



باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

پردازش سیگنال‌های گرافی

استاد: دکتر امینی

## تمرین کامپیوتری سری سوم

در این تمرین قصد داریم با برخی از روش‌های یادگیری گراف یا فشرده کردن گراف آشنا شویم و کاربردی از آن‌ها را ببینیم.

برای این تمرین، یکی از ۲ سوال زیر را با توجه به علاقه خود انتخاب کنید و حل کنید.

**نکته:** در شبیه‌سازی‌های سوال ۲، جهت مقایسه بهتر نتایج الگوریتم‌های یادگیری گراف، ماتریس وزن بدست آمده در هر بخش را به گونه‌ای نرمالیزه کنید که بیشینه وزن یال‌ها برابر واحد باشد. همچنین می‌توانید برای مقایسه نتایج بخش‌های مختلف، ماتریس وزن نرمالیزه را با دستور `imagesc` در متلب نمایش دهید.

## ۱ Multiscale Pyramid Transform

## ۱.۱ پیاده‌سازی توابع مورد نیاز

۱) روش `polarity of the largest eigenvector` برای انتخاب گره مناسب برای `downsampling` را به صورت خلاصه توضیح دهید. سپس تابعی به نام `MyVertexSelection` تعریف کنید که با دریافت یک گراف در ورودی، اندیس گره‌هایی را که این الگوریتم انتخاب می‌کند را برگرداند.

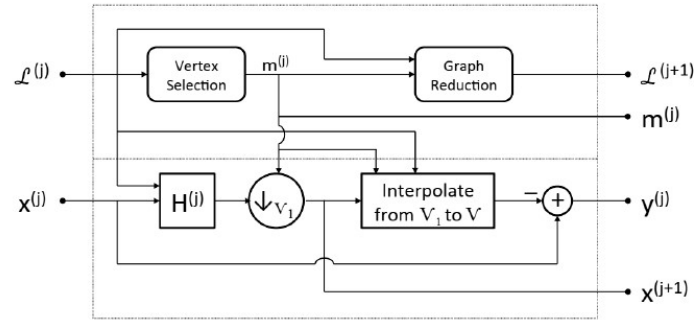
۲) عملیات `sparsified Kron reduction` را به صورت خلاصه توضیح دهید. سپس تابعی به نام `MySKReduction` تعریف کنید که با دریافت گراف در ورودی و شماره اندیس‌های مناسب `V1`، گراف را با روش گفته شده کاهش داده و گراف کاهش یافته را بازگرداند (می‌توانید یک `struct` برگردانید یا ماتریس لاپلاسین گراف را).

۳) تابعی به نام `MyHfilter` تعریف کنید که یک سیگنال گرافی و گراف متناظر را در ورودی دریافت و سیگنال را با فیلتر `H` فیلتر کرده و سیگنال فیلترشده را بازگرداند. فیلتر `H` را یک فیلتر `lowpass` با پاسخ فرکانسی  $h(\lambda) = 1/(1 + 2\lambda)$  در نظر بگیرید.

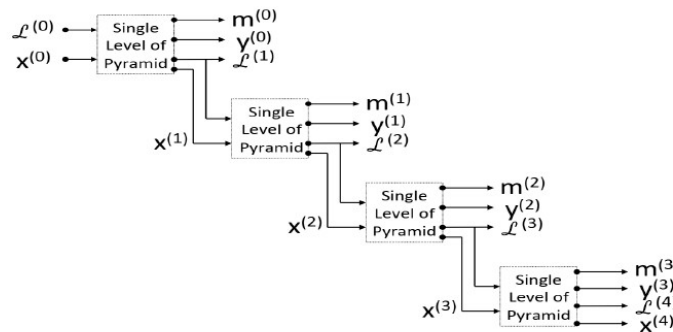
۴) تابعی به نام `MyDS` تعریف کنید که یک سیگنال گرافی و مجموعه‌ی گره‌های `V1` مدنظر در `downsampling` را دریافت کرده و سیگنال `downsamp` شده را بازگرداند.

۵) تابعی به نام `MyInterpolate` تعریف کنید که یک سیگنال `downsamp` شده، اندیس گره‌های `V1` و گراف موردنظر را دریافت کند و سپس سیگنال را بر روی کل گراف `Interpolate` کند. یک `flag` نیز در ورودی تعریف کنید که در صورت صفر بودن از روش `Interpolate` توضیح داده شده در بخش `V-C` مقاله، و در صورت یک بودن از روش بازسازی معرفی شده در کلاس استفاده کند.

