانس پیوسته‌ای است، نکات قسمت تبدیل فوریه گسسته و به کمک شکل ۱ حد فرکانس نمونه‌برداری را پیدا کنید که پهنای باند قسمت معادل  $[0, \pi]$  با پهنای باند متقارن خود که از نقطه‌ی  $2\pi$  شروع شده و تا  $\pi$  امتداد دارد، تداخلی نداشته باشد. به طور معادل، یعنی در شکل بالا دو مثلث ناقص با یکدیگر تداخلی نداشته باشند. این حد پایین فرکانس نمونه‌برداری، فرکانس نایکوئیست نام دارد.
- فرض کنید که نمودار شکل ۱ نمونه‌برداری شده از یک سیگنال با فرکانس نمونه‌برداری  $\frac{1}{6}$  است. پهنای باند سیگنال اولیه چند بوده است؟ با استفاده از رابطه ۲ و تعبیری که از آن داده شد، اگر از سیگنال ذکر شده به جای فرکانس  $\frac{1}{6}$  با فرکانس  $\frac{1}{4}$  نمونه‌برمی‌داشتیم، چه اتفاقی می‌افتاد؟ شکل متناظر با این حالت را به کمک متلب رسم کرده و در گزارش کار ذکر کنید. به این پدیده aliasing گفته می‌شود.

## آشنایی با سیگنال‌های EEG

برای داده‌برداری از مغز ابزارهای متفاوتی وجود دارند که هر یک مزایا و معایبی دارد. یکی از روش‌های ذخیره سیگنال‌های مغزی، EEG<sup>۱</sup> است. این نوع سیگنال‌ها در واقع تغییرات سطح ولتاژهایی هستند که در اثر تغییرات و سیگنال‌های مغزی، در سطح الکترودهای دستگاه حس شده‌اند. این ولتاژها در سطح میکروولت هستند پس به شدت می‌توانند تحت تاثیر کوچک‌ترین نویزها قرار بگیرند. از مزایای EEG دقت زمانی بالا، یعنی فرکانس نمونه برداری بالا است اما از معایب آن، می‌توان به دقت مکانی کم اشاره کرد. از دیگر مزیت‌های EEG اندازه کوچک‌تر آن در مقایسه با مثلا دستگاه fMRI است که به خودی خود یک اتاق را اشغال می‌کند در حالی نسخه‌های قابل حمل EEG نیز موجودند.

کلاه‌های EEG ابزار ذخیره سیگنال‌های EEG هستند. این کلاه‌ها می‌توانند تعداد زیادی الکترود داشته باشند. برای مثال کلاه‌هایی با ۲۵۶ الکترود موجود هستند. این که هر الکترود در کدام نقطه از سر قرار بگیرد به صورت استاندارد موجود است. برای مثال، استاندارد کلاه ۶۴ الکترودی به شکل زیر است:



هر کدام از الکترودها نیز اسمی دارند که بر حسب موقعیت مکانی روی سر و این‌که چه ناحیه‌ای از مغز را پوشش می‌دهند تعیین شده‌است.

سیگنال‌های EEG فعالیت‌های متفاوت، مشخصه‌های متفاوتی دارند. برخی از این مشخصه‌ها در حوزه زمان و برخی دیگر در حوزه فرکانس بیشتر مشهود هستند. برای مثال یک مشخصه حوزه زمان، P300 است و از مشخصه‌های فرکانسی می‌توان به باندهای فرکانسی مختلف که هر یک نمایانگر دسته فعالیت‌های متفاوت هستند اشاره کرد.

- با جستجو در اینترنت در مورد مشخصه P300 و دیگر مشخصه‌هایی که به عنوان ERP<sup>۲</sup> شناخته می‌شوند اطلاعات کسب کنید و به صورت خلاصه در گزارش ذکر کنید.
- با جستجو در اینترنت در مورد باندهای فرکانسی مختلف اطلاعات لازم را بدست آورید و در گزارش ذکر کنید. هر باند فرکانسی نمایانگر چه فعالیت‌هایی است؟
- با توجه به این باندهای فرکانسی و قضیه نایکوئیست، چه فرکانس‌های نمونه‌برداری مناسب سیگنال‌های EEG است؟

<sup>۱</sup>Electroencephalography

<sup>۲</sup>Event Related Potential

اکنون که با مبانی سیگنال‌های EEG آشنا شده‌اید، به داده‌های موجود می‌پردازیم. در این فاز از پروژه تنها به داده‌های یک فرد دسترسی داریم. داده‌گیری توسط کلاه EEG با ۸ الکتروود است. اگرچه مقاله‌ی موجود ذکر کرده است که این ۸ الکتروود معادل کدام یک از الکتروودهای یک کلاه ۶۴ کاناله هستند اما در هیچ جای دیتابیس موجود ذکر نشده است که هر یک از بردارهایی که در متلب موجود هستند، کدام یک از این ۸ الکتروود هستند. این ۸ الکتروود عبارتند از:

Fz, Cz, Pz, P4, P3, Oz, Po7, Po8

فایل داده‌شده تحت عنوان Subject1.mat را در متلب باز کنید. این فایل حاوی ماتریسی با ۱۱ سطر است. سطر اول، زمان آزمایش است و هر یک از سطرها دوم تا نهم، داده‌های یکی از الکتروودها هستند. سطرها دهم و یازدهم مربوط به ساختار آزمایش هستند که در فاز بعدی با آن آشنا خواهیم شد.

● با استفاده از سطر اول، فرکانس نمونه برداری را محاسبه کنید.

داده‌های EEG به شدت می‌توانند تحت تاثیر نویز قرار بگیرند و به همین دلیل و برای حذف نویز یکی از مراحل ابتدایی پیش‌پردازش داده‌ها، فیلتر کردن آن‌ها است. ساخت فیلتر خود یک مبحثی مهم است که در فاز دوم پروژه به آن خواهیم پرداخت. در این فاز با استفاده از توابع متلب می‌خواهیم فیلتری ساخته و نتیجه‌ی اعمال آن را بر داده‌ها ببینیم.

