



دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی
گرایش مهندسی نرم‌افزار

عنوان:

خوشه بندی چنددیدگاهی و بی نظارت داده‌های ژنتیکی با شبکه‌ی خودرمزگذار گراف

نگارش:

دانیال ملک‌محمد

استاد راهنما:

دکتر محمدامین فضلی

شهریور ۱۳۹۹

سلام افلا

به نام خدا
دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی

عنوان: خوشه بندی چنددیدگاهی و بی نظارت داده‌های ژنتیکی با شبکه‌ی خودرمزگذار گراف
نگارش: دانیال ملک‌محمد

کمیته‌ی ممتحنین

استاد راهنما: دکتر محمدامین فضلی
امضاء:

استاد مشاور: دکتر محمدحسین رهبان
امضاء:

استاد مدعو: استاد ممتحن
امضاء:

تاریخ:

چکیده

در بسیاری از کاربردها چندین دسته ویژگی از جنس‌های مختلف برای نمونه‌ها موجود است که اطلاعاتشان می‌تواند برای یکدیگر تکمیل‌کننده باشد. به‌ویژه در حوزه بیوانفورماتیک تکنولوژی‌های اخیر امکان جمع‌آوری داده‌های ژنومیک زیاد و گوناگونی را فراهم کرده است. این امر موجب پراهمیت شدن روش‌هایی شده که به جای آموزش متریکی روی یک مجموعه داده چندین مجموعه داده از جنس‌های مختلف را به‌کار می‌گیرند. در این مقاله سعی شده که چندین روش مبتنی بر شبکه‌های عصبی گراف کانولوشن ارائه شود که هر یک از مجموعه دادگان را به عنوان یک گراف در نظر بگیرد و با تجمیع آن‌ها به تک گرافی جامع دست یابد. در این پژوهش از سه مجموعه دادگان Methylation mRNA DNA و Micro RNA استفاده شد که مربوط به ۵ مجموعه داده‌ی سرطانی بود. عمده‌ی پژوهش روی دادگان سرطان GBM انجام شد که دارای ۲۱۵ مریض بود و هر مریض به عنوان یکی از رئوس گراف مشابَهت فرض شد. پس از بدست آوردن گراف نهایی روی آن خوشه‌بندی طیفی انجام شد تا برزسیتی مریض‌ها پیش‌بینی شود.

کلیدواژه‌ها:

Multi-view Learning, Data Fusion, Data Aggregation, Graph Convolutional Networks, Survival Analysis, Spectral Clustering, Graph Autoencoder, Autoencoder, Variational Autoencoder

فهرست مطالب

۱۰	۱ مقدمه
۱۳	۲ مفاهیم اولیه
۱۳	۱-۲ واپخشی
۱۵	۲-۲ خودرمگذار
۱۷	۳-۲ خوشه‌بندی طیفی
۲۰	۳ کارهای پیشین
۲۰	۱-۳ روش Cross Diffusion
۲۱	۲-۳ خودرمگذار متغیر گراف
۲۲	۳-۳ استفاده از خودرمگذار برای پیش‌بینی برزیستی سرطان کبد
۲۳	۴-۳ چهارچوب SIMLR
۲۴	۵-۳ مشابهت دارویی با شبکه‌ی توجه‌وار چنددیدگاهی خودرمگذار گراف
۲۵	۶-۳ ایده‌های خوشه‌بندی
۲۶	۴ روش پیشنهادی
۲۷	۱-۴ توضیح کدهای مربوط به آموزش شبکه‌های عمیق

۲۷	۴-۱-۱ نسخه‌ی ۱۰
۲۷	۴-۱-۲ نسخه‌ی ۱۱
۲۸	۴-۱-۳ نسخه‌ی ۱۲
۲۸	۴-۱-۴ نسخه‌ی ۱۳
۲۹	۴-۱-۵ نسخه‌ی ۱۴
۲۹	۴-۱-۶ نسخه‌ی ۱۵
۲۹	۴-۱-۷ نسخه‌ی ۱۶
۲۹	۴-۱-۸ نسخه‌ی ۱۷
۲۹	۴-۱-۹ نسخه‌ی ۱۸
۳۰	۴-۱-۱۰ نسخه‌ی ۲۰ تا ۲۳
۳۱	۴-۱-۱ نسخه‌ی ۲۴ تا ۲۷
۳۱	۴-۱-۲ نسخه‌ی ۲۸ تا ۳۰
۳۱	۴-۱-۳ نسخه‌ی ۳۱ تا ۳۴
۳۱	۴-۱-۴ نسخه‌ی ۳۵
۳۱	۴-۱-۵ نسخه‌ی ۳۶ تا ۳۹
۳۲	۴-۱-۶ نسخه‌ی ۴۰
۳۲	۴-۱-۷ نسخه‌ی ۴۱
۳۲	۴-۱-۸ نسخه‌ی ۴۲
۳۲	۴-۱-۹ نسخه‌ی ۴۳ و ۴۴
۳۲	۴-۱-۲۰ نسخه‌ی ۴۵ تا ۴۹
۳۳	۴-۱-۲۱ نسخه‌ی ۵۰
۳۴	۴-۱-۲۲ نسخه‌ی ۵۱

۳۴ ۴-۱-۲۳ نسخه‌ی ۵۲
۳۴ ۴-۱-۲۴ نسخه‌ی ۱ تا ۹
۳۵ ۴-۲ کد مربوط به آنالیزهای بعد آموزش شبکه

۴۰	۵ نتیجه‌گیری
----	--------------

فهرست شکل‌ها

- ۱-۲ نمای کلی یک خودرمزگذار [۱] ۱۶
- ۲-۲ نمای کلی یک خودرمزگذار متغیر [۲] ۱۷
- ۱-۴ نمای امبدینگ معماری نسخه‌ی ۱۲ در دوبعد با استفاده از TSNE ۲۸
- ۲-۴ نمودار وزن‌های ساختار توجه. قرمز مربوط به gene سبز مربوط به methy و آبی مربوط به mirna است. بعد افق شماره‌ی راس گراف‌ها را نشان می‌دهد و بعد عمود بزرگی عدد احتمالی. برای مثال نقطه‌ی آبی رنگ با مختصات (۴۳, ۰/۷) به این معنا است که $g^{mirna}(۲۵) = ۰/۷$ یعنی ضریب توجه برای راس ۴۳ ام در دید mirna ۰/۷ است و لذا در نتیجه مجموع دو دید methy و gene برای این راس ۰/۳ است. همانطور که از نمودار پیداست، اثر مذکور رخ داده و دید methy غلبه کرده‌است. . ۳۳
- ۳-۴ تصاویر حاصل از اجرای خوشه‌بندی طیفی روی دیدهای مختلف. با ترتیب سطری از راست به چپ و از بالاترین سطر به پایین‌ترین: gene ، methy ، mirna ، ترکیب سه دید با snftool و embedding حاصل از آموزش شبکه‌ی ۱۵. ۳۶
- ۴-۴ نمودار Kaplan-Meier برای خوشه‌بندی طیفی embedding حاصل از معماری ۱۵ ۳۷
- ۵-۴ نمودار violin برای خوشه‌بندی طیفی embedding حاصل از معماری ۱۵ ۳۸

فهرست جدول‌ها

۴-۱ مقدار P Value برای Cox Partial Likelihood خوشه‌بندی روی دیدهای مختلف.

۳۹

فصل ۱

مقدمه

تکنولوژی های تولید داده های ژنومیکی در حال پیشرفت سریعی هستند و این امر سبب ایجاد داده های زیاد و گوناگونی در حوزه بیولوژیکی و درمانی شده است. برای مثال مجموعه دادگان بزرگ TCGA^۱ اطلاعات ژنتیکی بیش از ۲۰ نوع سرطان و هزاران مریض را داراست. در دسترس بودن این چنین دادگان غنی سبب شده که روش های یادگیری مبتنی بر ترکیب که گوناگونی دادگان و روندهای زیستی را به کار می گیرند پراهمیت شوند. [۳] اما چالش هایی در این حوزه وجود دارند.

