

روش های خوشه بندی مبتنی بر هسته

پروژه درس یادگیری ماشین ، دکتر کتان فروش

مقاله انتخابی:

A Tutorial on Spectral Clustering

Ulrike von Luxburg

Max Planck Institute for Biological Cybernetics Spemannstr.

38, 72076 Tübingen, Germany

تهیه کننده : تحسین ایلخااص زاده

شماره دانشجویی : ۴۰۰۴۲۲۰۳۴

دانشکده علوم ریاضی، دانشگاه شهید بهشتی

چکیده

روش های خوشه بندی مبتنی بر هسته^۱ یک ابزار قدرتمند برای یادگیری بدون نظارت داده های غیرخطی جدایی پذیر هستند. در سال های اخیر، خوشه بندی طیفی به یکی از محبوب ترین الگوریتم های خوشه بندی مدرن تبدیل شده است. از آنجا که پیاده سازی آن ساده است، می توان آن را با نرم افزار جبر خطی استاندارد به طور موثری حل کرد و اغلب از الگوریتم های خوشه بندی سنتی مانند الگوریتم k-means بهتر عمل می کند.

کلمات کلیدی: روش های خوشه بندی مبتنی بر هسته ، خوشه بندی طیفی، گراف لاپلاسی

۱ مقدمه

خوشه بندی مبتنی بر هسته و خوشه بندی طیفی هر دو برای شناسایی خوشه هایی که در فضای ورودی ، خطی جدایی ناپذیر هستند استفاده شده اند. تجزیه و تحلیل داده های بدون نظارت با استفاده از الگوریتم های خوشه بندی ابزار مفیدی برای کشف ساختارهای داده فراهم می کند. در حقیقت ، خوشه بندی یک ابزار پیشرفته برای کشف دانش است و در زمینه های مختلف از جمله طبقه بندی، پردازش تصویر، داده کاوی، تشخیص الگو، بینایی کامپیوتری، تقسیم بندی تصویر ، بازیابی اطلاعات و غیره کاربرد دارد.

خوشه بندی به معنای سازمان دهی مجموعه ای از الگوها در خوشه ها است، به طوری که الگوهای درون یک خوشه معین دارای درجه بالایی از شباهت باشند، در حالی که الگوهای متعلق به خوشه های مختلف دارای درجه بالایی از عدم تشابه هستند. یکی از جنبه های

^۱kernel k-means clustering

مهم در خوشه بندی، نمایش الگو و اندازه گیری شباهت است. هر الگو معمولاً با مجموعه ای از ویژگی های سیستم مورد مطالعه نشان داده می شود. توجه به این نکته بسیار مهم است که انتخاب خوب نمایش الگوها می تواند منجر به بهبود عملکرد خوشه بندی شود. اینکه آیا امکان انتخاب مجموعه ای مناسب از ویژگی ها وجود دارد یا خیر به سیستم مورد مطالعه بستگی دارد. هنگامی که یک نمایش ثابت شد، می توان یک معیار شباهت مناسب را از بین الگوها انتخاب کرد. محبوب ترین معیار عدم تشابه برای نمایش های متریک فاصله است، برای مثال معیار اقلیدسی .

در یادگیری ماشین، در سال ۱۹۶۴ استفاده از توابع هسته [۵] توسط آیزرمن و همکاران معرفی شده است. [۱] . در سال ۱۹۹۵ کورتز و وپنیک ماشین های بردار پشتیبان ^۲ را معرفی کردند [۴] که عملکرد بهتری نسبت به سایر الگوریتم های طبقه بندی در چندین مسأله دارند. موفقیت SVM باعث گسترش استفاده از هسته ها به سایر الگوریتم های یادگیری (به عنوان مثال، Kernel PCA [۶]) شده است. انتخاب هسته برای گنجاندن دانش پیشین در برنامه بسیار مهم است، که برای آن امکان طراحی کرنل های موقت وجود دارد.

روش های خوشه بندی مرسوم مبتنی بر هسته (K-means مبتنی بر هسته و K-means فازی مبتنی بر هسته و غیره) وزن مربوط به متغیرها را در نظر نمی گیرند، این روش ها در نظر می گیرند که همه متغیرها به یک اندازه با فرآیند خوشه بندی مرتبط هستند به این معنا که همه وزن مرتبط یکسانی دارند. با این حال، در بیشتر زمینه ها معمولاً باید با مجموعه داده های با ابعاد بالا سر و کار داشته باشیم. بنابراین، برخی از متغیرها ممکن است نامربوط باشند و در میان متغیرهای مربوط، برخی ممکن است دارای مقدار بیشتر یا کمتر باشند. [۳]

