بسمه تعالی

دانشگاه صنعتی شریف
دانشکده مهندسی برق

دکتر کربلایی آقاجان

گزارش کار تمرین سری سوم متلب
سیگنال و سیستم

طاها انتصاری 95101117

پیش پردازش

با توجه به توضیحات صورت دستور کار،از آنجایی که ما با فیلتری با فرکانس قطع 60 داده ها را فیلتر کردیم،پس فرکانس سمپلینگ را 120 هرتز انتخاب میکنیم.از آنجایی که 2400/120=20 پس داده ی جدید ما به صورت زیر خواهد بود

X[n]=x[20n].البته اگر بخواهیم کد را در متلب وارد کنیم،باید چیزی شبیه کد زیر را وارد کنیم X[n]=x[20(n-1)+1]

توجیه دیگری برای علت کاهش فرکانس سمپلینگ را میتوان به صورت زیر فرض کرد:

از آنجایی که میدانیم که سیگنال های مغز در فرکانس های بیشتر از 30 هرتز اطلاعاتی ندارند،پس استفاده از فرکانس سمپلینگ 2400 لازم نیست چرا که با همان فرکانس 120(البته بنابه قضیه فرکانس نیکوئیست فرکانس 60 کافی مباشد) هم میتوان اطلاعات لازم را بدست آورد و داده های اولیه را بازسازی کرد و فرکانس سمپلینگ بالا تنها مقدار محاسبات را زیاد میکند

که از آنجایی که حجم داده های اولیه بسیار زیاد میباشد(حدود 3 گیگابایت) پس بهتر است فرکانس را کاهش دهیم.بعد از کاهش فرکانس سمپلینگ حجم داده ها به کمتر از 200 مگابایت کاهش یافت.

اثر خاصی از نویز برق شهر در فرکانس 50 یا 60 هرتز وجود ندارد و دامنه تبدیل فوریه در این فرکانس ها کم میباشد.برای اطمینان یک فیلتر پایین گذر با فرکانس قطع 45 بر داده ها اعمال میکنیم.

سیگنال انرژی بسیار زیادی در فرکانس صفر دارد که حاوی اطلاعاتی نیستند.

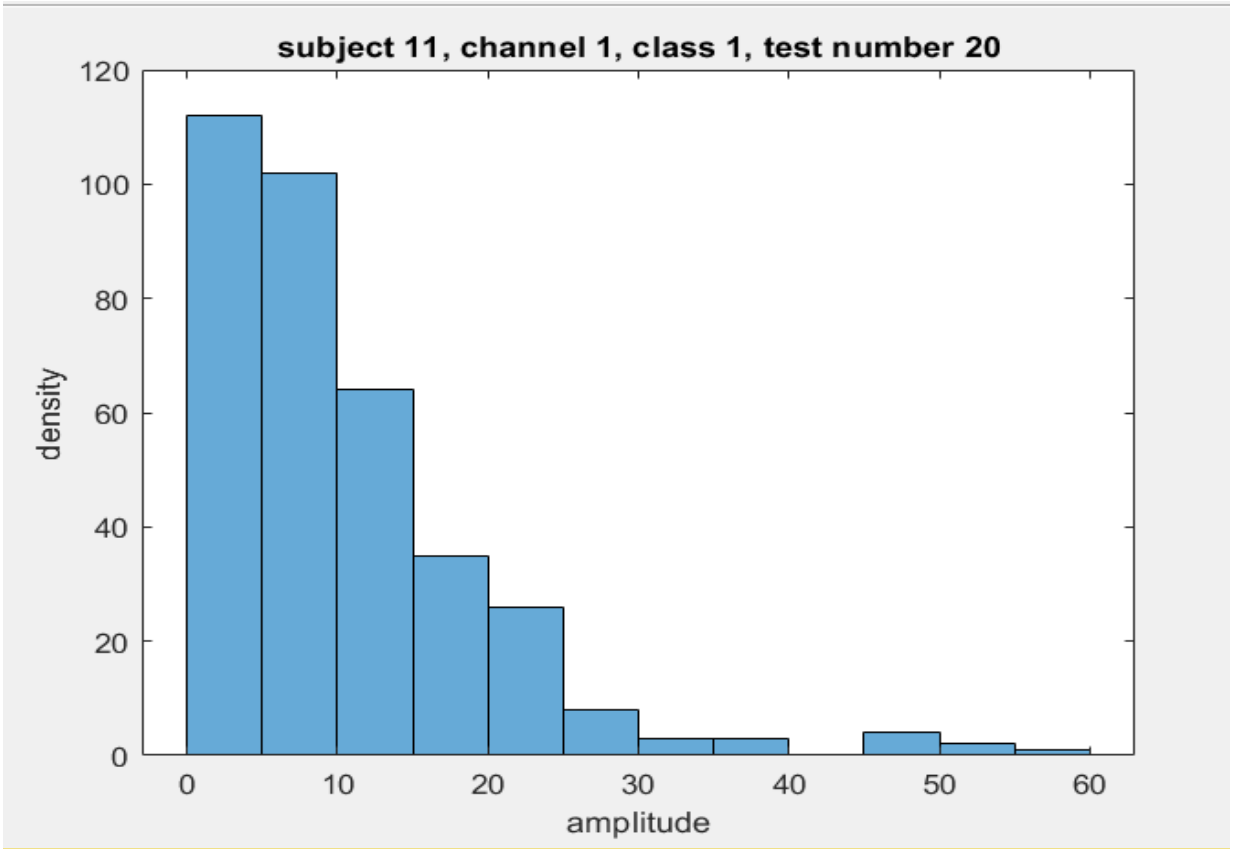
بنا به سایت ویکی پدیا، باند فرکانسی دلتا(که پایین ترین فرکانس را داراست) حاوی اطلاعات در بازه 4-0.5 هرتز میباشد.پس یه فیلتر بالاگذر با فرکانس قطع 0.5 هرتز نیز بر فیلتر اعمال میکنیم تا اثر dc ناخواسته را حذف کنیم.