۱. داده های ژنومیکی معمولاً دارای ابعاد ویژگی بسیار بزرگی هستند. مثلاً مجموعه داده بیان ژنی که در این پژوهش بکار رفت دارای ۱۲۰۴۲ بعد بود که در مقایسه با ۲۱۵ نمونه بیمار عدد بزرگی است. این درحالی است که نمونه ها به صورت ذاتی در فضاها و منیفلدهای با ابعاد بسیار کوچکتری هستند. در روش ما برای حل این مسئله دادگان را با استفاده از خودرمرگذارها به ابعاد بسیار کمتری مثل ۱۰۰ بعد می بریم.

۲. متفاوت بودن مقیاس، بایاس^۲ جمع آوری و نویز و ابعاد ویژگی در هر مجموعه داده: داده های ژنتیکی مختلف ابعاد متفاوتی دارند همچنین تفاوت ماهیت بیولوژیکی و تکنولوژی جمع آوری هریک سبب می شود نوع توزیع، ویژگی های آماری و حتی اندازه ی بزرگی اعداد هر مجموعه داده با بقیه متفاوت شود. از این رو ایده های ساده ای همچون میانگین گیری همی مجموعه دادگان یا الحاق ابعاد تمام ویژگی ها نمی توانند منجر به مجموعه داده ای واحد شوند که اثر همی دیده ها در

^۱The Cancer Genome Atlas

^۲سوگیری

آن‌ها به یک‌اندازه باشد و مجموعه داده‌ای غالب نشود. برای حل این چالش به جای مطالعه‌ی خود دادگان، ماتریس شباهت^۳ نرمالایز شده‌شان (ماتریس مجاورت گراف مریض‌ها) را به کار می‌گیریم.

۳. آموزش روی هر یک از مجموعه دادگان به تنهایی منجر به دقت‌های متفاوتی می‌شود و هدف از روش‌های ترکیبی به کارگیری کل دادگان برای رسیدن به دقتی بالاتر از دقت آموزش روی تک تک دادگان است. اما این ترکیب شدن الزاما به معنای توافق همگانی^۴ نیست. توافق در مسائلی همچون رای‌گیری^۵ کاربرد دارد و با روش‌های ساده‌ای همچون میانگین، مد و... قابل انجام است. اما در این پژوهش فرض شده که هر مجموعه داده جنبه‌ای ناقص از واقعیت کلی را نشان می‌دهد. برای تمثیل فرض کنید واقعیت کلی که از آن بی خبر هستیم یک مکعب است و سه مجموعه داده داریم که هر کدام یک تصویر دوبعدی تقریباً عمود بر دوتای دیگر از این مکعب هستند. در این مثال با اجرای توافق همگانی به یک چهار ضلعی دست خواهیم یافت که غلط است. برای حل این چالش از ولگشت^۶ و واپخش^۷ در گراف بهره می‌گیریم تا گراف‌های ناقص و نویزی را ترکیب کرده و به تک گرافی جامع‌تر و دقیق‌تر برای تحلیل دست یابیم. [۴]

۴. در اغلب اوقات خوشه‌ها در فضای ویژگی به صورت محدب از یکدیگر جداپذیر نیستند و روش‌های متداول خوشه‌بندی مثل kmeans خوشه‌های محدب ارائه می‌دهند. برای رفع این چالش kmeans را به جای اینکه در دامنه‌ی ویژگی‌ها اجرا کنیم در دامنه‌ی طیفی گراف شباهت مبتنی بر ویژگی‌ها انجام می‌دهیم. این روش مرسوم به خوشه‌بندی طیفی^۸ است.

۵. روش‌های متداول ترکیب مجموعه دادگان عمدتاً مبتنی بر یادگیری ماشین کلاسیک هستند و از ایده‌ی یادگیری عمیق استفاده نمی‌کنند. این درحالی است که روش‌های مبتنی بر یادگیری عمیق در چند سال اخیر نتایج بسیار بهتری در اکثر حوزه‌ها کسب کرده‌اند. از این رو روش پیشنهادی ما مبتنی بر شبکه‌های یادگیری عمیق با لایه‌های GCN^۹ بود. که اخیراً بسیار مورد توجه قرار گرفته‌اند. [۵] [۶]

^۳ Similarity Matrix
^۴ Consensus
^۵ Voting
^۶ Random Walk
^۷ diffusion
^۸ Spectral Clustering
^۹ Graph Convolutional Network

این نوشتار گزارش تلاش‌های ما برای آموزش بدون سرپرستی^{۱۰} شبکه‌های عمیق روی مجموعه دادگان به‌کار گرفته‌شده در [۷] است. عمده‌ی بررسی‌ها روی داده‌ی سرطانی GBM بود که مربوط به ۲۱۵ بیمار مبتلا است. به ازای هر مریض اطلاعات مربوط به بیان ژنی (۱۲۰۴۲ بعد)، متیلشن دی.ان.ای (۱۳۰۵ بعد) و میکرو آر.ان.ای (۵۳۴ بعد) موجود است. برچسب مریض‌ها هم تعداد روزهای برزیستی^{۱۱} آن‌ها است. ابتدا از هرکدام از سه مجموعه داده یک ماتریس مشابهت به‌دست می‌آید. سپس تعدادی معماری شبکه‌ی خودرمزگذار با لایه‌های GCN آزموده می‌شود که از ماتریس لاپلاسیون این سه گراف به عنوان کرنل^{۱۲} و از خود دادگان به عنوان سیگنال در آن‌ها استفاده می‌شود. در نهایت ماتریس لاپلاسیون ترکیب‌شده و embedding ها مورد مطالعه قرار می‌گیرند، با روش‌هایی مثلی مثل خوشه‌بندی طیفی خوشه‌بندی می‌شوند و صحتشان با استفاده از مدل Likelihood Cox Partial تست فرضیه می‌شود تا P Value بدست آمده با [۷] مقایسه شود. هدف این است که بیماران هر خوشه توزیع برزیستی یکسانی با یکدیگر و متفاوت با خوشه‌های دیگر داشته باشند.

فصل ۲

مفاهیم اولیه

۱-۲ واپختی

در این نوشتار و مقاله‌های مشابه از هریک از مجموعه‌دادگان به‌عنوان یک دید^۱ یاد می‌شود. در پژوهش ما^۳ دید X_1, X_2, X_3 وجود داشت که هریک $N \times d_v$ بعدی بود و سطر j ام در همه‌ی آن‌ها مربوط به نمونه بیمار j ام است. از روی هر دید می‌توان یک ماتریس مشابهت ساخت. یک روش معمول و پایه‌ای استفاده از کرنل گاوسی است.

$$A_v(i, j) = \exp\left(-\frac{\rho(x_i, x_j)^2}{\sigma^2}\right) \quad (1-2)$$

که $\rho(x_i, x_j)$ فاصله‌ی اقلیدسی بیمار i و j است.