۱.۱ دسته بندی الگوریتم های خوشه بندی

با توجه به روش های خوشه بندی مبتنی بر هسته، چندین روش خوشه بندی با ترکیب هسته ها اصلاح شده اند (روش های به عنوان مثال، Neural Gas و SOM، Fuzzy c-Means، K-Means). استفاده از هسته ها اجازه می دهد تا به طور ضمنی داده ها را در فضایی با ابعاد بالا به نام فضای ویژگی نگاشت کنید. محاسبه پارتیشن بندی خطی در این فضای ویژگی، منجر به پارتیشن بندی غیرخطی در فضای ورودی می شود. بطور کلی انواع مختلف خوشه بندی عبارتند از:

- خوشه بندی مبتنی بر اتصال (خوشه بندی سلسله مراتبی) ، ^۳
- خوشه بندی مبتنی بر مرکز (روش های پارتیشن بندی) ، ^۴
- خوشه بندی مبتنی بر توزیع ^۵
- خوشه بندی مبتنی بر تراکم (روش های مبتنی بر مدل) ، ^۶
- خوشه بندی فازی ^۷
- خوشه بندی مبتنی بر محدودیت (خوشه نظارت شده) ^۸

اخيراً برخی از روش های خوشه بندی که ابر صفحه های جداکننده غیرخطی در میان خوشه ها تولید می کنند، پیشنهاد شده اند. این الگوریتم ها را می توان به دو خانواده بزرگ تقسیم کرد:

^۲Support vector machines: SVM

^۳Hierarchical clustering

^۴Partitioning methods

^۵Distribution-based Clustering

^۶Model-based methods

^۷Fuzzy Clustering

^۸Supervised Clustering

- روش‌های خوشه‌بندی طیفی
- روش‌های خوشه‌بندی مبتنی بر هسته.

۱.۲ الگوریتم خوشه‌بندی طیفی

این الگوریتم ابتدا یک ماتریس وابستگی^۹ می‌سازد و با ساخت این ماتریس وابستگی، در واقع مسأله‌ی ما به یک گراف تبدیل می‌شود که اجزای به هم متصل گراف تشکیل یک خوشه را با هم می‌دهند. در واقع در این گراف، یال‌هایی که در یک عناصر آن‌ها در یک خوشه هستند وزن زیادی دارند، و برعکس یال‌هایی که عناصر آن‌ها در یک خوشه نیستند، وزن کمتری را دارند. بعد از آن لاپلاسیان گراف را ایجاد کرده و بردارهای ویژه را برای آن انتخاب می‌کنیم. در آخر با الگوریتمی مانند K-Means از میان بردارهای ویژه می‌توان به خوشه‌بندی‌های مورد نظر دست پیدا کرد. البته در نهایت این الگوریتم نیز نیاز به گرفتن تعداد مورد انتظار خوشه‌ها از کاربر دارد ولی در شرایطی و با استفاده از فاصله‌ی بین بردارهای ویژه، می‌توان تعداد بهینه را برای تعداد خوشه‌ها انتخاب کرد. به این کار به اصطلاح Rounding می‌گویند.

۱.۳ الگوریتم خوشه‌بندی مبتنی بر هسته

این الگوریتم همان ترفند k-means را اعمال می‌کند اما با یک تفاوت که در اینجا در محاسبه فاصله، از روش هسته به جای فاصله اقلیدسی استفاده می‌شود. مزیت اصلی این رویکرد نسبت به روش مرسوم این است که امکان استفاده از فواصل تطبیقی هسته‌ای را فراهم می‌کند که برای یادگیری پویای وزن متغیرها مناسب است و عملکرد الگوریتم‌ها را بهبود می‌بخشد.

بین K-Means مبتنی بر هسته^{۱۰} و خوشه‌بندی طیفی^{۱۱} ارتباط نظری وجود دارد که Dhillon و همکارانش آن را بررسی کرده‌اند. آنها نشان دادند که چگونه خوشه‌بندی طیفی معادل یک نسخه آسان گرفته شده مبتنی بر هسته K-Means برای یک هسته و وزن خاص است. از این رو، می‌توانید از بسته خوشه‌بندی طیفی موجود در sklearn استفاده کنید.

همانطور که قبلاً ذکر شد، در حالی که روش‌های مبتنی بر هسته از نظر محاسباتی ارزان‌تر از نگاشت ویژگی مستقیم هستند، همچنان از نظر محاسباتی فشرده هستند زیرا شما نیاز به محاسبه ماتریس هسته $N \times N$ دارید. در سناریوهایی با نقاط داده زیاد، استفاده از روش‌های مبتنی بر نمودار، مانند خوشه‌بندی طیفی، ممکن است گزینه بهتری باشد.