● به کمک اطلاعاتی که از باندهای فرکانسی بدست آوردید، چه فرکانس قطعی مناسب است؟

● به کمک تابعی که در قسمت قبل نوشته‌اید، اندازه طیف فرکانسی کانال‌ها را بکشید. با توجه به این نمودارها چه فرکانسی به عنوان فرکانس قطع مناسب است؟

● معیاری دیگر برای انتخاب فرکانس قطع می‌تواند انرژی سیگنال باشد. با استفاده از توابعی که نوشته‌اید، فرکانسی را بیابید که بخش بزرگی از انرژی سیگنال در فرکانس‌های کوچک‌تر از آن تجمع شده باشد. (برای این کار البته بهتر است از فرکانس DC که ممکن است بخش بزرگی از انرژی سیگنال در آن ذخیره شده باشد، صرف‌نظر کنید.)

● به کمک موارد بالا، فرکانس قطع فیلتر پایین‌گذر را نهایی کنید

با توجه به این‌که فرکانس DC اطلاعاتی را منتقل نمی‌کند، آن را نیز از داده‌ها بایستی حذف کنیم. برای این کار، در مرحله اول میانگین سیگنال را از آن کم می‌کنیم و سپس یک فیلتر بالاگذر بر سیگنال اعمال می‌کنیم. با توجه به اینکه هم به یک فیلتر پایین‌گذر و هم یک فیلتر بالاگذر داریم، می‌توانیم با استفاده از یک فیلتر میان‌گذر، مراحل بالا را خلاصه کنیم. پس به طور خلاصه، پس از کم کردن میانگین داده‌ها، یک فیلتر میان‌گذر بر سیگنال اعمال می‌کنیم.

● چرا تنها کم کردن میانگین داده‌ها از سیگنال برای حذف فرکانس DC کافی نیست؟

در نهایت با توجه به نکات بالا و اطلاعاتی که بدست آورده‌اید فیلتری میان‌گذر را با استفاده از توابع متلب و یا هر تابع دیگری پیاده‌سازی کنید. می‌توانید از افزونه‌های ساخت فیلتر متلب نیز استفاده کنید.

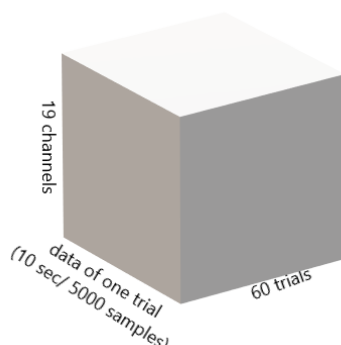
مرحله‌ی دیگری که معمولاً در پیش‌پردازش طی می‌شود، کاهش فرکانس نمونه برداری است. با توجه به اصل نایکوئیست، فرکانس نمونه‌برداری معادل با ۲ برابر پهنای باند سیگنال برای انتقال کامل اطلاعات سیگنال کافی است پس می‌توان فرکانس نمونه برداری را در صورتی که بیشتر از حد نایکوئیست باشد، تا آن حد کاهش دهیم.

● با توجه به نکته‌ای که گفته شد و با توجه به فرکانس قطعی که در مرحله پیش انتخاب کردید، فرکانس نمونه‌برداری را کاهش دهید.

**توجه** اثبات اینکه چرا ما قادر هستیم بعد از نمونه‌برداری از سیگنال، همچنان فرکانس نمونه‌برداری را کاهش دهیم، در اینجا بررسی نشده است. البته این اثبات چندان پیچیده نیست و افراد مشتاق می‌توانند با جست‌جو در اینترنت آن را فراگیرند. اغلب داده‌گیری‌هایی که انجام می‌شوند به صورت آزمایش به آزمایش و یا دقیق‌تر، trial-trial هستند. یعنی به فرد در هر

مرحله تحریکی داده می‌شود و یا از وی خواسته می‌شود که کاری را انجام دهد. پس می‌توانیم این آزمایش‌های پشت‌سرهم را نیز از یک‌دیگر جدا کنیم تا بررسی راحت‌تر شود. به این عمل epoching گفته می‌شود.

حال برای مراحل بعدی، باید داده‌ها را epoch کنیم. epoch یک ماتریس سه بعدی است که شامل تمام دیتای EEG به تفکیک چنل‌ها و ترايال‌ها است. هر سطر این ماتریس مربوط به یک چنل است و ترايال‌ها در بعد سوم قرار می‌گیرند. در واقع دیتای موجود در فایل ('Subject01.txt') را باید طوری تقسیم بندی کنید که به شکل زیر در بیاید.

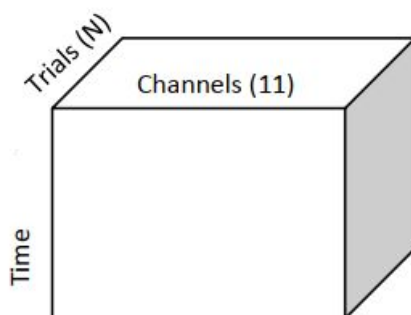


برای epoch کردن دیتا، به زمان شروع هر تسک نیاز داریم. برای اینکه شروع هر یک از تسک‌ها را در دیتا مشخص کنیم، از trigger یا mark استفاده می‌کنیم. به این صورت که یک ستون به دیتای EEG اضافه می‌کنیم که تنها در زمان شروع تسک، مقدار یک دارد (بقیه درایه‌های این ستون صفر هستند). از زمان تریگر، ۳۰ ثانیه طول می‌کشد تا ترايال‌ها آغاز شوند. **توجه کنید که اعداد در شکل بالا لزوماً صحیح نیستند و این شکل صرفاً برای نمایش تانسور epoch آورده شده است**