ماتریس $A_v \in \mathbb{R}^{N \times N}$ را می‌توان ماتریس مجاورت گرافی فرض کرد که راس‌هایش مریض‌ها هستند و وزن یال‌هایش میزان شباهت بین مریض‌ها مطابق دید v ام است. برای هر گراف ماتریس لاپلاسیان به صورت زیر تعریف می‌شود.

$$L = D - A \quad (2-2)$$

ماتریس قطری $D \in \mathbb{R}^{N \times N}$ ماتریس درجات نام‌دارد؛ درایه‌ی (i, i) آن درجه راس i ام است. همچنین

View'

ماتریس لاپلاسین متقارن و نرمالایز شده به صورت زیر تعریف می شود.

$$\tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \quad (۳-۲)$$

حال می خواهیم فرآیند واپخشی در گراف را به صورت ریاضی تعریف کنیم. فرض کنید به هر راس گراف یک مقدار اسکالر نسبت داده ایم و قصد داریم این مقادیر را از طریق یال ها پخش کنیم. بردار $x \in \mathbb{R}^N$ را بیانگر این مقادیر تعریف کنید. با ضرب ماتریس لاپلاسین در آن مقادیر رئوس از طریق یال ها و باتوجه به وزنشان در گراف پخش می شوند. بردار

$$x' = Lx \quad (۴-۲)$$

مقادیر رئوس بعد از پخش شدن است. برای درک بهتر این فرآیند ضرب داخلی رئوس قبل و بعد از پخش شدن را در نظر. به راحتی می توان ثابت کرد

$$\langle x, x' \rangle = x^\top Lx = \sum_{(u,v) \in E} A(u,v) (x(u) - x(v))^2 \quad (۵-۲)$$

طبق این رابطه فرآیند واپخشی به فراخور وزن یال ها اتفاق می افتد، یعنی همسایگان راس v مقادیر خود را از طریق یال واصلشان به این راس می دهند و هرچقدر وزن یال واصل و یا مقدار اسکالر همسایه بیشتر باشد، عدد بیشتری از آن همسایه به v می رسد. اگر به جای L از \tilde{L} استفاده کنیم مقداری که از u به v می رسد به جای اینکه به $A(u,v)$ بستگی داشته باشد، متاثر از $\frac{A(u,v)}{\deg(u)}$ خواهد بود (اصطلاحاً برحسب مجموع یال های خروجی از هر راس نرمالایز می شود). همچنین اگر از L^k به جای L استفاده شود در فرآیند واپخشی به جای اینکه فقط مقادیر رئوس همسایه ی یک راس به آن سرازیر شوند، مقادیر رئوس با فاصله ی حداکثر k به آن سرازیر می شوند.

به سادگی می توان مفهوم واپخشی در گراف را از انتساب فقط یک اسکالر به هر راس، به انتساب یک بردار d بعدی به هر راس تعمیم داد (هر بعد این بردار مستقل از ابعاد دیگر مطابق آنچه گفته شد پخش شود).

$$X' = LX \quad \& \quad X \in \mathbb{R}^{N \times d} \quad (۶-۲)$$

در فرآیند واپخشی دو نوع دامنه وجود دارد: دامنه ی فضای برداری d بعدی ویژگی ها (دامنه ی سیگنال)

و دامنه‌ی فضای برداری N بعدی رئوس گراف (دامنه‌ی طیفی ^۲). فرآیند واپخشی همانگونه که دیدیم در دامنه‌ی اسپکترم به صورت یک عمل خطی تعریف شد اما تفسیر آن در دامنه‌ی سیگنال به این سادگی نیست.

برای درک بهتر فرآیند واپخشی جالب است بدانید که تمامی تعاریف این حوزه مشابه فوریه در مبحث سیگنال است؛ دامنه‌ی سیگنال و دامنه‌ی طیفی در واپخشی به ترتیب دامنه‌ی زمانی و دامنه‌ی فرکانسی در مبحث سیگنال است. انجام فرآیند واپخشی با ماتریس لاپلاسن هم همان به کارگیری فیلتر فرکانسی است.

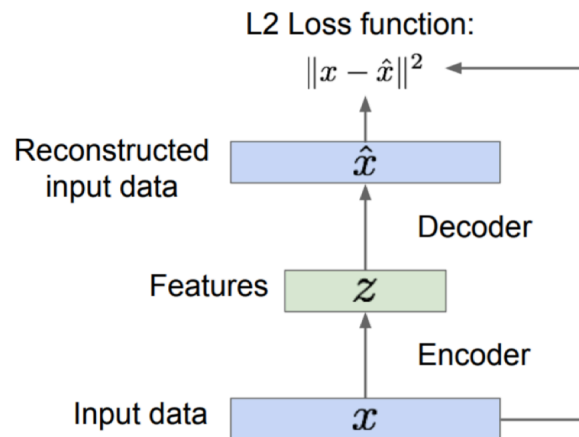
۲-۲ خودرمزگذار

خودرمزگذارها ^۳ گونه‌ای از شبکه‌های عمیق هستند که به صورت بدون سرپرستی یک embedding (معمولاً در ابعادی کمتر از ابعاد داده‌ی اصلی) را یاد می‌گیرند. این شبکه‌ها دارای دو قسمت هستند: بخش رمزگذار ^۴ و رمزگشا ^۵. داده ابتدا به رمزگذار داده می‌شود تا طی چندین لایه ابعادهای کاهش یابد، سپس رمزگشا این نمایش در ابعاد پایین را دوباره به ابعاد اولیه می‌برد و سعی می‌کند داده‌ی اصلی را بازسازی کند. شکل ۱-۲ نمای کلی این شبکه‌ها را نشان می‌دهد.

یک تابع ضرر متداول برای این شبکه‌ها $(X - X')^2$ می‌تواند باشد که X ورودی و X' خروجی حاصل از بازسازی رمزگشا است.

تجربه نشان داده‌است که خودرمزگذار ساده الزاماً نمونه‌ها را به صورت خوش تعریف در فضای embedding رمزگذاری نمی‌کند، بلکه ممکن است هر داده را به نقطه‌ای تصادفی در فضای embed-ding ببرد به صورتی که فاصله‌ی نقاط و محلیت در این فضا معنای خاصی نداشته باشد، به عبارتی دیگر خودرمزگذار صرفاً به عنوان یک تابع هش عمل کند. این درحالی است که ما در بسیاری از کاربردها علاقه‌مندیم که فضای embedding دارای خواصی همچون پیوستگی و همواری ^۶ باشد، فاصله‌ها بامعنی باشند، داده‌های مشابه در فضای embedding نزدیک هم باشند و حرکت در یک جهت خاص از embedding موجب نوعی تغییر پیوسته در ویژگی‌ای از نمونه‌ها شود. برای حل این مشکل از شبکه‌ی

Spectral^۲
Autoencoder^۳
Encoder^۴
Decoder^۵
Smoothness^۶



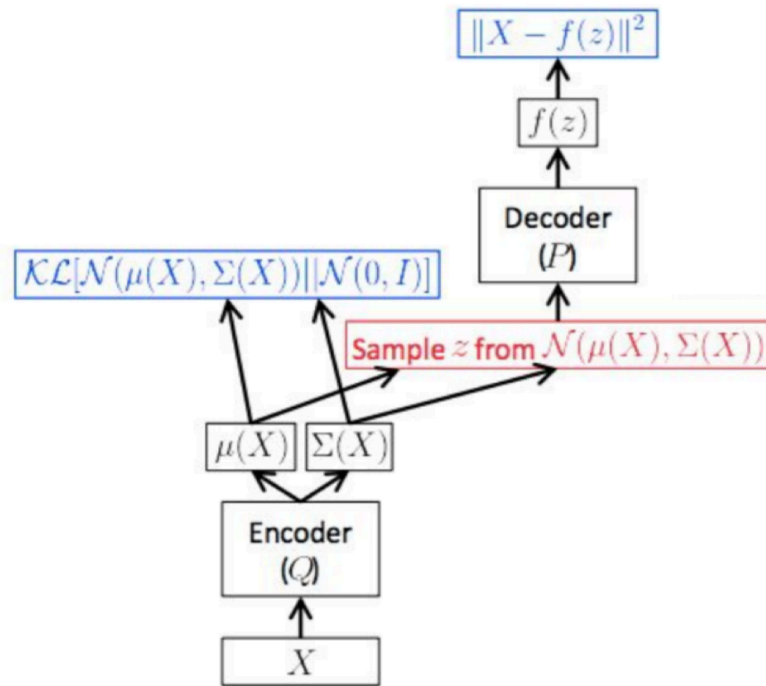
شکل ۲-۱: نمای کلی یک خودرمزگذار [۱]