نکات کلیدی :

- روش‌های مبتنی بر هسته مجموعه ویژگی‌ها را به ابعاد بالاتر گسترش می‌دهند و مرزهای غیر خطی را یاد می‌گیرند.
- تابع هدف K-Means را می‌توان به گونه‌ای بردار کرد که عبارت $X^T X$ را داشته باشد، که امکان اعمال روش‌های مبتنی بر هسته را فراهم می‌کند.
- برای مواردی که خوشه‌ها به صورت غیر خطی قابل تفکیک هستند، Kernel K-Means می‌تواند خوشه‌های دقیق‌تری را ارائه دهد.
- بسته Tslern دارای گزینه خوشه‌بندی Kernel K-Means است. خوشه‌بندی طیفی، معادل Kernel K-Means برای یک مجموعه پارامتر خاص، در sklearn موجود است.

^۹ Affinity Matrix

^{۱۰} kernel k-means

^{۱۱} Spectral clustering

۱.۴ نماد گذاری گراف

فرض کنید $G = (V, E)$ یک گراف بدون جهت با مجموعه راس $V = v_1, \dots, v_n$. در ادامه فرض می کنیم که نمودار G وزن دارد، یعنی هر یال بین دو راس v_i و v_j دارای یک غیر منفی است. وزن $w_{ij} \geq 0$. ماتریس مجاورت وزنی نمودار ماتریس $W = (w_{ij})_{i,j=1,\dots,n}$ است. اگر $w_{ij} = 0$ به این معنی است که رئوس v_i و v_j با یک یال به هم متصل نیستند. از آنجایی که G بدون جهت است، ما به $w_{ij} = w_{ji}$ نیاز داریم. درجه یک راس $v_i \in V$ به صورت تعریف می شود:

$$d_i = \sum_{j=1}^n w_{ij}$$

۱.۵ گراف شباهت

با توجه به مجموعه ای از نقاط داده x_1, \dots, x_n و تصویری از شباهت $s_{ij} \geq 0$ بین تمام جفت نقاط داده x_i و x_j ، هدف شهودی خوشه بندی تقسیم نقاط داده به چند گروه است به طوری که نقاط یک گروه مشابه و نقاط در گروه های مختلف با یکدیگر متفاوت هستند.

اگر اطلاعاتی بیش از شباهت بین نقاط داده نداشته باشیم، یک راه خوب برای نمایش داده ها به شکل گراف شباهت^{۱۲} $G = (V, E)$ است. هر رأس v_i در این نمودار نشان دهنده یک نقطه داده x_i است. اگر شباهت s_{ij} بین نقاط داده متناظر x_i و x_j مثبت یا بزرگتر از یک آستانه معین باشد، دو راس به هم متصل می شوند و یال با s_{ij} وزن می شود. اکنون می توان با استفاده از گراف شباهت، مساله خوشه بندی را مجدداً فرمول بندی کرد:

می خواهیم پارتیشنی از نمودار پیدا کنیم که یال های بین گروه های مختلف وزن بسیار کمی داشته باشند (به این معنی که نقاط در خوشه های مختلف با یکدیگر متفاوت هستند) و یال ها در یک گروه، وزن های بالایی دارند (به این معنی که نقاط درون یک خوشه مشابه یکدیگر هستند).

۱.۶ انواع گراف شباهت

چندین ساخت و ساز محبوب برای تبدیل یک داده وجود دارد فرض کنیم نقاط داده x_1, \dots, x_n با شباهت های زوجی یا فواصل جفتی s_{ij} در یک نمودار قرار می گیرند. هنگام ساخت گراف شباهت هدف مدل سازی همسایه محلی است روابط بین نقاط داده

• گراف ϵ همسایگی:

در اینجا همه نقاط را به هم وصل می کنیم که فواصل زوجی آنها کوچکتر از ϵ است. از آنجایی که فواصل بین تمام نقاط متصل تقریباً در یک مقیاس هستند (حداکثر ϵ)، وزن دادن به لبه ها اطلاعات بیشتری در مورد داده ها در نمودار گنجانده نمی شود. از این رو، گراف همسایگی معمولاً به عنوان یک نمودار بدون وزن در نظر گرفته می شود.

• گراف های k -نزدیکترین همسایه:

در اینجا هدف اتصال راس v_i به راس v_j است اگر v_j در میان k نزدیکترین همسایه های v_i باشد. با این حال، این تعریف منجر به یک نمودار جهت دار می شود، زیرا رابطه همسایگی متقارن نیست. دو راه برای غیر جهت دار کردن این نمودار وجود دارد. راه اول این است که به سادگی جهت یال ها را نادیده بگیریم، یعنی اگر v_i در میان k نزدیکترین همسایگان v_j باشد یا اگر v_j در میان k نزدیکترین همسایه های v_i باشد، v_i و v_j را با یک یال غیر جهت دار وصل می کنیم. نمودار

^{۱۲} Similarity graph

حاصل همان چیزی است که معمولاً گراف k نزدیکترین همسایه نامیده می شود. انتخاب دوم این است که رئوس v_i و v_j را به هم وصل کنیم اگر هر دو v_i در میان k -نزدیک ترین ها باشند همسایگان v_j و v_j جزو k نزدیکترین همسایگان v_i است. نمودار حاصل را گراف k نزدیکترین همسایه متقابل می نامند. در هر دو مورد، پس از اتصال رئوس مناسب، یال ها را با شباهت نقاط انتهایی آنها وزن می کنیم.

