پردازش سیگنال گرافی (۲۵۱۵۰) **پاییز ۱۴۰۰**

تمرین کامپیوتری امتیازی

مدرس : دکتر آرش امینی

در این تمرین قصد داریم با برخی از روش‌های یادگیری گراف آشنا شویم و از آنها در کاربرد پردازش و دسته‌بندی پرس و جوهای ورودی به یک موتور جستجو استفاده کنیم.

نکته : در شبیه سازی های زیر، جهت مقایسه بهتر نتایج الگوریتم های یادگیری گراف، ماتریس وزن بدست آمده در هر بخش را به گونه ای نرمالیزه کنید که بیشنه وزن یال ها برابر واحد باشد. همچنین می توانید برای مقایسه نتایج بخش های مختلف، ماتریس وزن نرمالیزه را با دستور imagesc در متلب نمایش دهید.

بخش اول : پیاده‌سازی الگوریتم‌های یادگیری گراف

دیتای موردنظر، با نام `Data.m` در اختیار شما قرار گرفته است. این دیتا مربوط به ۳۳ گونه جانوری می‌باشد که قصد داریم میزان شباهت این ۳۳ گونه را با گرافی وزن‌دار نمایش دهیم. برای اینکار هر گونه را بایستی به عنوان یک رأس درنظر بگیریم و گراف بین این رأس‌ها را بدست بیاوریم. فایل `Data.m` شامل موارد زیر می‌باشد:

names: که نام ۳۳ گونه جانوری در آن ذخیره شده است.

data: که همان ماتریس $\mathbf{X}$ 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 ویژگی‌ای می‌باشد. 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 گونه‌ها (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 فایل‌های 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 بخش‌ها می‌توانید از ترشولدینگ جهت اسپارس کردن ماتریس $\mathbf{W}$ گراف استفاده کنید.

۱) یکی از رویکردها برای یادگیری گرافی منطبق بر داده، همانطوری که در سؤال ۲ تمرین تئوری چهارم دیدیم، مبتنی بر فرض همواری سیگنال گرافی است. بر همین مبنا دو الگوریتم در تولباکس GSPBOX قرار داده شده است که بر اساس ماتریس فاصله $\mathbf{Z}$ و با در نظر گرفتن انتخاب هایی مختلف برای $f(\mathbf{W})$ یادگیری گراف را انجام می‌دهند. 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 $a = b = ۵$ بدست آورده و نمایش دهید (می‌توانید از توابع موجود در این تولباکس برای محاسبه ماتریس فاصله $\mathbf{Z}$ 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 ماتریس لاپلاسینی برای گراف بدست می‌آورد؟ نقش هر جمله در تابع هزینه چیست؟ ماتریس‌های $\widetilde{\mathbf{K}}$، $\mathbf{S}$، $\mathbf{J}$ و $\mathbf{H}$ چیست و چگونه محاسبه می شود؟

۵) کد الگوریتم‌های این مقاله را از لینک زیر دانلود کنید

https://github.com/STAC-USC/Graph_Learning

۶) به کمک تابع estimate_cgl.m موجود در این تولباکس و به ازای آلفای ۰.۰۸ و ۰.۰۰، گراف مربوط به دیتای موردنظر را بدست آورده و نمایش دهید. نقش پارامتر آلفا چیست؟

بخش دوم : انتخاب توکن های برگزیده ی یک پرس و جو

در این مسئله می خواهیم، یک query (پرس و جو) را به صورت یک گراف نمایش دهیم و با استفاده از این گراف، query را خلاصه کنیم.

۱. تابعی به نام query2graph پیاده کنید که یک ماتریس به عنوان ورودی بگیرد که سطرهای آن در واقع بردار توکن های یک query است و گراف آن query را به صورت زیر تشکیل دهد و آن را به عنوان خروجی برگرداند :
هر رأس این گراف متناظر با یکی از توکن های این کوئری است و وزن یال های آن را باید به نحو مناسب با استفاده از cosine similarity بین بردار این توکن ها تشکیل دهید. می‌توانید برای این منظور از الگوریتم‌های بخش قبل استفاده کنید.

۲. در فایل queries.csv تعدادی query قرار داده شده اند. در ستون tokens، لیست توکن های کوئری آمده است. در فایل tokens_of_queries.mat یک cell array قرار دارد که cell شماره i آن یک ماتریس ۲ بعدی است که مربوط به کوئری i ام است و سطرهای این ماتریس بردار توکن های آن کوئری است. فایل tokens_of_queries.mat را بخوانید و با استفاده از تابع query2graph برای این query ها گراف نظیرشان را تشکیل دهید و گراف آن ها را نمایش دهید.