خودرمزگذار متغیر^۷ استفاده می‌شود. این شبکه نیز دارای دو قسمت رمزگذار و رمزگشا است با این تفاوت که نقاط در فضای embedding به صورت احتمالاتی تصور می‌شوند؛ رمزگذار داده را به جای اینکه به یک نقطه در فضای embedding ببرد، به یک توزیع نرمال چندبعدی^۸ در آن فضا متناظر می‌کند. از این رو قسمت رمزگذار دارای دو شبکه است: یکی برای ارائه μ یا میانگین توزیع و دیگری برای ارائه ماتریس کواریانس Σ توزیع. سپس رمزگشا مطابق توزیع یک نمونه تصادفی ایجاد می‌کند و از روی آن داده‌ی اصلی را بازسازی می‌کند. شکل ۲-۲ نمای کلی این شبکه‌ها را نشان می‌دهد. تابع هدف در این شبکه‌ها به صورت

$$E_z[\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)) \quad (۷-۲)$$

که عبارت اول برای بازسازی است و عبارت دوم سعی می‌کند توزیعی که هر نمونه در فضای embed-ding پیدا می‌کند شبیه توزیع پیشین (معمولاً $\mathcal{N}(0, \sigma^2 I)$) باشد.

^۷Variational Autoencoder
^۸Multivariate Normal



شکل ۲-۲: نمای کلی یک خودرمزگذار متغیر [۲]

۳-۲ خوشه‌بندی طیفی

تعریف ۲-۱ (برش) در گراف مفهوم برش^۹ برای دو مجموعه رئوس A و B به صورت

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij} \quad (۸-۲)$$

تعریف می‌شود.

تعریف ۲-۲ (RatioCut) برای k مجموعه‌های $A_1 \dots A_k$ به صورت

$$RatioCut(A_1, \dots, A_k) = \sum_{i=1}^k \frac{cut(A_i, \overline{A_i})}{|A_i|} \quad (۹-۲)$$

تعریف می‌شود که $|A_i|$ تعداد رئوس مجموعه‌ی A_i است.

بر اساس RatioCut می‌توان یک نوع خوشه‌بندی برای گراف تعریف کرد:

یافتن افزای k مجموعه‌ای (k خوشه) به صورت $A_1 \dots A_k$ که مقدار $RatioCut(A_1, \dots, A_k)$ شان

کمینه باشد.

Cut^۹

خوشه‌بندی طیفی در حقیقت نسخه‌ی راحت‌شده^{۱۰} شده‌ی این مسئله است که در ادامه توضیح داده می‌شود.

ماتریس لاپلاسین طبق ۲-۵ مثبت نیمه معین بوده و متقارن بودنش نیز پرواضح است از این رو همواره L را می‌توان به صورت

$$L = QSQ^T \quad (۲-۱۰)$$

تجزیه کرد که Q ماتریس متعامد و S ماتریس قطری است. در واقع هرستون Q یکی از بردار ویژه‌های L و درایه‌ی متناظرش در S مقدار ویژه‌اش است. یک الگوریتم برای خوشه‌بندی طیفی بدین صورت است که k تا بردار ویژه‌ی مربوط به k تا کوچک‌ترین مقدار ویژه‌های ماتریس لاپلاسین را نگه می‌داریم و بقیه را دور می‌ریزیم.

$$L' = Q_{[\cdot:k]} S_{[\cdot:k]} Q_{[\cdot:k]}^T \quad (۲-۱۱)$$

سپس در فضای k بعدی جدید متشکل از این k تا بردار ویژه $(Q_{[\cdot:k]})$ الگوریتم kmeans را اجرا می‌کنیم تا رئوس گراف خوشه‌بندی شوند.

یک راه دیگر برای اجرای خوشه‌بندی طیفی حل

$$\min_Y \{\text{Trace}(Y^T LY)\} \quad (۲-۱۲)$$

با شرط

$$Y \in \mathbb{R}^{N \times k} \ \& \ Y^T Y = I_k \quad (۲-۱۳)$$

است (هر ستون ماتریس Y معرف یکی از خوشه‌ها می‌شود).

به عنوان نکته‌ی آخر اگر در دو الگوریتم مذکور برای خوشه‌بندی طیفی از ماتریس لاپلاسین نرمالایز شده به جای ماتریس لاپلاسین استفاده کنیم Ncut کمینه می‌شود.

تعریف ۲-۳ (Ncut)

$$Ncut(A_1, \dots, A_k) = \sum_{i=1}^k \frac{cut(A_i, \overline{A_i})}{vol(A_i)} \quad (۲-۱۴)$$

تعریف می‌شود که

$$vol(A) = \sum_{i \in A} d_i \quad (2-15)$$

فصل ۳

کارهای پیشین

۱-۳ Cross Diffusion روش

پژوهش [۸] در سال ۲۰۱۲ این روش را برای ترکیب چندین مجموعه داده ارائه داد. در آن ابتدا براساس هر مجموعه داده یک ماتریس مشابهت می سازیم. (یک روش معمول رابطه‌ی ۱-۳ است).

$$\mathcal{P}(i, j) = \frac{W(i, j)}{\sum_{k \in V} W(i, k)} \quad (1-3)$$

سپس این ماتریس مشابهت را به ماتریس مارکوف تبدیل می کنیم. در ادامه برای $t = 1..T$ بار ادغام را به صورت

$$P_{t+1}^{(i)} = \mathcal{P}^{(i)} \times \left(\frac{1}{m-1} \sum_{j \neq i} P_t^{(j)} \right) \times \mathcal{P}^{(i)\top} \quad (2-3)$$

انجام می دهیم. توجه کنید که $\mathcal{P}^{(i)}$ ماتریس مارکوف حاصل از دید i ام است و

$$P_{t=1}^{(i)} = \mathcal{P}^{(i)} \quad (3-3)$$

در نهایت ماتریس نهایی به صورت

$$P^{(c)} = \frac{1}{m} \sum_{i=1}^m P_T^{(i)} \quad (4-3)$$

بدست می آید. برای درک شهودی این روش حالتی که دو مجموعه داده داشته باشیم را در نظر بگیرید.

$$P_{t+1}^{(1)} = \mathcal{P}^{(1)} \times P_t^{(2)} \times \mathcal{P}^{(1)\top} \quad (5-3)$$

$$P_{t+1}^{(2)} = P^{(2)} \times P_t^{(1)} \times P^{(2)\top} \quad (۳-۶)$$

زمانی که $t = 1$ باشد، سمت راست رابطه‌ی ۳-۶ احتمال مسیرهای به طول ۳ ای را در نظر می‌گیرد که یال اول و سوم آن‌ها مطابق گراف دید ۱ و یال دوم مطابق گراف دید ۲ است. حال اگر T را برای مقادیر بزرگتری در نظر بگیریم مسیرهایی در نظر گرفته می‌شوند که به طور یک درمیان طبق گراف دید ۱ و دید ۲ می‌شوند و بدین ترتیب گراف دو مجموعه داده ادغام می‌شوند. در [۹]، [۱۰] و [۷] از همین این روش استفاده شد.