• گراف کامل:

در اینجا ما به سادگی تمام نقاط دارای شباهت مثبت را به یکدیگر متصل می کنیم و تمام یال ها را با s_{ij} وزن می کنیم. از آنجایی که نمودار باید روابط همسایگی محلی را نشان دهد، این ساخت تنها زمانی مفید است که تابع شباهت خود محله های محلی را مدل کند. یک مثال برای چنین تابع شباهتی، تابع شباهت گاوسی

$$s(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

است، که در آن پارامتر σ پهنای همسایگی ها را کنترل می کند. این پارامتر نقشی مشابه پارامتر ϵ در گراف ϵ همسایگی ایفا می کند.

تمام گراف های ذکر شده در بالا به طور منظم در خوشه بندی طیفی استفاده می شوند.

۱.۷ گراف های لاپلاسیان و ویژگی های اساسی آنها

ابزار اصلی برای خوشه بندی طیفی، ماتریس های گراف لاپلاسی هستند. مبحث کاملی وجود دارد که به مطالعه این ماتریس ها اختصاص داده شده است به نام نظریه گراف طیفی. در این قسمت می خواهیم به تعریف گراف های لاپلاسی مختلف بپردازیم و به مهم ترین ویژگی های آنها اشاره کنیم.

در ادامه همیشه فرض می کنیم که G یک گراف بدون جهت و وزن دار با ماتریس وزن W است که در آن $w_{ij} = w_{ji} \geq 0$ است. هنگام استفاده از بردارهای ویژه یک ماتریس، لزوماً فرض نمی کنیم که آنها نرمال شده باشند.

برای مثال، بردار ثابت 1 و یک مضرب $a1$ برای برخی $a \neq 0$ به عنوان بردارهای ویژه یکسان در نظر گرفته می شوند. مقادیر ویژه همیشه به طور فزاینده ای مرتب می شوند و به چندگانگی احترام می گذارند. با "اولین بردارهای ویژه k " ما به بردارهای ویژه مربوط به k کوچکترین مقادیر ویژه اشاره می کنیم. ماتریس گراف لاپلاسی نرمال نشده به صورت زیر تعریف شده است:

$$L = D - W.$$

گراف نرمال شده لاپلاسیان:

دو ماتریس وجود دارد که در ادبیات به آنها گراف نرمال شده لاپلاسیان گفته می شود. هر دو ماتریس ارتباط نزدیکی با یکدیگر دارند و به این صورت تعریف می شوند:

$$L_{sym} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

$$L_{rw} := D^{-1} L = I - D^{-1} W.$$

ماتریس اول را با L_{sym} نشان می دهیم که یک ماتریس متقارن است و ماتریس دوم را با L_{rw} نشان می دهیم زیرا ارتباط نزدیکی با طی مسیر تصادفی دارد.

یک سوال اساسی مربوط به خوشه بندی طیفی این سوال است که کدام یک از گراف های لاپلاسی باید استفاده شود. برای محاسبه بردارهای ویژه قبل از تصمیم گیری در مورد این سوال، همیشه باید به توزیع درجه نمودار شباهت نگاه کرد. اگر نمودار بسیار منظم باشد

و اکثر رئوس دارای درجه یکسانی باشند، پس همه لاپلاسی ها بسیار شبیه به یکدیگر هستند و برای خوشه بندی به همان اندازه خوب عمل خواهند کرد. با این حال، اگر درجات در نمودار بسیار گسترده توزیع شده باشند، لاپلاسی ها به طور قابل توجهی متفاوت هستند. استدلال های متعددی وجود دارد که از استفاده از خوشه بندی طیفی نرمال شده به جای نرمال نشده و در حالت نرمال شده استفاده از بردارهای ویژه L_{rw} به جای L_{sym} حمایت می کنند.

۲ روش الگوریتم

۲.۱ ترفند هسته

روش کرنل راهی برای نمایش داده های ما در فضای ابعادی بسیار بالاتر، حتی برابر با فضای بی بعدی فراهم می کند، بنابراین مدل ما می تواند در مجموعه داده های غیرخطی بهتر عمل کند. هم برای k-means مبتنی بر هسته و هم برای خوشه بندی طیفی، از دو هسته RBF تعریف شده در زیر برای محاسبه ماتریس Gram استفاده خواهیم کرد.

$$k(x, \hat{x}) = e^{-\gamma_s \|s(x) - s(\hat{x})\|^2} \times e^{-\gamma_c \|C(x) - C(\hat{x})\|^2}$$

این هسته تعریف شده جدید، اساساً دو هسته RBF را ضرب می کند تا شباهت فضایی و شباهت رنگ را همزمان در نظر بگیرد. $S(x)$ اطلاعات مکانی (یعنی مختصات پیکسل) داده x و $C(x)$ اطلاعات رنگی (یعنی مقادیر RGB داده x است. هر دو گاما فوق پارامترهایی هستند که می توانید به روش خود تنظیم کنید.