https://en.wikipedia.org/wiki/Delta_wave

استخراج ویژگی

ویژگی های مذکور را برای داده ها حساب میکنیم.

- در زیر یک نمونه نمودار هیستوگرام دامنه آمده است:



همبستگی را برای یک تست مذکور و تمامی کانال ها و هم برای یک کانال و تمامی تست های آن کانال انجام داده ایم.البته به نظر میرسد که تنها به همبستگی بین کانال های یک فعالیت یا یک تست نیاز داریم.چرا که شباهت بین 2 فعالیت یک کانال،در صورت وجود، به نظر میرسد که نمایانگر ویژگی خاصی نباشد.

- فرم فاکتور بنا به تعریف برابر است با نسبت RMS به مقدار میانگین قدرمطلق سیگنال،یعنی:

$$k_{\mathrm{f}} = \frac{\mathrm{RMS}}{\mathrm{ARV}} = \frac{\sqrt{\frac{1}{T}\int_{t_0}^{t_0+T} [x(t)]^2\, dt}}{\frac{1}{T}\int_{t_0}^{t_0+T} |x(t)|\, dt} = \frac{\sqrt{T\int_{t_0}^{t_0+T} [x(t)]^2\, dt}}{\int_{t_0}^{t_0+T} |x(t)|\, dt}$$

این ویژگی برابر با نسبت "جریان " معادل دی سی به "جریان" ای سی سیگنال میباشد.

- گشتاور مرکزی مرتبه 3 یا همان skewness را نیز حساب کرده ایم.این ویژگی معیاری از عدم تقارن توزیع احتمالی داده ها حول میانگین میباشد.

- فرکانس میانگین،میانگین وزن دار انرژی سیگنال میباشد

- فرکانس مد بنا به تعریف، فرکانسی میباشد که بیشترین دامنه سیگنال در حوزه فرکانسی را داراست،نحوه بدست آوردن این ویژگی در کد موجود میباشد.

- ویژگی DST یا تبدیل سینوسی گسسته،تبدیلی شبیه به تبدیل فوریه گسسته است با این فرق که از ماتریس حقیقی کار میکند.در متلب این تابع به صورت زیر تعریف شده است:

$$y(k)=N\Xi n=1x(n)\sin(\pi knN+1),\quad k=1,...,N$$

ویژگی DCT نیز مشابه بالا میباشد.

داده ها را نیز با 4 فیلتر مناسب به باند های فرکانسی متناظر جدا کرده و انرژی آن ها را نیز در همان باند ها حساب کرده ایم.