۳-۲ خودرمزگذار متغیر گراف

این نوع از شبکه‌ی خودرمزگذار توسط کیپف و ماکس ولینگ در سال ۲۰۱۶ معرفی شد [۱۱] و به سرعت مورد توجه قرار گرفت [۱۲]، [۱۳] و [۱۴]. در این پژوهش از یک گراف بدون جهت استفاده شد که هر راس آن یک مقاله بود و وجود یال بین دو راس به معنای ارجاع دهی یکی توسط دیگری بود. علاوه بر آن تمام مقالات دارای یک بردار صفرویکی بودند که هر درایه وجود یا عدم وجود کلمه‌ای از دیکشنری در آن مقاله را نشان می‌داد. در این پژوهش حدود ۱۰٪ از یال‌های گراف ارجاع دهی به عنوان داده‌ی آزمایش حذف شدند و با استفاده از بردار کلمات مذکور این یال‌های با دقت ۹۰٪ بازسازی شدند. این خودرمزگذار در هر اپوک ۱ یادگیری تمام بردارهای مربوط به وجود یا عدم وجود کلمات دیکشنری در هر مقاله را می‌گیرد (این ماتریس $X \in \mathbb{R}^{N \times d}$ توصیف مقالات در دامنه‌ی سیگنال است).

تفاوت اصلی این شبکه با شبکه‌های عادی خودرمزگذار استفاده از لایه‌های GCN بود. یک لایه MLP ساده به صورت

$$MLP(X) = ReLU(XW) \quad (۳-۷)$$

اما در لایه‌ی GCN عمل واپخشی نیز انجام می‌شود.

$$GCN(X, A) = ReLU(\tilde{A}XW) \quad (۳-۸)$$

$$\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = I - \tilde{L} \quad (۳-۹)$$

Epoch^۱

بخش رمزگذار دارای دو لایه GCN است و بخش رمزگشا هدفش بازسازی ماتریس مجاورت A است، به این صورت که اگر z_i و z_j نمایش دو راس i و j در embedding باشند،

$$p(A_{ij} = 1 | z_i, z_j) = \sigma(z_i^T z_j) \quad (۱۰-۳)$$

است. از این رو رمزگشا $\hat{A} = \sigma(ZZ^T)$ را به عنوان خروجی ارائه می‌کند و تلاش دارد \hat{A} شبیه A شود.

۳-۳ استفاده از خودرمزگذار برای پیش‌بینی برزیستی سرطان کبد

پژوهش [۱۵] در سال ۲۰۱۷ از خودرمزگذارهای ساده برای ترکیب ۳ مجموعه داده‌ی RNA-seq، methylation و miRNA-seq برگرفته از TCGA استفاده کرد تا برزیستی ۳۶۰ نمونه بیمار مبتلا را پیش‌بینی کند. در این پژوهش هرسه مجموعه داده الحاق شدند و خود رمزگذار با تابع هزینه‌ی

$$\text{logloss}(x, x') = \sum_{k=1}^d (x_k \log(x'_k) + (1 - x_k) \log(1 - x'_k)) \quad (۱۱-۳)$$

، رگولاریزیشن L_1 روی وزن‌های شبکه، رگولاریزیشن L_2 روی خروجی تابع‌های فعال‌سازی شبکه و لایه‌های dropout با نرخ ۵۰٪ آموزش داده‌شد. معماری خودرمزگذار به‌صورت سه لایه‌های با ابعاد به ترتیب ۵۰۰، ۱۰۰، ۵۰۰ بود. بعد از آموزش شبکه از embedding به عنوان ویژگی‌های جدید استفاده گردید و با مدل تک بعدی Cox Partial Likelihood ابعادی از آن که p value قابل توجه ($\log\text{-rank } p\text{-value} < 0.05$) داشتند انتخاب شدند. سپس در محیط ویژگی جدید خوشه‌بندی kmeans انجام شد تا برچسب‌های خوشه‌بندی به‌دست آیند. در نهایت ابعادی از ویژگی اولیه که بیشترین کورولیشن با برچسب‌های خوشه‌بندی را داشتند انتخاب شدند تا الگوریتم SVM روی آن‌ها آموزش داده‌شود (۱۰۰ بعد از mRNA، ۵۰ بعد از methylation و ۵۰ بعد از miRNA).

۴-۳ چهارچوب SIMLR

در سال ۲۰۱۷ [۱۶] چهارچوب مبتنی بر بهینه‌سازی SIMLR^2 را برای ترکیب چندین کرنل ارائه داد. در این پژوهش با استفاده از روش بهینه‌سازی تابع هزینه‌ی

$$-\sum_{i,j,l} w_l K_l(c_i, c_j) S_{ij} + \beta \|S\|_F^2 + \gamma \text{tr}(L^\top (I_N - S)L) + \rho \sum_l w_l \log w_l \quad (12-3)$$

با شروط

$$L^\top L = I_C, \sum_l w_l = 1, w_l \geq 0, \sum_j S_{ij} = 1 \ \& \ S_{ij} \geq 0 \quad (13-3)$$

کمینه می‌شود که K_l ها کرنل‌هایی هستند که می‌خواهیم با یکدیگر ترکیبشان کنیم، w_l ضریب اهمیتی است که به هر کرنل انتساب داده می‌شود و S ماتریس مشابهتی است که قصد داریم به دستش بیاوریم. عبارت اول در تابع هزینه موجب می‌شود که اگر فاصله‌ی بین دو سلول i و j در کرنل‌ها زیاد باشد میزان مشابهت آن دو در ماتریس S یعنی S_{ij} کم شود. عبارت دوم یک رگولاریزیشن برای جلوگیری از تبدیل S به ماتریس همانی است. عبارت سوم موجب می‌شود که گراف با ماتریس مجاورت S تقریباً به صورت c خوشه باشد که یال‌های درون خوشه‌ای سنگین و یال‌های بین خوشه‌ای ضعیف باشند (هر ستون ماتریس L بردار معرف یکی از این خوشه‌ها است). عبارت آخر نیز از اینکه یک کرنل خیلی مهم و بقیه بی اهمیت شوند جلوگیری می‌کند و به مساوات اثر کرنل‌ها در تشکیل S کمک می‌کند.

تعریف ریاضی این مسئله محدب نیست اما بهینه‌کردن هریک از سه متغیر به شرط ثابت بودن دوتای دیگر یک مسئله‌ی محدب است و به همین دلیل [۱۶] به صورت متناوب آن را بهینه می‌کند. 4 بعد از به‌دست آمدن S ماتریس انتقال P به صورت

$$P_{ij} = \frac{S_{ij} \setminus \{j \in A_K(i)\}}{\sum_l S_{il} \setminus \{l \in A_K(i)\}} \quad (14-3)$$

تعریف می‌شود تا به صورت

$$H_{ij}^{(t+1)} = \tau H_{ij}^{(t)} P + (1 - \tau) I_N \quad (15-3)$$

برای چند مرحله روی آن فرآیند واپختشی رخ دهد. این امر سبب کاهش نویزهای موجود در S می‌شود. در نهایت از $H_{ij}^{(T)}$ به عنوان ماتریس مشابهت برای تحلیل‌هایی همچون خوشه‌بندی استفاده می‌شود.