۲.۲ مشخص کردن centroid ها

ما باید مرکز را در فضای داده خود مشخص کنیم، در این مورد، از دو روش مختلف مانند Random و K-Means++ Initialization استفاده می کنیم.

- K-Means++: ایده اصلی این روش این است که ما مرکزهایی را انتخاب می کنیم که دورترین فاصله را از یکدیگر دارند. این باعث می شود که در ابتدا سنترئوئیدهایی را که در خوشه های مختلف قرار دارند، انتخاب کنید. همچنین، از آنجایی که سنترئوئیدها از نقاط داده برداشت می شوند، هر مرکز دارای برخی از نقاط داده مرتبط با آن در انتها است.
- شروع تصادفی: مقداردهی اولیه تصادفی مرکز یکی از روش های رایج در خوشه بندی K-Means است. ایده ساده است، ما فقط k مرکز را به طور تصادفی در فضای نقاط داده خودمان انتخاب می کنیم. برای این روش، از روش نمونه برداری از تابع توزیع نرمال استفاده می کنیم تا نقطه مرکز را از نقطه داده خود به دست آوریم، بنابراین باید میانگین و واریانس را به عنوان پارامتر آن تابع با استفاده از کتابخانه NumPy بدست آوریم. مرکز را با اندازه $n \times k$ ایجاد می کنیم (n بعد ویژگی های ما و k تعداد خوشه های ما است).

۲.۳ الگوریتم k-Means

الگوریتم k-Means داده ها را به k گروه تقسیم می کند، به گونه ای که داده های هر گروه به یکدیگر نزدیک تر هستند و «میانگین» یکسانی را به اشتراک می گذارند که گروه را نشان می دهد. الگوریتم k-Means در مورد یافتن انتساب نقاط داده به خوشه هایی با حداقل مجموع

مربعات فواصل تا نزدیکترین مرکز آن است. در این کد، الگوریتم خوشه‌بندی استاندارد k-Means را با نام الگوریتم لوید ساخته‌ایم. با فرآیند دو مرحله‌ای، اول مرحله توقع^{۱۳} و دوم مرحله ماکزیم سازی است.

ما با استفاده از تابعی که در قسمت قبل مشخص کردیم که k-means++ و روش تصادفی است، مرکزهای خود را مقداردهی اولیه می‌کنیم. این توابع حاوی مختصات مرکز k ما هستند. و سپس while به عنوان حلقه ای برای الگوریتم ما برای محاسبه فاصله و اختصاص دادن نقاط داده عمل می‌کند.

اولین کد داخل حلقه مرحله Expectation است، در این مرحله با استفاده از فاصله اقلیدسی، تمام نقاط داده را بر اساس نزدیکترین مرکز که قبلاً تعریف کردیم طبقه بندی می‌کنیم. و سپس به مرحله دوم در داخل حلقه که مرحله ماکزیم سازی است ادامه می‌دهیم، این مرحله برای به روز رسانی موقعیت مرکز نقاط در خوشه با گرفتن میانگین جدید آن از مرکز خود است و می‌توانیم تعداد نقاط داده‌ای که در سنتروئیدهای جدید ما به‌عنوان شاخص تعیین می‌شوند را محاسبه کنیم و مشخص کنیم که آیا نیاز به توقف یا ادامه فرآیند داریم. این فرآیند تا زمانی که موقعیت سنتروئیدهای قدیمی و جدید ما (که با تغییر داده های اختصاص داده شده در سنتروئید مشخص می‌شود) هیچ تفاوت قابل توجهی نداشته باشد تکرار می‌شود. همچنین تفاوت بین مقدار میانگین سنتروئید قدیمی و مقدار میانگین مرکز جدید را با استفاده از میانگین مربعات خطای آنها مشخص می‌کنیم.

قبل از اجرای الگوریتم خوشه‌بندی k-Means، باید تصویر خود، مجموعه k از مرکز (خوشه‌ها)، شباهت فضایی گاما^{۱۴}، شباهت رنگ گاما^{۱۵} و روش اولیه مرکز سازی ("تصادفی" یا "k-Means++") را انتخاب کنیم. خلاصه مراحل انجام k-Means مبتنی بر هسته در زیر توضیح داده شده است:

۱. ماتریس گرام آرایه داده تصویر ما را با استفاده از هسته درون `kernel_function()` محاسبه کنید.

۲. سنتروئیدها را مشخص کنید.

۳. وارد الگوریتم خوشه بندی k-Means شوید.