• تابعی بنویسید که با ورودی گرفتن سیگنال تمامی الکترودها و ۲ عدد اضافی، ماتریس epoch شده را در خروجی برگرداند. عدد اول، مشخص می‌کند که از چه زمانی قبل از اعمال تحریک، شروع به نگه‌داری داده برای هر پنجره بکنیم و عدد دوم نشانگر پایان هر پنجره است. مشخص است که به زمان شروع تحریک‌ها نیاز داریم. سطر دهم داده‌ها، این اطلاعات را دارد. هرجایی که داده‌ی غیر صفر آمده باشد، تحریک به فرد اعمال شده تا زمانی که تحریک بعدی برسد. پس تابع شما به صورت زیر خواهد بود:

epoching (InputSignal, #BackwardSamples, #ForwardSamples, StimuliOnset)

خروجی این تابع، ماتریسی ۳ بعدی به شکل زیر است:



• به کمک تابعی که نوشته‌اید، سیگال را به پنجره‌های حدوداً ۱۰۰۰ میلی‌ثانیه‌ای تقسیم کنید که از حدود ۲۰۰ میلی‌ثانیه قبل از شروع تحریک، شروع به نگه‌داری داده کرده و تا حدود ۸۰۰ میلی‌ثانیه پس از اعمال تحریک ادامه دارد.

**توجه** در مرحله‌ای که تا کنون طی کرده‌ایم، لازم است که مرحله فیلترکردن داده قبل از همه‌ی دیگر مراحل انجام شود. دلیل این امر را در سوال‌های زیر بررسی می‌کنیم:

• با توجه به اصل نایکوئیست و پدیده aliasing بگویید که چرا نمی‌توانیم فرکانس نمونه‌برداری را قبل از فیلترکردن کاهش دهیم.

● فرض کنید از سیگنالی به طول ۱۰۰۰ نمونه، از نمونه‌ی ۲۰۰ تا ۱۰۰۰ به شما داده شده است. فرض کنید فیلتری به طول ۱۰۰ در اختیار دارید، یعنی تنها درایه‌های ۰ تا ۹۹ مقادیر غیرصفر دارند. نکته‌ی گفته شده در بالا را بررسی کنید؛ در واقع به بررسی این پردازید که با این کار که ابتدا epoch کنیم و سپس فیلتر، چه فرقی خواهد داشت با این که ابتدا فیلتر کنیم و سپس پنجره را انتخاب کرده و جدا کنیم.

● با استفاده از نکته بالا بگویید که چرا لازم است در ابتدای داده‌گیری حتماً مدت زمانی پس از شروع داده‌گیری توسط کلاه، EEG هنوز آزمایش را شروع نکنیم.

نکات سوال‌های بالا را در یاد داشته باشید به خصوص در قسمت استخراج انرژی باندهای فرکانسی که خود طول فیلتر می‌تواند مشکل‌زا باشد. با توجه به سوال‌های بالا بدیهی است که بایستی طول فیلتر از طول سیگنال و همچنین از طول epoch ها نیز کمتر باشد. شرط اول که معمولاً برقرار است چرا که طول خود سیگنال در حدود چند میلیون نمونه است اما شرط دوم همواره باید بررسی شود.

## خوشه‌بندی بر مبنای همبستگی

در این بخش قصد داریم که الکترودهای مختلف را بر مبنای شباهت سیگنال‌های زمانی هر کدام، به چند دسته تقسیم کنیم. یک معیار برای شباهت دو سیگنال زمانی، همبستگی متقابل این دو سیگنال در  $\tau = 0$  است. فرض کنید دو سیگنال زمانی  $X(t)$  و  $Y(t)$  در اختیار داریم، همان‌طور که می‌دانیم مقدار تابع خودهمبستگی این دو سیگنال در  $\tau = 0$  برابر

$$R_{XY}(0) = \int_{-\infty}^{\infty} X(t)Y(t)dt$$

است.

● نشان دهید که

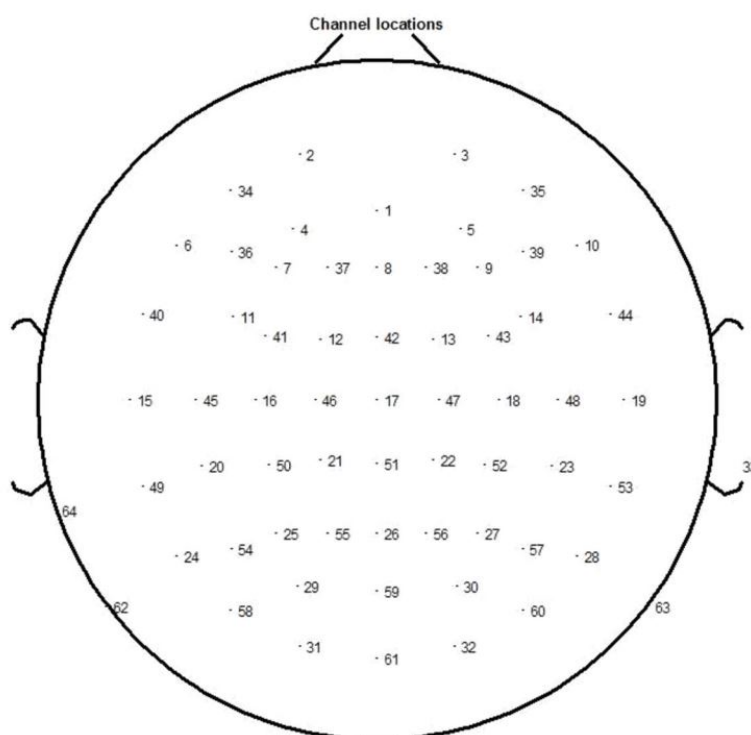
$$r_{XY} = \frac{\int_{-\infty}^{\infty} X(t)Y(t)dt}{\sqrt{\int_{-\infty}^{\infty} X^2(t)dt \int_{-\infty}^{\infty} Y^2(t)dt}}$$

عددی بین 1 و -1 است.

● نشان دهید  $|r_{XY}| = 1$  تنها در حالتی رخ می‌دهد که دو سیگنال تنها در یک ضریب تفاوت داشته باشند یعنی  $X(t) = \alpha Y(t)$ .

● استدلال کنید که چرا این معیار، معیار مناسبی برای سنجش شباهت دو سیگنال است.