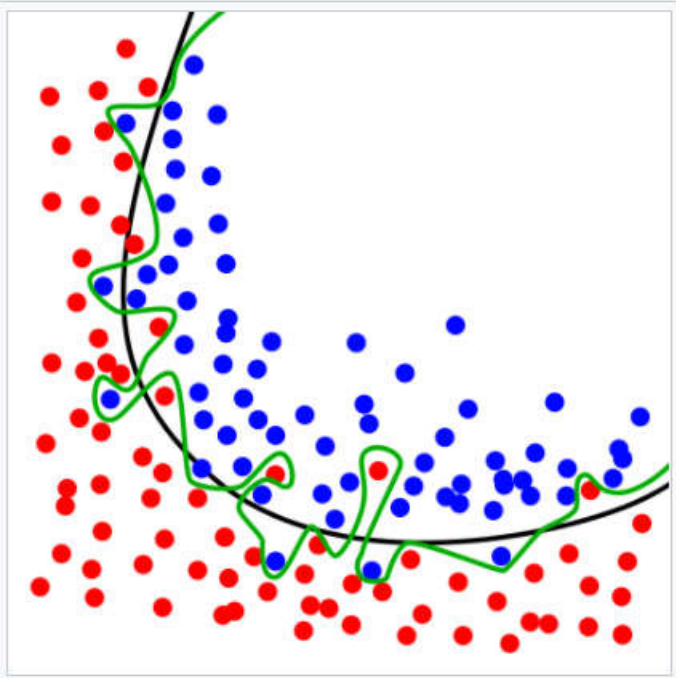
- ویژگی دیگری که استفاده شده است، مقدار انرژی بر فرکانس است.این ویژگی را میتوان با PSD و یا DWT پیاده سازی کرد.این تابع یا الگوریتم، داده را به پنجره های مشخصی در زمان جدا میکند و انرژی سیگنال تقسیم بر فرکانس در همسایگی آن پنجره را به ما میدهد.

نکته قابل بحث،تعداد ویژگی ها برای هر فرد میباشد.بسیاری از ویژگی های بالا(تا قبل DST) برای هر کلاس هر فرد، یک ماتریس 20*64 میباشند.این که آیا اکنون ما 64 ویژگی داریم یا 20*64 کمی مشکل برانگیز بود.البته با توجه به توضیحات تابع خوشه بند و تابع svmclassify معلوم شد که ورودی این توابع باید ویژگی هایی باشند که چندین بار مشاهده شده اند.پس برای ادامه کار،فرض بر این میباشد که ما برای هر کلاس دارای 64 ویژگی میباشیم که هر ویژگی 20 بار مشاهده شده اند(البته 2 کلاس از 2 فرد دارای 21 فعالیت بودند که با توجه به توضیحات دستور کار، از این 2 سری فعالیت صرف نظر کردم)

البته این عدد 64 در برخی ویژگی ها مانند DCT, DSTبه صورت 64 *360 میباشد.پس تعداد ویژگی های ما زیاد میباشد و از همین جا میتوان به این ایده پشت

j-value برای کاهش تعداد ویژگی ها پی برد.از طرفی اگر تعداد ویژگی ها زیاد باشد، مقدار زمان پردازش بالا میرود و از آن مهم تر، اگر تعداد ویژگی ها زیاد باشد تابع خوشه بند در جداسازی داده ها دچار مشکل میشود و نمیتواند داده ها را به صورت درست تقسیم بکند.این مشکل با عنوان overfitting در خوشه بندی به وجود میاید.به این صورت که اگر تعداد ویژگی ها در آموزش خوشه بند بسیار زیاد باشد، مرز بین 2 خوشه پیچیده میشود و اگرچه برای این داده ها بهتر جواب میدهد اما با تغییر داده ها، خطا زیاد میباشد.

https://en.wikipedia.org/wiki/Overfitting



مثلا در شکل بالا،اگرچه خط سبز تمایز بهتری بین داده ها میدهد اما با تغییر داده،خطای بزرگتری دارد و خط سیاه از آن بهتر میباشد.

در باب ویژگی های خوب البته میتوان اکنون نظر داد که ویژگی های حوزه فرکانس بهتر میباشند.برای مثال یک نمونه از مقالات موجود که در حین مقاله این موضوع را بررسی کرده است،مقاله زیر میباشد

Comparison of Features for Movement Prediction from Single-Trial Movement-Related Cortical Potentials in Healthy Subjects and Stroke Patients

Ernest Nlandu Kamavuako,[1] Mads Jochumsen,[1] Imran Khan Niazi,[1,2,3] and Kim Dremstrup[1]

در مقاله مذکور آورده شده است که ویژگی های فرکانسی برای جدا کردن حرکت ارادی از نویز احتمالی بهتر میباشند.در این آزمایش، خطای ویژگی های فرکانسی تنها 3.4 درصد بوده در حالی که ویژگی های آماری حوزه زمان خطایی در حدود 15 درصد را دارند.

البته در مراحل محاسبه j-value برای تمامی ویژگی ها (به جز همبستگی) محاسبه شده است.از طرفی میدانیم که همبستگی ویژگی بسیار مناسبی نمیباشد،چرا که تنها مقدار شباهت 2 کانال را نشان میدهد ولی هیچ معیاری از شباهت را مشخص نمیکند.