۲.۴ خوشه بندی طیفی

خوشه‌بندی طیفی در مورد یافتن خوشه به شکل نمودار است و می‌تواند خوشه‌هایی با شکل تقریباً دلخواه پیدا کند، به عنوان مثال در هم تنیده، مارپیچ و چون معیارهای اتصال را به جای فشردگی دارد. برای الگوریتم به این مقاله مراجعه می‌کنیم <https://arxiv.org/abs/0711.0189>.

۲.۵ خوشه بندی طیفی نرمال شده

الگوریتم:

ابتدا باید ماتریس لاپلاسی را با تفریق ماتریس درجه D و ماتریس شباهت W پیدا کنیم. ماتریس شباهت را از تابع `kernel` و ماتریس درجه را از مجموع ماتریس شباهت مورب بدست می‌آوریم. از آنجایی که این یک خوشه بندی طیفی نرمال شده است، دوم اینکه ما باید شکل نرمال شده ماتریس لاپلاسی خود را پیدا کنیم، این کار را می‌توان با استفاده از این فرمول انجام داد.

$$L_{sym} = D^{-1/2} L D^{-1/2}$$

^{۱۳}Expectation

^{۱۴}gamma spatial similarity

^{۱۵}gamma color similarity

Normalized spectral clustering according to Shi and Malik (2000)

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L .
- Compute the first k generalized eigenvectors u_1, \dots, u_k of the generalized eigenproblem $Lu = \lambda Du$.
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors u_1, \dots, u_k as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i -th row of U .
- Cluster the points $(y_i)_{i=1, \dots, n}$ in \mathbb{R}^k with the k -means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with $A_i = \{j \mid y_j \in C_i\}$.

شکل ۱: Normalized spectral clustering algorithm

بعد از اینکه numpyL_{sym} ، مقدار ویژه و فضای ویژه آن را جستجو کنیم. در مرحله بعد ماتریس T را از بردار ویژه U با نرمال کردن سطرها U به هنجار ۱ محاسبه می کنیم. در آخر، ما فقط ماتریس T را به الگوریتم k -means خود متصل می کنیم.

۲.۶ خوشه بندی طیفی نرمال نشده (Ratio Cut)

الگوریتم:

Unnormalized spectral clustering

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L .
- Compute the first k eigenvectors u_1, \dots, u_k of L .
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors u_1, \dots, u_k as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i -th row of U .
- Cluster the points $(y_i)_{i=1, \dots, n}$ in \mathbb{R}^k with the k -means algorithm into clusters C_1, \dots, C_k .

Output: Clusters A_1, \dots, A_k with $A_i = \{j \mid y_j \in C_i\}$.

شکل ۲: Unnormalized spectral clustering algorithm

برای خوشه بندی طیفی Ratio Cut، ما نیازی به محاسبه عبارت نرمال شده لاپلاسین نداریم. بنابراین، پس از دریافت ماتریس گراف لاپلاسی با استفاده از فرمول زیر:

$$L = D - W$$

مستقیماً مقدار ویژه و بردار ویژه آن را جستجو می کنیم. و ماتریس بردار ویژه U به الگوریتم k -means متصل می شود تا نقاط داده تصویر را خوشه بندی کند.

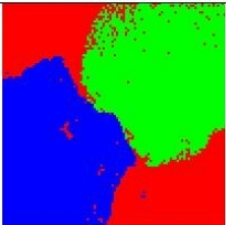
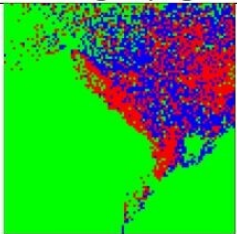
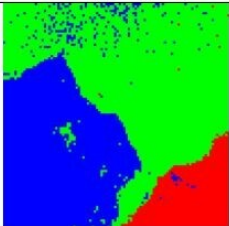
۳ دیتاست

دو تصویر 100×100 ارائه شده است که هر پیکسل در تصویر باید به عنوان نقطه داده در نظر گرفته شود، به این معنی که در هر تصویر 10000 نقطه داده وجود دارد. می‌توانیم از OpenCV برای باز کردن تصویر و استخراج داده‌های پیکسل به آرایه (1000×3) ، ارتفاع، عرض و تعداد کانال‌های رنگی (RGB) استفاده کنیم.