همان‌گونه که گفته شد، در دیتاست موجود، مکان ۸ الکتروده به صورت دقیق مشخص نیستند و تنها می‌دانیم که این ۸ الکتروده در کل کدام مجموعه الکترودها هستند. می‌خواهیم با استفاده از همبستگی و خوشه‌بندی به کمک آن، ببینیم که آیا می‌توانیم ارتباطی بین این الکترودها پیدا کنیم یا نه. برای این کار ابتدا بر روی دیتاست دیگری که مکان دقیق الکترودهای آن را می‌دانیم تابع خوشه‌بندی‌کننده را می‌نویسیم و پس از صحت از کارکرد آن، نهایتاً تابع را بر روی داده‌های آزمایش فعلی اعمال خواهیم کرد تا سعی کنیم ارتباطی میان الکترودها پیدا کنیم. دیتاستی که برای این قسمت اضافه در اختیار دارید، برای یک آزمایش BCI دیگر با هدف تشخیص نوع حرکت است. این دیتاست با فرکانس نمونه برداری ۶۰۰ اخذ شده‌است و خود داده‌ها به صورت epoch شده هستند. مکان ۶۴ الکتروده این آزمایش در شکل زیر آمده‌اند.





مراحل زیر را برای دیتاست موجود انجام دهید. توجه کنید که برای انجام صحیح این قسمت بایستی قسمت‌های قبل را به درستی فهمیده باشید و توجه شما به تمامی نکاتی که در قسمت‌های پیشین گفته شده ضروری است. مرحله‌ای که بایستی طی کنید را به دقت در گزارش ذکر کرده و علت هر یک را با توجه به قسمت‌های پیشین بنویسید. عدم رعایت هر یک از مراحل باعث کسر نمره و مهم‌تر از آن باعث نویزی شدن نتایج و کم شدن دقت و صحت داده‌ها می‌شود.

- با مشاهده طیف فرکانسی، در صورت لزوم فیلتری را بر داده‌ها اعمال کنید تا نویز در نتایج حاصل اثر هرچه کم‌تری داشته باشد.

- فرکانس نمونه‌برداری را تا حد معقولی کاهش دهید.

با توجه به اینکه شصت و چهار سیگنال زمانی از کانال‌های مختلف در اختیار دارید، ماتریسی  $64 \times 64$  بسازید که درایه‌ی  $ij$  آن برابر  $r_{X_i X_j}$  است. بدیهی است که این ماتریس، ماتریسی متقارن است. به این ماتریس، ماتریس همبستگی می‌گویند.

- به کمک ماتریسی که در بالا پیدا کردید، برای خوشه‌بندی (یعنی دسته‌بندی کانال‌ها) بر مبنای معیار شباهت، می‌توان از ایده‌ای مشابه ایده‌ی زیر استفاده کرد.

۱. هر کانال را در یک خوشه قرار دهید.

۲. دو خوشه با نزدیک‌ترین فاصله را با هم ادغام کنید.

۳. قدم قبل را آن قدر تکرار کنید تا تمام کانال‌ها در یک خوشه قرار بگیرند.

برای فاصله‌ی دو کانال، که زیاد بودن آن نشان دهنده‌ی عدم شباهت دو کانال است، چه رابطه‌ای را پیشنهاد می‌کنید؟ دو پیشنهاد برای فاصله‌ی دو خوشه (هر کدام متشکل از چند کانال) نیز ارائه دهید.

- الگوریتم فوق را پیاده‌سازی کنید. در پیاده‌سازی خود، از فواصلی که در بخش قبل پیشنهاد کردید استفاده کنید. تابع خود را به صورت زیر تعریف کنید. اگر به ورودی‌های دیگری نیز نیاز دارید آن‌ها را بیافزایید.

`CorrelationCluster (InputCorrMat, DistanceMeasure)`

- در هر مرحله‌ای که الگوریتم فوق را متوقف کنیم، یک خوشه‌بندی به دست می‌آید. به نظر شما چگونه می‌توان خوشه‌بندی مناسب را پیدا کرد؟ معیاری برای مقایسه‌ی خوشه‌بندی‌های مختلف معرفی شده توسط الگوریتم بالا که هر کدام تعداد خوشه‌های متفاوتی دارند ارائه کرده و به کمک این معیار، یک خوشه‌بندی را به عنوان خوشه‌بندی منتخب معرفی کنید.

- با شهودی که خود دارید و یا با جست‌وجو در اینترنت به دنبال این بگردید که چه ارتباطی بین الکترودها می‌تواند برقرار باشد؟ آیا مثلاً الکترودهای سمت چپ از سمت راست مستقل هستند؟ چه انتظاری از الکترودهای نزدیک به یکدیگر دارید؟

- با توجه به نتایج حاصل از خوشه‌بندی و مکان الکترودها آیا رابطه‌ای بین الکترودهای بین موجود در هر خوشه مشهود است؟

- پس از اطمینان از کارکرد الگوریتم و انتخاب معیار مناسب فاصله، تمام مراحل بالا را برای دیتاست اصلی ۸ الکترودی که می‌خواهیم بر روی آن کار کنیم پیاده‌سازی کنید. آیا ارتباطی بین الکترودها داریم؟

- برای تشخیص بهتر و نمایش بهتر خوشه‌بندی الکترودها از تابع `plottopography` که در تمرین به شما داده شده استفاده کنید و الکترودهایی که با یکدیگر در یک دسته قرار می‌گیرند را توسط این تابع یک عدد به آنها نسبت دهید و این این طریق ارتباط میان الکترودها را ترسیم کنید. همچنین اگر معیار طبقه‌بندی شما به چندین حالت می‌رسد تمامی حالات را با استفاده از این تابع ترسیم کنید.

## طراحی فیلتر

فیلترهای انتخاب فرکانس<sup>۳</sup> ابزارهایی برای حذف محتوای فرکانسی ناخواسته مانند نویز، و نگهداری باندهای فرکانسی مطلوب می‌باشند. برای طراحی و استفاده از فیلترها لازم است نکاتی را مورد توجه قرار داد که در این بخش، به بررسی نمونه‌هایی از مهم‌ترین این نکات می‌پردازیم.