۶) به کمک توابعی که تعریف کرده‌اید، تابعی با نام `MyAnalysis` بنویسید که مطابق بلوک دیاگرام زیر، یک گراف و سیگنال گرافی تعریف شده بر روی آن را دریافت کرده و در خروجی اش گراف کاهش یافته، اندیس گره‌های مناسب برای `downsampling` بدست آمده، سیگنال کاهش یافته و خطای `Interpolate` را بازگرداند.



۷) تابعی با نام MyPyramidAnalysis تعریف کنید که در ورودی خود یک گراف، سیگنال تعریف شده بر روی آن و عدد  $N$  را دریافت کرده و سپس بر روی گراف عملیات Multiresolution را مطابق با بلوک دیاگرام زیر اعمال کند. تابع شما بایستی در خروجی خود، برای هریک از مجموعه (ماتریس لاپلاسیان) گراف های کاهش یافته  $(L^{(i)})$ ، مجموعه سیگنال های خطای Interpolate  $(y^{(i)})$ ، مجموعه اندیس های انتخاب شده از گراف مرحله قبل  $(m^{(i)})$ ، و مجموعه سیگنال های downsample شده  $(x^{(i)})$ ، یک cell array با سایز  $N$  بازگرداند.



۸) تابعی با نام MySynthesis تعریف کنید که عملیات سنتز یک مرحله ای را از روی  $L^{(i)}$  و  $y^{(i)}$  و  $m^{(i)}$  انجام داده و در خروجی،  $x^{(i)}$  را بازگرداند.

۹) تابعی با نام MyPyramidSynthesis تعریف کنید که عملیات سنتز  $N$  مرحله ای را انجام دهد. تابع شما بایستی cell array های گراف ها، خطاها، اندیس های مناسب را به همراه آخرین سیگنال کاهش یافته  $(x^{(N+1)})$  دریافت کرده و گراف اولیه و سیگنال اولیه بر روی آن را بازسازی کند.

## ۲.۱ اعمال توابع آنالیز و سنتز بر روی دیتاست

۱۰) فایل Data\_city.csv که شامل مختصات جغرافیایی شهرهای ایران است را بخوانید. به کمک تابع getDistance.m که در اختیاران قرار داده شده است، فاصله ی دو به دوی شهرهای مراکز استان ها را (که با "primary" و "admin" در ستون هفتم مشخص شده اند) بیابید. با روش مناسب این فواصل را به بازه ی بین صفر و یک برده و گراف متناظر بین این شهرها را تشکیل دهید. توجه کنید که هرچه دو شهر به یکدیگر نزدیک تر باشند وزن یال بین آن ها بایستی به یک نزدیک تر باشد و برعکس. گراف بدست آمده را رسم کنید. برای مختصات گره ها در گراف میتوانید از مختصات گراف sensor استفاده کنید و یا می توانید مختصات هر شهر را در صفحه ی x-y بدست آورده و مطابق با نقشه رسم کنید.

۱۱) سیگنال دمای شهرهای بخش قبل را از ستون Temp دیتاست خوانده و بر روی گراف آن را تعریف کنید. بر روی این سیگنال عملیات MyPyramidAnalysis را با  $N = 3$  مرحله اعمال کنید و سیگنال های کاهش یافته و خطای آن ها را بر روی گراف های کاهش یافته نمایش دهید. همچنین مجموع توان دوی خطای هر مرحله را بر روی کل گراف محاسبه کرده و برای هر مرحله از کاهش گزارش کنید. میانگین این سه خطا را در دو حالت استفاده از دو روش Interpolate (ذکر شده در قسمت ۵ بخش ۱) مقایسه کرده و بیان کنید کدام روش بهتر عمل می کند.

۱۲) حال به کمک خروجی های بخش قبل و به کمک آخرین سیگنال کاهش یافته، با استفاده از تابع MyPyramidSynthesis سیگنال اولیه را بازسازی و با سیگنال داده شده مقایسه کرده و از صحت عملکرد آنالیز و سنتز توابع خود اطمینان حاصل کنید. همچنین هر دو سیگنال اولیه و بازسازی شده را بر روی گراف نمایش دهید.

۱۳) فرض کنید از سیگنال مرحله ی آخر آنالیز به عنوان feature های سیگنال اولیه استفاده کنیم. در صورتی که این feature ها را انتقال دهیم اما در حین انتقال بر روی آن نویز سفید گوسی با  $SNR = 5dB$  اضافه شود، سیگنال سنتز شده نسبت به سیگنال اولیه چقدر خطا خواهد داشت؟ خطا را به صورت MSE محاسبه کنید.