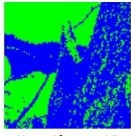
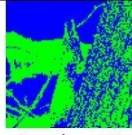
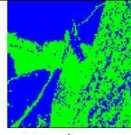
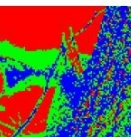
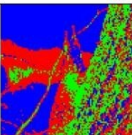
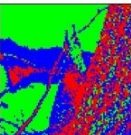
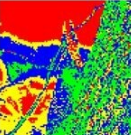
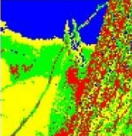
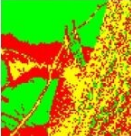
۴ نتایج و مقایسه

۴.۱ تنظیم پارامتر


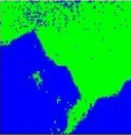
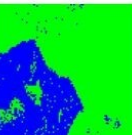
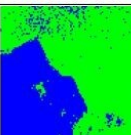
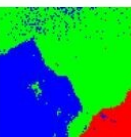
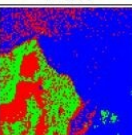

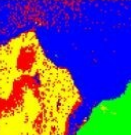
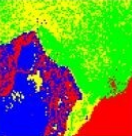
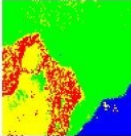
پس از مصور سازی خوشه بندی تصاویر ، در مرحله نهایی الگوریتم، خوشه بندی را با مقادیر مختلف گاما آزمایش می‌کنیم. بدیهی است که باید مقدار مناسبی را برای گاما فضایی و رنگ گاما مشخص کنیم. این را می‌توان از تصاویر خوشه ای زیر با مقادیر مختلف آن ضرایب مشاهده کرد. اگر مقدار فضایی گاما از مقدار رنگ گاما بیشتر باشد، نتیجه خوشه‌بندی بیش از مقدار پیکسل خواهد بود، به عبارت دیگر، خوشه‌بندی جزئیات زیادی ندارد. با این حال، اگر مقدار فضایی گاما بسیار کمتر از رنگ گاما باشد، نتیجه خوشه‌بندی خیلی نامشخص یا نادرست است.

Gamma spatial : 0.001 Gamma color : 0.00007	 Image1.png
Gamma spatial : 0.00007 Gamma color: 0.001	 Image1.png
Gamma spatial : 0.00007 Gamma color: 0.00007	 Image1.png

شکل ۳: Several spatial and color gamma value results

	Kernel K-Means Clustering	Normalized Spectral Clustering	Ratio Cut Spectral Clustering
k = 2	 Iteration = 13	 Iteration = 23 Processing time = 427.0s	 Iteration = 7 Processing time = 402.0s
k = 3	 Iteration = 14	 Iteration = 30 Processing time = 442.0s	 Iteration = 13 Processing time = 429.0s
k = 4	 Iteration = 14	 Iteration = 15 Processing time = 440.0s	 Iteration = 41 Processing time = 473.0s

parameters for image.2 (ب)

	Kernel K-Means Clustering	Normalized Spectral Clustering	Ratio Cut Spectral Clustering
k = 2	 Iteration = 7	 Iteration = 6 Processing time = 416.0s	 Iteration = 7 Processing time = 426.0s
k = 3	 Iteration = 9	 Iteration = 8 Processing time = 425.0s	 Iteration = 6 Processing time = 438.0s
k = 4	 Iteration = 14	 Iteration = 17 Processing time = 446.0s	 Iteration = 17 Processing time = 457.0s

parameters for image.1 (آ)


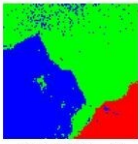
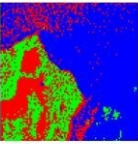
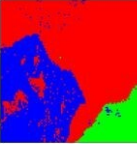

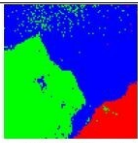
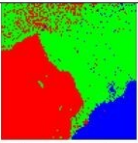
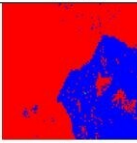

در این آزمایش از روش مقداردهی اولیه مرکزهای k-means++ استفاده کردیم و مشخص شد که پارامتر فضایی گاما و رنگ گاما برابر با 0.00007 برای همه روش‌های خوشه‌بندی مناسب است، زیرا می‌تواند بهترین تجسم را از خوشه‌بندی تصاویر به دست آورد. (شکل شماره ۳)

۴.۲ نتایج خوشه‌بندی


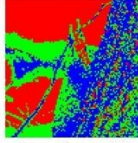
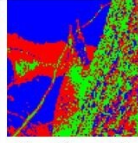
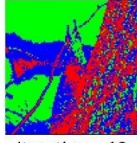

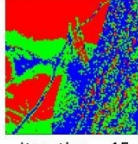
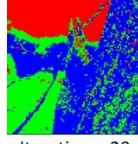
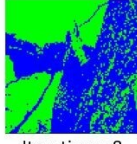

۴.۲.۱ تصویر نمونه شماره ۱

ما می‌توانیم برای $k = 2$ و $k = 3$ نشان دهیم، k-means مبتنی بر هسته و خوشه‌بندی طیفی Ratio cut می‌تواند به خوشه‌بندی به وضوح متمایز شود، در حالی که روش نرمال شده، نتایج خوشه‌بندی پراکنده‌تری دارد. برای $k = 4$ ، همه آن روش‌ها یک نتیجه خوشه‌بندی مشابه دارند. با این حال، روش k-means مبتنی بر هسته، زمان اجرای سریع‌تری برای خوشه‌بندی (30 ~ ثانیه) در یک خوشه بزرگ نسبت به هر دو روش خوشه‌بندی طیفی (7 ~ دقیقه) دارد، زیرا در خوشه‌بندی طیفی ما باید تجزیه ویژه را از ماتریس لاپلاسی نمودار پیدا کنیم که نیاز به هزینه محاسباتی بالایی دارد.