اندازه و فاز، دو مشخصه‌ی مهم یک فیلتر هستند. در حالت کلی، تبدیل فوریه را می‌توانیم به صورت زیر (بر مبنای اندازه و فاز) نمایش دهیم:

$$H(j\omega) = |H(j\omega)| e^{j\angle H(j\omega)} \quad (۴)$$

اگر چه تعریف فوق از نظر ریاضی معتبر است، اما ویژگی ناخوش‌آیندی دارد که به شرح زیر است: فرض کنید  $H(j\omega)$  به صورت حاصل ضرب دو عبارت به شکل  $H(j\omega) = F(j\omega)G(j\omega)$  قابل بیان باشد که در آن،  $F(j\omega)$  همواره حقیقی باشد. در این صورت بدیهی است که هرگاه  $F(j\omega)$  تغییر علامت بدهد، فاز  $H(j\omega)$  به اندازه  $\pi$  رادیان تغییر می‌کند؛ لذا همواره در محل‌های تغییر علامت  $F(j\omega)$ ، فاز تبدیل فوریه ناپیوستگی‌هایی با اندازه  $\pi$  خواهد داشت. با این حال، این ناپیوستگی‌ها صرفاً ناشی از تعریف ما از فاز هستند و رفتار طبیعی فاز تبدیل را – که احتمالاً یک رفتار هموار و پیوسته بوده است – به درستی توصیف نمی‌کنند. برای رفع این مشکل، کافی است از تعریف زیر استفاده کنیم:

$$H(j\omega) = A(j\omega)e^{j\phi(j\omega)} \quad (۵)$$

که در آن  $A(j\omega)$  حقیقی است و می‌تواند مثبت یا منفی باشد. دقت کنید که تنها تفاوت معادله‌ی ۵ با معادله‌ی ۴ آن است که در تعریف جدید،  $|H(j\omega)|$  را که یک عدد حقیقی مثبت است، با  $A(j\omega)$  جایگزین کرده‌ایم که یک عدد حقیقی مثبت یا منفی است. این کار، مشکل مذکور را حل کرده و اثر تغییر علامت‌های ممکن را از فاز تبدیل خارج کرده و به  $A(j\omega)$  منتقل می‌کند. عبارت  $\phi(j\omega)$  از معادله‌ی ۵ را فاز تعمیم‌یافته‌ی خطی یا generalized linear phase می‌نامیم. فاز فیلتر از ویژگی‌های مهم آن است که در نگاه اول شاید به آن توجه نشود. یکی از مهم‌ترین ویژگی‌هایی که در فیلترها مطلوب است، خطی بودن فاز تعمیم‌یافته‌ی فیلتر است. پرسش زیر، اهمیت این موضوع را بررسی می‌کند.

● فرض کنید فیلتری دارید که اندازه‌ی آن ثابت است. فاز این فیلتر را نیز به صورت زیر در نظر بگیرید:

$$\phi(\omega) = -\frac{\pi}{3}\text{sgn}(\omega)$$

ورودی فیلتر، سیگنال زیر است:

$$x(t) = \cos(\omega_0 t) + \cos(2\omega_0 t)$$

خروجی سیستم را پیدا کنید. سیگنال ورودی از دو سیگنال تک‌فرکانس تشکیل شده بود. هر یک از این دو سیگنال، پس از عبور از این فیلتر چه مقدار در زمان تأخیر پیدا کرده‌اند؟ این بار فرض کنید فاز فیلتر خطی باشد، یعنی

$$\phi(\omega) = -\frac{\pi}{3}\omega$$

پاسخ دو پرسش فوق را برای این حالت نیز بیان کنید. به نظر شما خطی بودن فاز فیلتر چه اثر مثبتی دارد؟ خطی نبودن فاز فیلتر چگونه می‌تواند باعث ایجاد اعوجاج در سیگنال خروجی شود؟

با توجه به پرسش فوق، باید به این نکته پی برده باشید چرا در طراحی فیلتر، همواره به دنبال آن هستیم که در صورت امکان، فاز فیلتر در باندهای فرکانسی مورد توجه ما، خطی باشد. برای این منظور متغیری تحت عنوان group delay را به صورت زیر تعریف می‌کنیم:

$$\text{gd}(\omega) = -\frac{d\phi(\omega)}{d\omega} \quad (۶)$$

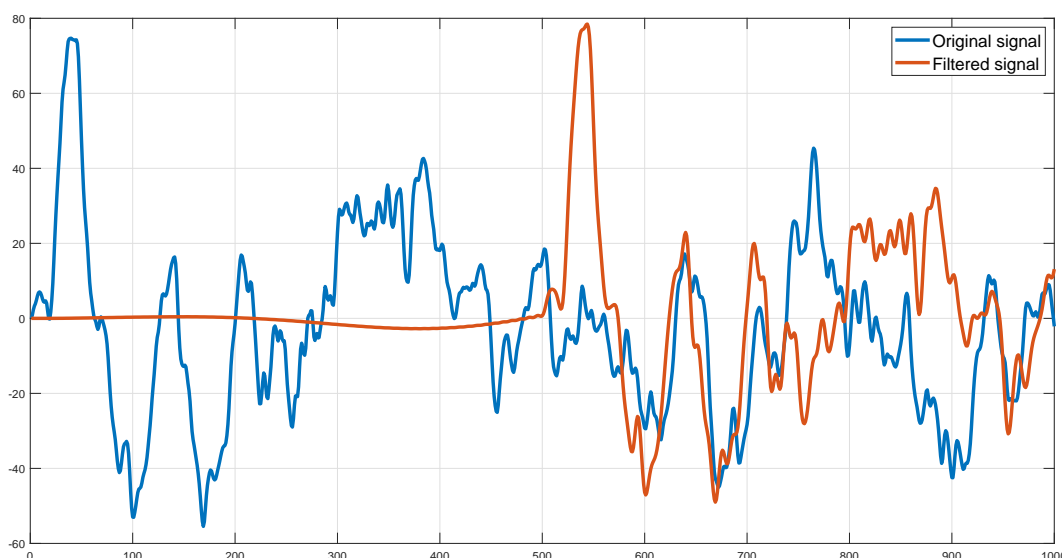
<sup>۳</sup>frequency selective filters

که در آن،  $\phi(\omega)$  فاز تعمیم‌یافته‌ی خطی است. بُعد group delay برای سیستم‌های گسسته، تعداد نمونه یا سمپل بوده و برای سیستم‌های پیوسته، زمان (بر حسب ثانیه) است.