## ۳.۱ بررسی معادله‌ی انتشار

(۱۴) با فرض سیگنال بالا به عنوان دما در روز اول ماه و با فرض ثابت انتشار به صورت  $c=0, 0.1$  دمای این شهرها را در ۵ روز متوالی و همچنین در زمان بینهایت بدست آورده و بر روی گراف نمایش دهید. هر روز را معادل با یک واحد زمانی در نظر بگیرید. همچنین ۶ نمودار مستطیلی برای هر روز و نیز روز بینهایت رسم کنید که در هر نمودار دمای ۳۱ شهر نمایش داده شده باشد.

## ۲ آشنایی با پیاده سازی الگوریتم های یادگیری گراف

دیتای موردنظر، با نام Data.mat در اختیار شما قرار گرفته است. این دیتا مربوط به ۳۳ گونه جانوری می باشد که قصد داریم میزان شباهت این ۳۳ گونه را با گرافی وزن دار نمایش دهیم. برای اینکار هر گونه را بایستی به عنوان یک رأس در نظر بگیریم و گراف بین این رأس ها را بدست بیاوریم. فایل Data.mat شامل موارد زیر می باشد:

• **name** : که نام ۳۳ گونه جانوری در آن ذخیره شده است.

• **data** : که همان ماتریس  $X$  در مبحث یادگیری گراف است و برای هر گونه جانوری شامل ۱۰۲ ویژگی باینری است.

• **features** : که بیان میکند هر ستون از data مربوط به چه ویژگی ای می باشد. به بیان دیگر، بر روی ۳۳ رأس موردنظرمان، ۱۰۲ نمونه (اندازه گیری) از سیگنال گرافی داریم که در ماتریس data ذخیره شده است.

در هر بخش، پس از یادگیری گراف و به کمک ماتریس لاپلاسین گراف و نام گونه ها (names) از تابع draw\_animal\_graph.m موجود در فولدر draw برای رسم گراف یادگیری شده استفاده کنید. توجه کنید که این تابع برای اجرا به توابع دیگر موجود در فولدر draw نیز نیاز دارد.

همچنین تولباکس GSPBOX در توابع یادگیری گراف خود به ۴ تابع دیگر نیز نیاز دارد که در فایل های آن موجود نیست. این ۴ تابع در فولدر learn\_graph در اختیار شما قرار داده شده است. آن ها را به فولدر learn\_graph موجود در فولدر GSPBOX اضافه کنید تا توابع یادگیری گراف GSPBOX اجرا شوند. در تمامی بخش ها می توانید از ترشولدینگ جهت اسپارس کردن ماتریس  $W$  گراف استفاده کنید.

(۱) یکی از رویکردها برای یادگیری گرافی منطبق بر داده، مبتنی بر فرضهمواری سیگنال گرافی است. بر همین مبنا دو الگوریتم در تولباکس GSPBOX قرار داده شده است که بر اساس ماتریس فاصله  $Z$  و با در نظر گرفتن انتخاب هایی مختلف برای  $f(W)$  یادگیری گراف را انجام می دهند. این توابع عبارتند از gsp\_learn\_graph\_log\_degrees و gsp\_learn\_graph\_l2\_degrees.

(۲) حال به کمک این توابع گراف مربوط به دیتای موردنظر را با پارامترهای  $a = b = 5$  بدست آورده و نمایش دهید (می توانید از توابع موجود در این تولباکس برای محاسبه ماتریس فاصله  $Z$  استفاده کنید).

(۳) با ثابت نگه داشتن یکی از پارامترهای  $a$  یا  $b$  و سپس تغییر پارامتر دیگر، نتایج مختلف را بدست آورده و نمایش دهید. نقش این پارامترها چیست؟

(۴) الگوریتم CGL برای یادگیری گراف را از مقاله Reference.pdf مطالعه کرده و آن را مختصرا توضیح دهید. این روش چه نوع ماتریسلاپلاسینی برای گراف بدست می آورد؟ نقش هر جمله در تابع هزینه چیست؟ ماتریس های  $H$ ،  $J$  و  $S$  چیست و چگونه محاسبه می شود؟

(۵) کد الگوریتم های این مقاله را از لینک زیر دانلود کنید

[https://github.com/STAC-USC/Graph\\_Learning](https://github.com/STAC-USC/Graph_Learning)

به کمک تابع estimate\_cgl.m موجود در این تولباکس و به ازای آلفای ۰.۰۸ و ۰.۰۱ گراف مربوط به دیتای موردنظر را بدست آورده و نمایش دهید. نقش پارامتر آلفا چیست؟