Optimization^۲
Single Cell Interpretation via Multi-kernel Learning^۳
Alternating^۴

۳-۵. مشابهت دارویی با شبکه‌ی توجه‌وار چنددیدگاهی خودرمزگذار گراف

الهام‌بخش‌ترین مقاله برای پژوهش ما [۱۳] بود که در معماری‌های نسخه‌ی ۲۰ به بعد از آن استفاده کردیم (البته روش‌های پیشنهادی این مقاله به صورت یادگیری بانظارت^۵ یا یادگیری نیمه‌نظارت^۶ و ترارسانی^۷ است ولی تمامی معماری‌های ما به صورت یادگیری بی‌نظارت^۸ انجام شد). در این پژوهش برای تعدادی دارو چندین دید (مثلاً CPI^۹ یا TTD^{۱۰}) وجود دارد که با ترکیب آن‌ها سعی می‌شود پیش‌بینی بهتری از DDI^{۱۱} داروها به دست آید. نوآوری اصلی این پژوهش استفاده از ساختار توجه برای کرنل لاپلاسیون لایه‌های GCN در خودرمزگذار گراف کیپف [۱۱] است. سابقاً در پژوهش کیپف واپخشی با لاپلاسیون فقط یک گراف (گراف ارجاع‌دهی مقالات) انجام می‌شد اما در این پژوهش از

$$\hat{A} = \sum_u \text{diag}(g^u) A^u \quad (۱۶-۳)$$

به عنوان کرنل واپخشی استفاده شد که A^u ماتریس مشابهت دید u ام است. $g \in \mathbb{R}^N$ نیز بردار خروجی ساختار توجه است (برخلاف [۱۶] که به هر دید یک ضریب توجه می‌دهد، این پژوهش به هر راس گراف در هر دید یک ضریب توجه می‌دهد) که یک شبکه‌ی تک لایه‌ی MLP با تابع فعال‌سازی softmax و A^u ها به عنوان ورودی است.

$$\sum_u \text{diag}(g^u) = I \quad (۱۷-۳)$$

بخش رمزگذار یک شبکه‌ی دولایه به صورت

$$Z^{(u)} = f(X^{(u)}, A^{(u)}; W_1^u, W_2^u) = \text{Softmax}(\hat{A}^u \text{ReLU}(\hat{A}^u X^{(u)} W_1^{(u)}) W_2^u) \quad (۱۸-۳)$$

است که $Z^{(u)}$ ها با یکدیگر الحاق می‌شوند تا امبدینگ Z تشکیل شود. بخش رمزگشا بسته به نوع یادگیری (نیمه‌نظارتی، بانظارت و یا ترارسانی) متفاوت است. مثلاً می‌تواند به صورت

$$X' = f'(X, \hat{A}) = \text{Sigmoid}(\hat{A} Z W_3) \quad (۱۹-۳)$$

Supervised Learning^۵

Semi-Supervised Learning^۶

Transductive^۷

Unsupervised Learning^۸

Drug Chemical Protein Interactome^۹

Protein and Nucleic Acid Targets^{۱۰}

Drug Drug Interaction^{۱۱}

با تابع هزینه‌ی

$$\sum ||X - X'||^2 \quad (۲۰-۳)$$

باشد و همچنین تابع ضرر بخش یادگیری بانظارت نیز به آن افزوده شود (شبکه‌ی مربوط به پیش‌بینی هم مثل بخش رمزگشا ورودیش Z است). در این پژوهش زمانی که سیگنال ورودی X وجود نداشت از بردار one-hot رئوس استفاده شد همچنین در مدلی دیگری از آن برچسب‌ها به عنوان سیگنال ورودی به‌کار برده شدند.

۳-۶ ایده‌های خوشه‌بندی

در [۱۷] ایده‌ی خوشه‌بندی طیفی با توجه به یک ماتریس Q حاوی اطلاعات اضافه ارائه شده‌است. درایه‌ی (i, j) از این ماتریس $n \times n$ بعدی در صورتی که دو نمونه‌ی i ام و j ام هم خوشه باشند $+1$ و در غیر این صورت -1 است. در این مقاله، که مبتنی بر یادگیری ماشین کلاسیک است، ماتریس Q در تابع هزینه‌ی ۲-۱۲ بکارگرفته می‌شود تا در خوشه‌بندی حاصل Q حتی‌المکان برقرار باشد. در [۱۸] همین ایده با استفاده از شبکه‌های عصبی با لایه‌های MLP پیاده‌سازی می‌شود. اما در [۱۹] شبکه‌ای با لایه LSTM ارائه شده که مبتنی بر ایده‌های قبلی نیست و به ترتیب زمانی تک تک بیماران را به عنوان ورودی می‌گیرد تا خوشه‌بندی را انجام دهد. در [۲۰] ایده‌ی الگوریتمی برای خوشه‌بندی چنددیدگاهی داده‌ها با مقایس بالا ارائه شده‌است و در آن ماتریس مشابهت توسط یک گراف دوبخشی تقریب زده می‌شود تا ۲-۱۲ سریعتر محاسبه شود. در [۲۱] روشی برای اجرای خوشه‌بندی kmeans به صورت چنددیدگاهی ارائه شده‌است که برخلاف [۱۶] که به هردیدگاه یک ضریب اهمیت نسبت دهد، به هر نمونه یک ضریب می‌دهد.

فصل ۴

روش پیشنهادی

در این پژوهش از داده‌گان [۷] استفاده شد. قسمت‌هایی از کد این مقاله (پکیج SNFtool) که به زبان R نوشته شده برای تحلیل ضروری بودند و از سوی دیگر آموزش شبکه‌های عصبی در محیط pytorch و به زبان پایتون انجام می‌شد. برای حل این مشکل از پکیج rpy۲ استفاده کردیم تا بتوانیم کدهای به زبان R را در محیط پایتون اجرا کنیم.

ابتدا هر دید را به صورت

$$\tilde{f} = \frac{f - E(f)}{\sqrt{Var(f)}} \quad (۱-۴)$$

نرمالایز کردیم. سپس از روزی هریک ماتریس مشابهتی به صورت

$$A(i, j) = \exp\left(-\frac{\rho^2(x_i, x_j)}{\mu \epsilon_{i,j}}\right) \quad (۲-۴)$$

ساختیم که ρ فاصله‌ی اقلیدسی بود^۱، $\epsilon_{i,j}$ از طریق رابطه‌ی

$$\epsilon_{i,j} = \frac{\text{mean}(\rho(x_i, N_i)) + \text{mean}(\rho(x_j, N_j)) + \rho(x_i, x_j)}{۳} \quad (۳-۴)$$

بدست آمد و $\rho(x_i, N_i)$ میانگین فاصله‌ی x_i از همسایگانش بود. البته قبل از این نرمالیز کردن ماتریس مشابهت را تنک کردیم: برای هر راس، در سطر مربوطش فقط K همسایه‌ی نزدیکش را در نظر گرفتیم و بقیه‌ی درایه‌ها را صفر کردیم. (μ و K ابرپارامتر هستند و به صورت تجربی تنظیم می‌شدند اما به‌طور معمول حدود آن‌ها به صورت $۱۰ \leq K \leq ۳۰$ و $۰/۸ \leq \mu \leq ۰/۳$ و $۱۰ \leq T \leq ۲۰$ بود).

^۱ البته کورولیشن را هم امتحان کردیم اما نتیجه‌ی مطلوبی نداد

در ادامه نسخه های مختلف کدهایی که ایجاد کردیم را به ترتیب زمانی توضیح می دهیم. فایل iphyton مربوط به هر نسخه به همراه دیگر کدها و نتایج در **این نشانی** موجود است. عمدتاً تفاوت هر نسخه با بقیه در ایده های اصلی است و تغییرات جزئی از قبیل تعداد ابعاد هر لایه از شبکه یا مقادیر ابرپارامترها در همان نسخه بررسی شده است.

۴-۱ توضیح کدهای مربوط به آموزش شبکه های عمیق

۴-۱-۱ نسخه ۱۰

برای شروع ساده ترین ایده یعنی خودرمزگذار ساده آزمایش شد. هر سه مجموعه داده الحقاق^۲ شدند و به عنوان ورودی به شبکه استفاده شدند. بعد از آموزش نمایش داده ها در embedding بررسی شد. ^۳ برای خوشه بندی embedding ها هم از Kmeans و هم از Spectral Clustering استفاده شد و روی خوشه ها میزان P Value مربوط به Cox Partial Likelihood محاسبه شد و با نتایج [۷] مقایسه می شد. در طی این نسخه متوجه شدیم embedding بسیار متأثر از مجموعه داده ی ژنتیکی است و دلیلش هم منطقی بود، چراکه با داشتن ۱۲۰۴۲ بعد بود درمقایسه با ۱۳۰۵ و ۵۳۴ بعد تاثیر بسیار زیادی روی تابع هزینه می گذاشت. (تابع هزینه $(X - X')^2$ بود.) برای رفع این مشکل معماری بعدی ارائه شد.