گزاره‌ی مهم: تأخیر گروه برای محتوای هر فرکانس، نشان‌گر این است که پس از عبور از فیلتر، چه مقدار تأخیر (بر حسب سمپل یا ثانیه) خواهد یافت. (به عبارت دقیق‌تر، این کمیت در حالت گسسته بدون بُعد است.)

• درستی گزاره‌ی فوق را برای سیگنال ورودی و دو سیستم معرفی‌شده در پرسش قبل تحقیق کنید.

اگر در فاز قبل پروژه به خروجی‌های فیلتر توجه می‌کردید، می‌توانستید این تأخیر را مشاهده کنید. در شکل زیر نیز به وضوح می‌توان این تأخیر را برای سیگنالی که با فیلتری با  $\text{group delay} = 500$  samples فیلتر شده است، مشاهده نمود:



شکل ۲

با توجه به نکات فوق، واضح است که حتماً باید این تأخیر را در محاسبات خود وارد کرده و اثرات آن را در نظر بگیریم، در غیر این صورت به احتمال خوبی در محاسبات خود دچار اشتباه خواهیم شد. برای این کار، ابتدا لازم است مقدار  $\text{group delay}$  را برای یک فیلتر داده‌شده به دست آوریم.

• با استفاده از روابط ۵ و ۶، ثابت کنید:

$$\text{gd}(\omega) = \text{Re} \left\{ \frac{j \frac{d}{d\omega} H(\omega)}{H(\omega)} \right\} \quad (7)$$

• تابعی بنویسید تا به کمک معادله‌ی ۷، تأخیر گروه را برای هر فیلتر دلخواه محاسبه کند. بدیهی است مجاز به استفاده از توابع آماده‌ی متلب که مستقیماً تأخیر گروه را محاسبه می‌کنند، نیستید. نام‌گذاری تابع شما باید به صورت زیر باشد:

`groupdelay(h,N)`

که در آن، متغیر  $N$  مشخص می‌کند که DFT مورد استفاده در محاسبه‌ی تأخیر گروه (به کمک تابع `fft`) چند نقطه‌ای باشد. هر چه این عدد بزرگتر باشد دقت محاسبات بالاتر است. (چرا؟)

• نمودار خروجی تابع بالا را، بر حسب فرکانس، برای هر فیلتری که در این تمرین استفاده خواهید کرد، در گزارش بیاورید. (برای بررسی صحت عملکرد تابع خود، می‌توانید نتایج آن را با تابع `grpdelay` متلب مقایسه کنید.)

• با استفاده از تابع بالا، تابعی دیگر بنویسید که برای یک فیلتر و سیگنال داده‌شده، سیگنال را فیلتر کرده و تأخیر ناشی از فیلترینگ را نیز لحاظ کرده و خنثی کند. تابع را به صورت زیر نام‌گذاری کنید:

`zphasefilter(h,x)`

که در آن  $x$  سیگنال ورودی، و  $h$  پاسخ ضربه‌ی فیلتر است. آیا فیلتری علی و حقیقی وجود دارد که فاز آن در تمام فرکانس‌ها صفر باشد؟

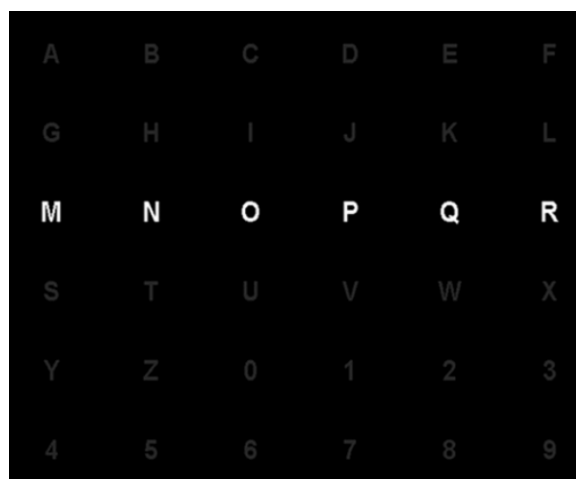
توجه: تولباکس متلب تحت عنوان `filter designer` مجموعه‌ای غنی از ابزارهای طراحی و بررسی فیلترها است. در قسمت‌های بعدی می‌توانید برای طراحی فیلتر از این تولباکس استفاده نموده و با استفاده از ابزارهای موجود در آن، اطلاعات زیادی در مورد فیلترها کسب کنید. همچنین می‌توانید از خود تابع `bandpass` متلب نیز استفاده کنید که به نوعی همان کار را با الگوریتم بهینه انجام می‌دهد.

## شناسایی کلمات

تمامی مراحل که تا کنون طی شده‌اند برای آماده‌شدن داده‌ها جهت استفاده در این بخش از پروژه بوده است. در این قسمت سعی داریم که با استفاده از الگوریتم‌های مبتنی بر یادگیری ماشین و از طریق سیگنال‌های مغزی EEG، کلمه‌ای را که فرد به آن فکر می‌کند، تشخیص دهیم و یک سیستم تایپ با مغز بسازیم!