به محاسبه j-value های متناظر با ویژگی ها میپردازیم.همانگونه که در فرمول موجود در دستور کار آمده است،به میانگین و واریانس ویژگی ها نیاز داریم.از آنجایی که هر ویژگی را ما 20 بار مشاهده آن خاصیت آن کانال در نظر گرفتیم پس ورودی های تابع نوشته شده برای j-value هر یک،یک ستون از این ویژگی میباشد،یعنی 1*20.

میانگین کل را،میانگین کل داده های آن ویژگی در 4 کلاس مذکور حرکت گرفتم.

از آنجایی که 4 کلاس داریم،با صرف نظر از حالتی که یک کلاس را 2 بار انتخاب کنیم، 6 حالت برای هر ویژگی مذکور برای محاسبه j-value داریم.j-value نهایی متناظر با این ویژگی را برابر با میانگین این 6 عدد میگیریم.

j-value ها اعدادی مثبت از مرتبه های کوچک تا اعدادی در حدود 2 و3 میباشند.میدانیم که بزرگ بودن j-value یعنی بیشتر متفاوت بودن یک ویژگی و مناسب بودن آن ویژگی برای جدا کردن داده ها از هم.بعد از چند بار تست(و اندکی گریز به مفهوم توزیع گوسی) داده هایی که در بیرون بازه mean(j-value)+4std(j-value) افتادند را نگه داشتیم و بقیه داده ها را صفر کردیم(برای راحتی از بین بردن داده ها،در اینجا داده ها را صفر میکنیم).بازه مذکور برابر است با داده های بزرگتر از میانگین کل داده ها + 4 برابر انحراف از معیار کل داده ها.

علت این انتخاب را علاوه بر آزمایش شاید بتوان در توزیع گاوسی دید که احتمال اینکه داده ای در فاصله ای بیشتر از 3 برابر انحراف معیار از میانگین بیافتد کمتر از 0.003 میباشد.پس اگر داده ای در این بازه باشد بسیار نادر بوده و برای تمییز بین ویژگی ها مناسب میباشد.

آموزش طبقه بندی کننده

در این مرحله از تابع svmtrain و svmclassify استفاده شده است.برای استفاده از حالت 4 کلاسی، تابع multisvm را نوشته ام که به پیوست میباشد.این تابع با گرفتن داده های آموزش دهنده و 4 کلاس و داده ی تست،تست را در صورت امکان به یکی از 4 گروه اختصاص میدهد و اگر به هیچ گروهی اختصاص ندهد و یا به تعداد بیشتر از یک گروه اختصاص بدهد،ارور میدهد.

برای k-fold cv در متن کد ی تعبیه شده که اگر مفهوم k-fold را درست فهمیده باشم، به صورت زیر به ازای k=5 پیاده سازی شده است:

برای هر فرد، 4 بار ماتریسی رندم 80 تایی به وجود می آوریم. این ماتریس را به 5 قسمت مساوی تقسیم میکنیم و به روشی این اعداد رندم را به اعداد 1 تا 80 تبدیل میکنیم.(به خاطر 4 کلاس داده،داده را به 5 قسمت تقسیم میکنیم و 4 قسمت را تحت عنوان آموزش و قسمت آخر را برای تست داده ها نگه میداریم) هر بار 4 قسمت برای آموزش را به تابع svmtrain میدهیم و هر بار یکی از 16 داده ی تستی که نگه داشته ایم را به آن میدهیم.جواب ها را نگه داشته و در آخر سر برای هر فرد میانگین تعداد جواب های درست را محاسبه میکنیم.

(1046 به ازای بازه اطمینان 3 انحراف معیار و 754 به ازای بازه اطمینان 4 انحراف معیار میباشد.)

مثلا در زیر، خروجی به ازای 1046 ویژگی آمده است:

```
for each person, the probability of correct answer is:
   19.6875
   23.7500
   25.6250
   33.1250
   34.6875
   19.3750
   46.8750

the total percentage of success is 29.017857
```

به ازای یک امتحان دیگر از همان تعداد ویژگی نیز داریم:

```
for each person, the probability of correct answer is:
   21.2500
   25.0000
   28.1250
   34.6875
   38.7500
   19.0625
   48.1250

the total percentage of success is 30.714286
```

خروجی k-fold برای سری ویژگی های بالا به صورت زیر میباشد:

for each person, the probability of correct answer is:
0.1875
0.2469
0.2625
0.3281
0.3937
0.1969
0.4531

for each person, the variance of different cross validation
probabilities of success is:
0.2422
0.0576
0.0547
0.0264
0.0820
0.0264
0.0732

the total probability of success:0.295536 , error is 0.704464
mean variance of success: 0.080357

به ازای 754ویژگی به صورت زیر است:

for each person, the probability of correct answer is:
20.6250
21.5625
25.9375
34.3750
39.6875
27.1875
44.6875

the total percentage of success is 30.580357

یک امتحان دیگر:

for each person, the probability of correct answer is:
20.9375
21.2500
23.1250
35.0000
33.1250
25.6250
43.7500

the total percentage of success is 28.973214

به طور میانگین به جواب 30 درصد میرسیم.