۴-۱-۲ نسخه ۱۱

کماکان از خودرمزگذار ساده استفاده شد اما تابع هزینه به صورت

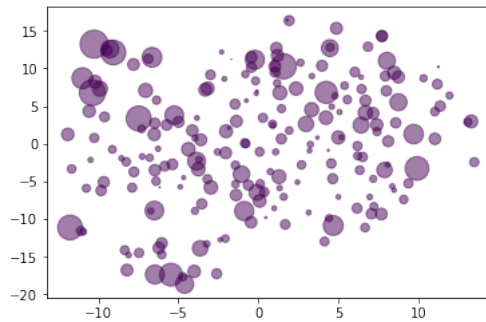
$$\frac{(X_{[gene]} - X'_{[gene]})^2}{\dim(gene)} + \frac{(X_{[methy]} - X'_{[methy]})^2}{\dim(methy)} + \frac{(X_{[mirna]} - X'_{[mirna]})^2}{\dim(mirna)} \quad (4-4)$$

تغییر داده شد تا میزان تاثیر مجموعه داده گان روی embedding متأثر از اندازه ی ابعادشان نباشد. پس از بررسی نتایج برای نسخه های مختلف این معماری (تغییر ابعاد لایه های مختلف و یا تغییر تعداد لایه ها) کماکان نتیجه نگرفتیم و حدس مان این بود که دلیلش بیش برآزش^۴ باشد چون ابعاد داده در مقایسه

^۲ Concatenate

^۳ دراکثر نسخه های کد این پژوهش هم فاصله ی cosine و هم فاصله ی اقلیدسی در embedding ها بررسی شد اما عموماً فاصله ی اقلیدسی نتیجه ی بهتری می داد.

^۴ Overfitting



شکل ۴-۱: نمای امبدینگ معماری نسخه‌ی ۱۲ در دوبعد با استفاده از TSNE

با تعداد نمونه‌ها خیلی زیاد بود (۱۳۸۸۱ در مقابل ۲۱۵). برای رفع این مشکل اولین ایده استفاده از لایه‌های Dropout بود.

۴-۱-۳ نسخه‌ی ۱۲

در نسخه‌ی ۱۲ به معماری خودرمزگذار لایه‌های Dropout اضافه شد و دوباره اندازه‌های مختلف ابعاد برای لایه‌های شبکه بررسی شد. علاوه بر بررسی های قبلی که روی embedding انجام می‌شد، با استفاده از PCA و TSNE تمام مریض‌ها را در embedding تصویرسازی کردیم. همانطور که از شکل ۴-۱ پیداست هر مریض یک دایره است و دایره‌های بزرگتر زمان برزیستی‌شان بیشتری دارند. کماکان نتایج خوبی نگرفتیم و مسئله‌ی بیش‌برازش مشهود بود.

۴-۱-۴ نسخه‌ی ۱۳

به تابع هزینه‌ی مدل قبلی رگولاریزیشن L_1 روی ماتریس‌های وزن رمزگذار اضافه کردیم. توجیه‌مان این بود که چون ابعاد نمونه‌ها بسیار زیاد است، احتمالاً همه‌ی ویژگی‌ها مهم نیستند و بتوان با تعداد کمتری از ویژگی‌ها embedding مناسبی ساخت. همانگونه که می‌دانیم رگولاریزیشن L_1 سعی می‌کند ماتریس را تنک و درایه‌های بی‌تاثیر را صفر کند.

۴-۱-۵ نسخه‌ی ۱۴

برای کاهش بیشتر بیش برآزش قبل از اینکه خروجی بخش رمزگذار را به رمزگشا بدهیم، یک نویز از جنس $\mathcal{N}(0, \sigma^2 I)$ به آن اضافه کردیم. توجه کنید این معماری خودرمزگذار متغیر نبود چرا که کماکان تابع هزینه‌ی آن

$$(X - X')^2 + \lambda l_1(W) \quad (4-5)$$

و فاقد عبارت KL Divergence بود.

۴-۱-۶ نسخه‌ی ۱۵

در نسخه‌های ۱۳ و ۱۴ برای اینکه فقط تاثیر یک تغییر جدید در مقایسه با خودرمزگذار ساده بررسی شود، از dropout استفاده نشد. برای همین در این نسخه به صورت همزمان از dropout، رگولاریزیشن l_1 و نویز گاوسی روی embedding بهره بردیم.

۴-۱-۷ نسخه‌ی ۱۶

به معماری نسخه‌ی قبلی تابع هزینه‌ی مربوط به KL Divergence را اضافه کردیم و شبکه تبدیل به یک خودرمزگذار متغیر شد که ماتریس کواریانس مربوط به توزیع نمونه‌گیری در embedding به صورت $\mathcal{N}(0, \sigma^2 I)$ بود که σ یک عدد ثابت است.

۴-۱-۸ نسخه‌ی ۱۷

در این نسخه معماری خودرمزگذار نویزدا^۵ بررسی شد. تفاوت این خودرمزگذار با خودرمزگذار ساده، افزودن نویز $\mathcal{N}(0, \sigma^2 I)$ به داده‌ی ورودی و تلاش شبکه برای بازسازی داده‌ی اولیه بدون نویز است.

۴-۱-۹ نسخه‌ی ۱۸

به خودرمزگذار متغیر نسخه‌ی ۱۶ dropout و رگولاریزیشن l_1 اضافه شد.

۴-۱-۱۰ نسخه‌ی ۲۰ تا ۲۳

از این نسخه به بعد ایده‌ی خودرمزگذار کلاسیک (که فقط روی فضای ویژگی عمل می‌کند) کنار گذاشته شد و ایده‌ای مشابه [۱۳] پیاده‌سازی شد. ساختار توجهی^۶ اضافه کردیم که به هر مریض ۳ عدد نسبت دهد که مجموعشان ۱ و همه بزرگتر از ۰ باشند. (خروجی لایه‌ی softmax هستند و هر کدام از آن‌ها بیانگر اهمیت آن مریض در هریک از ۳ مجموعه داده است). سپس ضرایب مربوط به هر دید را به صورت ماتریس قطری درآوردیم تا ماتریس‌های لاپلاسین را با توجه به آن‌ها جمع بزنیم.