۳. Community Detection Toolbox را که یک ابزار MATLAB است، نصب کنید. (در لینک زیر می توانید documentaion های مربوط به این ابزار را مشاهده کنید:

https://www.mathworks.com/matlabcentral/fileexchange/45867-community-detection-toolbox

همچنین فایل مربوط به این تولباکس با نام ComDetTBv090.zip در اختیار شما قرار داده شده است.

۴. با استفاده از تابع GCModulMax1.m موجود در پوشه Algorithms ابزار Community Detection Toolbox، گراف هر query را خوشه بندی کنید و سپس از هر خوشه یک رأس را به عنوان نماینده انتخاب کنید. (معیار انتخاب رأس نماینده در هر خوشه را میزان شباهت (cosine similarity) بردار آن توکن با بردار خود کوئری در نظر بگیرید. در فایل queries.mat یک cell array قرار دارد که cell شماره i آن بردار کوئری iام است.)
حال برای هر query لیست توکن های انتخاب شده ی آن و هم چنین لیست توکن های موجود در خوشه های مختلف آن را تشکیل و نمایش دهید.
تابع GCModulMax1 با استفاده از روشی موسوم به community detection گراف را خوشه بندی می کند. در ابزار Community Detection Toolbox توابع دیگری هم برای خوشه بندی کردن گراف وجود دارد که می توانید آن ها را هم امتحان کنید.

۵. هم چنین با استفاده از Multiscale Pyramid Transform که در تمرین کامپیوتری سری سوم آن را پیاده کردید، تعدادی رأس برای هر کدام از گراف های کوئری ها انتخاب کنید و لیست توکن های انتخاب شده برای هر query و هم چنین لیست توکن های موجود در خوشه های مختلف آن را نمایش دهید. برای این کار، سیگنال گرافی $\mathbf{x}$ را میزان شباهت هر توکن با کل query در نظر بگیرید. برای این کار از میزان شباهت (cosine

similarity) بردار هر توکن با بردار خود کوئری استفاده کنید. برای تعداد تکرار مراحل (N)، مقدار مناسبی قرار دهید.

به طور شهودی از روی نتایج به دست آمده، به نظرتان از بین روش community detection و Multiscale Pyramid Transform کدام یک در خلاصه کردن کوئری بهتر عمل کرده اند؟

بخش سوم : rerank نتایج مربوط به یک پرس و جو

در این مسئله قصد داریم با استفاده از مفاهیم مربوط به GSP عمل rerank نتایج مربوط به یک پرس و جو را انجام دهیم؛ یعنی از بین تعداد زیادی سند مرتبط با یک پرس و جو، یک زیر مجموعه ی کوچک آن ها را به عنوان نتایج نهایی انتخاب کنیم با این هدف که اولا اسناد این زیرمجموعه شباهت زیادی به پرس و جو داشته باشند و ثانیا تنوع زیادی داشته باشند و به عبارتی تا حد خوبی تنوع مجموعه ی اسناد اولیه را حفظ کرده باشند.

۱. در پوشه ی docs تعدادی سند مرتبط با پرس و جوی اول موجود در فایل `queries.csv` قرار دارند. در فایل `docs.mat` یک cell array قرار دارد که cell شماره ی i آن بردار سند i-ام است. با استفاده از cosine similarity بردار اسناد، شباهت اسناد با یکدیگر را به دست آورید و به این وسیله، گراف اسناد را تشکیل دهید.

۲. query اول فایل queries.csv را با متن اسنادsimilarity بگیرید (به معنای cosine similarity بردار این کوئری و بردار هر سند) و میزان similarity را به عنوان سیگنالی روی گراف در نظر بگیرید.

۳. با استفاده از Multiscale Pyramid Transform از گراف اسناد بر اساس سیگنالsimilarity، تعدادی رأس (سند) انتخاب کنید (N را طوری تعیین کنید که تعداد اسناد انتخاب شده بین ۱۰ تا ۲۰ سند باشد). اسناد انتخاب شده را مشاهده کنید؛ آیا از لحاظ تنوع اسناد و شباهت آن ها با پرس و جو، اسناد خوبی انتخاب شده اند؟