بر اساس نتایج، خوشه‌بندی طیفی نرمال شده می‌تواند به نتیجه خوشه‌بندی بهتری دست یابد، زیرا ویژگی‌های ثابت بایاس شده توسط اندازه نرمال شده قطعات و به دلیل اینکه تصویر شماره 2 دارای کمی گسترش مقادیر پیکسل رنگی است (برخلاف تصویر ۱ که دارای مقادیر پیکسل رنگی است با تفاوت بزرگ در یک زمینه با بقیه زمینه). بنابراین عبارت نرمال شده می‌تواند داده‌های تصویر را با جزئیات بیشتری از k-means غیرعادی و هسته تشخیص دهد.

		Kernel K-Means Clustering	Normalized Spectral Clustering	Ratio Cut Spectral Clustering
K-means++				
	Iteration = 7 Processing time = 31.0s	Iteration = 8 Processing time = 425.0s	Iteration = 6 Processing time = 438.0s	
Random				
	Iteration = 4 Processing time = 10.0s	Iteration = 9 Processing time = 452.0s	Iteration = 4 Processing time = 452.0s	

شکل ۵: initialization for image.1

		Kernel K-Means Clustering	Normalized Spectral Clustering	Ratio Cut Spectral Clustering
K-means++				
	Iteration = 15 Processing time = 36.0s	Iteration = 30 Processing time = 442.0s	Iteration = 13 Processing time = 429.0s	
Random				
	Iteration = 15 Processing time = 17.0s	Iteration = 20 Processing time = 461.0s	Iteration = 8 Processing time = 432.0s	

شکل ۶: initialization for image.2

یکی از معایب الگوریتم k-means این است که به تعیین اولیه مرکزها یا نقاط میانگین حساس است. بنابراین، اگر یک مرکز به عنوان یک نقطه «دور» تعیین شود، ممکن است بدون هیچ نقطه‌ای مرتبط با آن باشد و در همان زمان، بیش از یک خوشه ممکن است به یک مرکز واحد متصل شود. به طور مشابه، بیش از یک مرکز ممکن است در یک خوشه تعیین اولیه شود که منجر به خوشه بندی ضعیف می شود. و این در آزمایش ما با روش تعیین اولیه تصادفی مرکزی اتفاق افتاد. همانطور که در شکل ۵ و ۶ می‌بینید در خوشه‌بندی

طیفی برش نسبت، الگوریتم فقط می‌تواند تصویر 2 کلاس را خوشه‌بندی کند، حتی اگر k برابر با ۳ خوشه باشد. علاوه بر این، تعیین اولیه تصادفی مرکز به زمان بیشتری برای خوشه‌بندی تصویر در خوشه‌بندی طیفی نیاز دارد. به همین دلیل است که برای غلبه بر اشکال فوق از k -means++ استفاده می‌کنیم.

۴.۴ نتیجه

در این تحقیق روش خوشه‌بندی هسته مانند k -means و خوشه‌بندی طیفی را پیاده‌سازی کرده ایم که ویژگی‌های متفاوتی دارند. ترجیح می‌دهیم از k -means برای داده‌هایی استفاده کنیم که ویژگی‌های فشردگی دارند. همچنین از خوشه‌بندی طیفی برای داده‌هایی که ویژگی‌های اتصال/پراکندگی دارند (مناسب برای تخمین وضعیت بدن، استخراج پیش‌زمینه، و غیره) استفاده کنیم. خوشه‌بندی طیفی از نظر محاسباتی گران‌تر از k -means برای مجموعه داده‌های بزرگ است، زیرا باید تجزیه ویژه (فضای کم بعد) را انجام دهد. هر دو نتیجه روش خوشه‌بندی ممکن است متفاوت باشد، بستگی به نوع تعیین اولیه مرکزها دارد.

- [1] M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.
- [2] Maurizio Filippone, Francesco Camastra, Francesco Masulli, Stefano Rovetta. "A survey of kernel and spectral methods for clustering". *Pattern Recognition* 2008; 41(1):176-190.
- [3] Marcelo R.P. Ferreira, Francisco de A.T. de Carvalho, Eduardo C. Simões, Kernel-based hard clustering methods with kernelization of the metric and automatic weighting of the variables, *Pattern Recognition*, Volume 51, 2016, Pages 310-321, ISSN 0031-3203, <https://doi.org/10.1016/j.patcog.2015.09.025>.
- [4] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.
- [5] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Proceedings of the Royal Society of London*, 209:415–446, 1909.
- [6] B. Schölkopf, A. J. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.