در نهایت البته 2 سری ویژگی دیگر نیز اضافه شد که این 2 ویژگی داده ها را به صورت خیلی خوب تمییز داد.این ۲ ویژگی یکی mobility و دیگری complexity داده بودند.این 2 ویژگی هر دو جزو ویژگی های Hjorth میباشند.

https://en.wikipedia.org/wiki/Hjorth_parameters

Mobility بنا بر تعریف برابر است با ریشه ی دوم نسبت واریانس مشتق سیگنال به واریانس سیگنال.یعنی:

$$Mobility = \sqrt{\frac{var(\frac{dy(t)}{dt})}{var(y(t))}}.$$

Complexity نیز برابر است با نسبت موبیلیتی مشتق به موبیلیتی خود سیگنال.یعنی:

$$Complexity = \frac{Mobility(\frac{dy(t)}{dt})}{Mobility(y(t))}.$$

این ها درصد درست بودن را به صورت میانگین به 43 درصد افزایش داد.داده های نهایی با این سری ویژگی ها فیلتر شده اند. خروجی k-fold به شکل زیر است:

for each person, the probability of correct answer is:
0.3094
0.4125
0.3969
0.4844
0.4313
0.4375
0.5781

for each person, the variance of different cross validation probabilities of success is:
0.2061
0.1953
0.1123
0.1514
0.0039
0.0547
0.0264

the total probability of success:0.435714 , error is 0.564286
mean variance of success: 0.107143

همانطور که مشاهده میشود احتمال خروجی درست افزایش یافته است اما در همین حال مقدار واریانس نیز تا حد خیلی خوبی افزایش یافته(بیشتر از 2 برابر شده است).

با چند آزمون و تست،از تعداد داده های اضافه شده تعدادی را حذف میکنیم. خروجی k-fold cv برای ویژگی هایی که با آنها داده ی نهایی خوشه بندی شده اند به صورت زیر است:

for each person, the probability of correct answer is:
0.2469
0.4000
0.3625
0.4969
0.4125
0.4406
0.5906

for each person, the variance of different cross validation
probabilities of success is:
0.1904
0.0547
0.0391
0.0186
0.0234
0.0811
0.0420

the total probability of success:0.421429 , error is 0.578571
mean variance of success: 0.064174

در این سری ویژگی ها،احتمال درست بودن در همان حد میباشد اما واریانس کمتری را شاهد هستیم.

طبقه بندی داده های آزمون

اکنون همه 80 داده ی Train را به تابع svmtrain میدهیم و داده های test را یکی یکی به داده و خروجی را میگیریم.خروجی ها در ماتریس test_class به پیوست میباشند.در این ماتریس اعداد 1 تا 4 هر یک متناظر با هر یک از کلاس های 1 تا 4 دقیقا متناظر با شماره بندی دستور کار میباشند.خروجی های NaN ناشی از 2 حالت میباشندُیا این که تابع خوشه بند نتوانسته این داده را در هیچ یک از گروه ها قرار بدهد و یا در بیشتر از یک گروه قرار داده است.که هر 2 حالت،ناخواسته میباشند.

در داده های تست نیز اندکی مشکل وجود دارد و آن این که برخی از افراد به جای 49 بار تست،48 بار تست دارند.برای این افراد نیز خروجی 49 ستون داشته است که ستون 49 فاقد معنی میباشد. این افراد عبارتند از:

4,6 ,7

یعنی سه سطر آخر ماتریسی که به پیوست هستند،تنها دارای 48 ستون با معنی میباشند.

لازم به ذکر است که ترتیب افراد موجود در ماتریس test_class به صورت صعودی نیست و به گونه ای بوده است که خود متلب این داده ها را load کرده است.این ترتیب به صورت زیر است:

11
12
16
3
4
6
7

در پیوست،2 ماتریس سری فایل mat. موجود است.فایل test همان ماتریس بالا.فایل subjects_classified در واقع هر یک از سطر های ماتریس بالا میباشد که ستون آخر 3 فرد ذکر شده که 48 بار تست شده بودند نیز حذف شده است.

End of File