### شرح آزمایش

فرد مورد آزمایش در مقابل نمایش‌گری به شکل ۳ قرار می‌گیرد. این نمایش‌گر می‌تواند همه‌ی حروف انگلیسی و اعداد از صفر تا ۹ را با آرایش  $6 \times 6$  (مجموعاً ۳۶ کاراکتر) نشان دهد. هدف نهایی آن است که فرد تنها با نگاه کردن به این صفحه‌ی نمایش بتواند با یک سیستم مبتنی بر هوش مصنوعی و یادگیری ماشین ارتباط برقرار کند، به گونه‌ای که کلمه‌ای که در ذهن دارد از طریق سیستم شناسایی شده و نمایش داده شود. برای این منظور، لازم است کلمه به صورت حرف به حرف مشخص شود. برای تعیین یک حرف مشخص، فرد برای مدتی به محل آن کاراکتر روی صفحه‌ی نمایش خیره می‌شود. در طول این زمان، همه‌ی کاراکترهای نمایش‌گر با الگوی زمانی مشخصی روشن و خاموش می‌شوند، به گونه‌ای که در هر لحظه از زمان، ممکن است همه‌ی کاراکترها خاموش باشند، و یا بسته به نوع پیاده‌سازی آزمایش، یک یا چند کاراکتر روشن باشند. (در مورد نوع آزمایش و الگوی زمانی روشن و خاموش شدن کاراکترها در ادامه توضیح داده خواهد شد.) در تمام طول این آزمایش، سیگنال‌های مغزی فرد در حال ثبت شدن است.



شکل ۳

هدف نهایی آزمایش آن خواهد بود که با استفاده از سیگنال‌های مغزی، بتوانیم زمان‌هایی را تشخیص دهیم که حرف مدّ نظر فرد، روشن شده است. به عبارت دیگر، انتظار داریم وقتی فرد برای مدتی به حرفی خیره شده است، در لحظاتی که آن حرف روشن می‌شود، در سیگنال مغزی دریافتی اثراتی مشاهده شود، به گونه‌ای که بتوانیم با پردازش آن‌ها لحظاتی را که کاراکتر مطلوب روی صفحه‌ی نمایش روشن شده‌اند، شناسایی کنیم. اگر در تشخیص این زمان‌ها موفق باشیم، کار تقریباً تمام شده است، چرا که می‌توانیم با مراجعه به الگوی زمانی روشن و خاموش شدن کاراکترها (که از ابتدا توسط ما مشخص شده بود و اطلاعات آن را در دست داریم)، بفهمیم کدام کاراکتر در آن لحظات روشن شده و بنابراین، حرف مدّ نظر شخص مورد آزمایش را تشخیص دهیم. با تشخیص هر حرف، فرد محلّ نگاه خود را عوض می‌کند و به سراغ حرف بعدی می‌رویم. همچنین برای آن که آزمایش در مقابل نویز و خطا مقاوم‌تر باشد، این فرایند برای هر حرف چندین بار تکرار می‌شود. یعنی به عنوان مثال در مدت زمانی که فرد به یک کاراکتر خیره شده، فرایند روشن و خاموش شدن حروف ۱۵ بار تکرار شود. بنابراین می‌توانیم ۱۵ کاراکتر را با الگوریتم فوق تشخیص دهیم که اگر هیچ خطایی وجود نداشته باشد، باید همگی یکسان و برابر با کاراکتر هدف باشند. بدیهی است به دلیل وجود نویز و خطاهای عملیاتی، تمامی این ۱۵ حاصل یکسان نخواهند بود، اما انتظار داریم درصد بالایی از آن‌ها یکسان بوده و نتیجه‌ی مطلوب را در اختیار ما قرار بدهند. با اتمام این فرایند، با تغییر محلّ نگاه فرد مورد آزمایش، به سراغ حرف بعدی می‌رویم.

دو پارادایم مختلف برای آزمایش (چگونگی روشن و خاموش شدن کاراکترها روی صفحه) وجود دارد که آزمایش هر یک از افراد، توسط یکی از این دو روش انجام می‌شود.

#### ۱. Single Character Paradigm

کاراکترها یک به یک روشن و خاموش می‌شوند؛ بنابراین در هر لحظه از زمان، حداکثر یک کاراکتر بر روی صفحه نمایش روشن است.

#### ۲. Row-Column Paradigm

هر بار یک سطر یا یک ستون روشن و خاموش می‌شود و در نتیجه هر حرف از تقاطع یک سطر و یک ستون به دست می‌آید. (به عنوان مثال، شکل ۳ نمونه‌ای از این روش است.)

در توضیح دقیق‌تر آزمایش می‌توان گفت که در روش SC paradigm برای تشخیص یک حرف، تمامی حروف ۱۵ بار روشن می‌شوند و در روش RC paradigm هر سطر و هر ستون ۱۵ بار روشن می‌شوند. شاید در نگاه اول به نظر برسد که روش اول بهتر است و نیازی به روش دوم نیست، اما مشاهده نشان می‌دهد که ممکن است روش دوم به دقت بالاتری دست یابد.

### بررسی دیتاست و پردازش‌های اولیه

در دیتاست موجود می‌توان با استفاده از سطر ۱۰ام برای هر فرد، تشخیص داد که چه سطر، ستون و یا حرفی روشن شده است.

- با استفاده از داده‌های سطر ۱۰ام هر فرد، تشخیص دهید که هر فرد با کدام یکی از paradigmها (RC یا SC) آزمایش را انجام داده است و در گزارش ذکر کنید.

فرد در این آزمایش می‌خواهد یک کلمه را با استفاده از روش‌های بالا بگوید. تمامی داده‌های یادگیری (train) با استفاده از کلمه‌ی LOKUS اخذ شده‌اند. هر وقت که خود کاراکتر (SC) و یا سطر یا ستون حاوی کاراکتر (RC) روشن شود، در سطر ۱۱ام دیتاست عدد ۱ ذخیره شده است. با توجه به این نکات به سوال زیر پاسخ دهید:

- با توجه به این که کلمه‌ی یادگیری LOKUS است، نحوه‌ی شماره‌گذاری سطرها و ستون‌ها را برای RC paradigm و شماره‌گذاری کاراکترها را برای SC paradigm تشخیص دهید و در گزارش کار ذکر کنید. (یعنی باید بگویید هر یک از حروف صفحه نمایش، چگونه در سطر دهم دیتاست ذکر شده است.)

توجه: برای جلوگیری از زیاد شدن متغیرها و شلوغی، تمامی داده‌های مربوط به هر یک از افراد را در یک استراکت تحت عنوان subject ذخیره نمایید و فیلدها را با نام‌گذاری مناسب اضافه کنید.

- با استفاده از داده‌های آزمایش تابعی تحت عنوان IndexExtraction بنویسید که با ورودی گرفتن استراکت کلی، اندیس زمان نمایش هر حرف (برای پارادایم SC) یا سطر و یا ستون (برای پارادایم RC) را برگرداند. در خروجی باید یک فیلد جدید تحت عنوان time به استراکت ورودی اضافه شود که در آن به تفکیک target یا not-target بودن، اندیس لحظه‌ی روشن شدن سطر یا ستون و یا کاراکتر متناظر برای هر دوسری داده‌ی train و test آمده باشد. منظور از target این است که در اثر این روشن شدن، همان کاراکتر موردنظر فرد روشن شده است. البته تفکیک بین target و غیر آن برای داده‌های test میسر نیست. در واقع قرار است تنها داده‌های غیر صفر سطر دهم را انتخاب کنید و اندیس آن‌ها را برگردانید و سپس با استفاده از سطر یازدهم هم بین target و not-target تمایز قائل شوید.

با آشنایی که اکنون از آزمایش دارید می‌توان به سراغ پیاده‌سازی الگوریتم رفت. با توجه به نکات فاز قبل و قسمت قبل، داده‌ها را فیلتر کنید و سپس با استفاده از تابع epoching که در فاز قبل آن را نوشتید و با استفاده از فیلد time، پنجره‌بندی مناسب را انجام دهید. در پنجره‌بندی می‌توانید جهت سهولت کار در ادامه، برای سری داده‌های train، داده‌های target و non-target را در دو ماتریس جدا از هم بگذارید.

## پیاده‌سازی الگوریتم یادگیری ماشین

با توجه به قسمت‌های قبل اکنون می‌توان به فرآیند یادگیری پرداخت. بدین منظور لازم است که ابتدا یک مدل را بر اساس یک الگوریتم آموزش دهیم و سپس با استفاده از آن، داده‌های جدید را لیبِل بگذاریم. داده‌های آموزش هر فرد را در ماتریس `Train_Features` قرار دهید که در این ماتریس، هر ستون یک ویژگی و هر درایه، یک تحقق از آن ویژگی است. بدیهی است که ساختار این ماتریس برای داده‌های تست نیز باید دقیقاً مشابه باشد یعنی به طور مثال اگر ستون اول میانگین داده‌های هر پنجره است، برای داده‌های تست نیز این ستون بایستی همین ویژگی باشد. در واقع می‌توان بدین شکل فرض کرد که ستون‌ها بردارهایی هستند که فضای شما را می‌سازند پس باید برای داده‌های تست نیز در هر بعد داده‌ی متناظر با آن بعد را قرار داد. هر سطر از ماتریس بالا، یا متناظر با فعالیت `target` است و یا `non-target`. در یک بردار، برای هر سطر که متناظر با یک سطر `target` است، ۱ و در غیر این صورت، صفر بگذارید. با رعایت فرمت بالا اکنون می‌توانید مدل را آموزش دهید. برای این کار می‌توانید از الگوریتم‌های مختلف موجود استفاده کنید. دو الگوریتم پرکاربرد `LDA` و `SVM` هستند. این دو با استفاده از توابع `fitcdiscr` و `fitcsvm` پیاده‌سازی می‌شوند. خروجی این توابع یک مدل آموزش دیده از همان الگوریتم است. مدل هر فرد را در استراکت کلی تحت همان فرد ذخیره کنید.

در ادامه می‌توان با استفاده از تابع `predict` داده‌های جدید را طبقه‌بندی کرد. نحوه‌ی کار با تابع `predict` راحت است و به عنوان ورود مدل آموزش یافته و ماتریس ویژگی داده‌هایی که خواستار طبقه‌بندی آن‌ها هستیم را می‌خواهد و در خروجی برچسب هر سطر را برمی‌گرداند.

- برای هر فرد مدلی بر اساس هر دو روش یادگیری ماشین را آموزش دهید.
- اگر به عنوان ورودی به تابع `predict` همان ماتریس `Train_Features` را بدهیم، انتظار دارید که چه درصد صحتی بگیرید؟ این مورد را بررسی کنید و در گزارش، خروجی که در واقعیت دریافت می‌شود را ذکر کنید. علت این امر چیست؟
- اگر در این مرحله با مقایسه صفر و یکی تشخیص `target` و `non-target` بودن درصد صحت بالایی می‌گیرید بایستی بدین توجه کنید که با توجه به اینکه تعداد `trial`های `non-target` بسیار بیشتر از حالت `target` است، شما اگر خود بدین صورت ببایید و خروجی‌ها را به صورت دستی همگی را `non-target` تشخیص دهید، درصد صحت بالایی خواهید گرفت. پس بدین منظور لازم است که تابع هزینه شما وزن بیشتری برای تشخیص اشتباه داده‌های `target` به عنوان `non-target` داشته باشد. بدین منظور تابع هزینه خود را مناسب تعریف کنید.
- اگرچه تا کنون یک مدل را آموزش داده‌اید و بر حسب آن `target` و غیر آن را از یکدیگر تشخیص داده‌اید اما هنوز کلمه‌ای را تشخیص نداده‌اید. برای این منظور تابعی بنویسید که با ورودی گرفتن خروجی‌های فرایند `predict`، کلمه‌ای را که فرد در صدد گفتن آن بوده‌است را تشخیص بدهد.
- خروجی را برای داده‌های تست هر فرد ببایید و تشخیص بدهید که هر فرد در صدد گفتن چه کلمه‌ای بوده‌است. کلمه‌ی موردنظر در داده‌گیری `test` هم ۵ حرفی بوده‌است.
- دقت کنید که در تمامی مراحل بالا دیتای مورد یادگیری قرار گرفته شما همان `epoch` هایی است که از قسمت های قبلی برای هر فرد درآورده اید و نیازی به هیچ متغیر و یا ویژگی و `feature` دیگری جز همان `epoch` ها ندارید.