$$\hat{L} = g_1 L^{(1)} g_1 + g_2 L^{(2)} g_2 + g_3 L^{(3)} g_3 \quad (۴-۶)$$

این ساختار توجه به جای اینکه به هر دید یک ضریب اهمیت بدهد، برای هرفرد در هر دید یک ضریب در نظر می‌گرفت. علت اینکه برخلاف [۱۳] g_i از دو طرف در $L^{(i)}$ ضرب می‌شود، این است که ماتریس لاپلاسین باید متقارن و مثبت نیمه‌معین باشد تا بتوانیم خوشه‌بندی طیفی را روی آن اجرا کنیم. سپس از \hat{L} به عنوان کرنل واپخشی در لایه‌های GCN بخش رمزگذار استفاده می‌شد. تابع فعال‌سازی لایه‌ی آخر بخش رمزگذار نیز softmax بود تا هر بعد embedding متناظر با یکی از خوشه‌ها باشد (طبق [۷] داده‌ی gbm ۳ خوشه‌ای بود) و اندازه‌ی هر نمونه بیمار در آن بعد بیانگر احتمال تعلقش به خوشه‌ی متناظرش شود. این خاصیت تابع softmax که سعی می‌کند یک بعد را نزدیک به ۱ و بقیه را تقریباً صفر کند آن را برای هدف مذکور مناسب‌تر می‌کرد. همچنین لازم به ذکر است که این معماری کاملاً بدون استفاده‌ی مستقیم از ویژگی‌ها آموزش دید و بردار one-hot رؤس به عنوان ورودی سیگنال به شبکه داده شد. بخش رمزگشا دارای یک لایه با تابع فعال‌سازی سیگموئید بود تا همان بردارهای one-hot بازسازی شوند. (تابع هزینه به صورت $(\tilde{X} - \tilde{X}')^2$ تعریف شد) در این معماری و نسخه‌های بعدی هم قسمت embedding و هم قسمت \hat{L} بررسی شد (روش‌های مختلف خوشه‌بندی مثل خوشه‌بندی طیفی). در معماری نسخه‌ی ۲۰ embedding کم بعد بود و حدس زده شد که ۳ بعد برای رمز کردن این همه اطلاعات کافی نیست لذا ابعاد آن را از ۳ به ۱۰ افزایش دادیم. در نسخه‌ی ۲۲ این عدد به ۴۰ افزایش یافت و در نسخه‌ی ۲۳ بعد embedding ۱۳۰ شد.

^۶ Attention

۱۱-۱-۴ نسخه‌ی ۲۴ تا ۲۷

تابع هزینه به فاصله‌ی ۱ تغییر یافت ($|\tilde{X} - \tilde{X}'|$) و برای embedding های به طول ۳ و ۱۰ و ۴۰ و ۱۳۰ آزمایش شد.

۱۲-۱-۴ نسخه‌ی ۲۸ تا ۳۰

تابع هزینه‌ی Log Sigmoid به جای فاصله‌ی نرم ۱ برای چندین اندازه‌ی embedding (۳، ۱۰، ۴۰) بررسی شد.

۱۳-۱-۴ نسخه‌ی ۳۱ تا ۳۴

همانطور که پیشتر گفته شد از معماری ۲۰ به بعد طبق [۱۳] تابع فعال‌سازی بخش رمزگذار soft-max و بخش رمزگشا sigmoid بود. اما تجربه‌مان نشان داد که softmax آموزش شبکه را دشوار می‌کند. از این رو تابع فعال‌سازی رمزگذار را به relu تغییر دادیم و برای embedding های به اندازه‌ی ۳، ۱۰، ۴۰، ۱۳۰ شبکه را بررسی کردیم.

۱۴-۱-۴ نسخه‌ی ۳۵

این بار تابع فعال‌سازی بخش رمزگشا را به Softmax و تابع هزینه را به Cross Entropy تغییر دادیم.

۱۵-۱-۴ نسخه‌ی ۳۶ تا ۳۹

معماری بخش توجه را به صورتی تغییر دادیم که به جای اینکه ماتریس مجاورت ها را به عنوان ورودی بگیرد، ماتریس X که برارهای onehot رئوس گراف بود را دریافت کند. سپس شبکه را برای em-bedding های به اندازه‌های ۱۵۰، ۷۰، ۴۰، ۱۰ بررسی کردیم.

۴-۱-۱۶ نسخه‌ی ۴۰

تابع فعال‌سازی رمزگذار را relu و رمزگشا را softmax کردیم. همچنین تابع هزینه را به فاصله‌ی نرم ۱ تغییر دادیم.

۴-۱-۱۷ نسخه‌ی ۴۱

در این نسخه به جای اینکه بردار one-hot رؤس را به‌عنوان سیگنال ورودی به شبکه دهیم، یکی از سه داده را (مثلاً) mirna به شبکه می‌دادیم و برخلاف قبل که ماتریس \hat{L} حاصل جمع وزن دار سه ماتریس بود، آن را حاصل جمع توجه‌وار دو ماتریس وزن دار دیگر کردیم (methy و gene).

۴-۱-۱۸ نسخه‌ی ۴۲

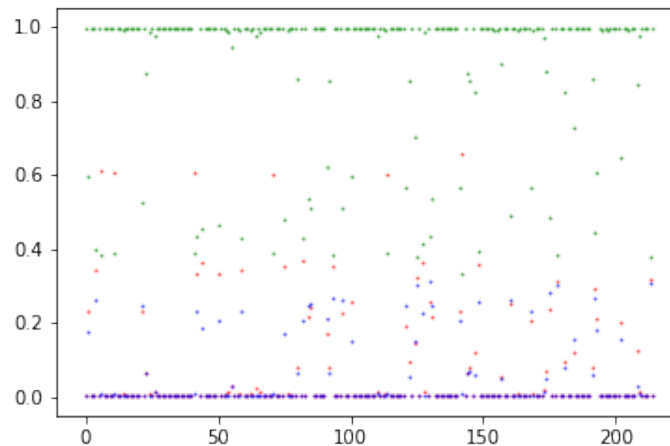
همان معماری قبلی را به‌کار بردیم اما به شبکه‌ی توجه مجموعه‌داده‌ی سیگنال (mirna) را به جای L_i ها ورودی دادیم.

۴-۱-۱۹ نسخه‌ی ۴۳ و ۴۴

دوباره تابع فعال‌سازی بخش رمزگذار را به softmax تغییر دادیم.

۴-۱-۲۰ نسخه‌ی ۴۵ تا ۴۹

از آنجا که قبلاً تمام خودرمزگذارهایی که پیاده‌سازی کرده بودیم (نسخه‌ی ۱ تا ۲۰) تابع هزینه‌شان فاصله‌ی نرم ۱ یا نرم ۲ داده با نسخه‌ی بازسازی‌اش بود، این بار تصمیم گرفتیم مطابق ایده‌ی [۱۵] از تابع هزینه‌ی logloss استفاده کنیم. معماری به‌صورت کلی یک خودرمزگذار متغیر روی خودویژگی‌ها (الحاق سه مجموعه داد) با لایه‌های dropout بود که تابع هزینه‌ی logloss داشت. همچنین مطابق [۱۵] با $\text{Cox Partial Likelihood}$ تک متغیره میزان اهمیت تک تک ابعاد embedding را بررسی کردیم.



شکل ۴-۲: نمودار وزن‌های ساختار توجه. قرمز مربوط به gene سبز مربوط به methy و آبی مربوط به mirna است. بعد افق شماره‌ی راس گراف هارا نشان می‌دهد و بعد عمود بزرگی عدد احتمالی. برای مثال نقطه‌ی آبی رنگ با مختصات $(43, 0/7)$ به این معنا است که $g^{mirna}(25) = 0/7$ یعنی ضریب توجه برای راس ۴۳ ام در دید mirna ۰/۷ است و لذا در نتیجه مجموع دو دید methy و gene برای این راس ۰/۳ است. همانطور که از نمودار پیداست، اثر مذکور رخ داده و دید methy غلبه کرده‌است.

۴-۱-۲۱ نسخه‌ی ۵۰

در معماری‌های ۲۰ تا ۴۹ که از ساختار توجه برای ترکیب خطی ماتریس‌های $L^{(i)}$ استفاده کردیم. نمودارهای g_{mirna} و g_{methy} و g_{gene} را رسم می‌کردیم ۴-۲ و در اکثر مواقع g مربوط به یکی از داده‌ها بزرگ و غالب می‌شد و دو مجموعه‌داده‌ی دیگر عملاً بی‌اثر می‌شدند. برای جلوگیری از این مسئله ایده‌ای مشابه [۱۶] را پیاده سازی کردیم.

$$\sum_{i=1}^{i=3} \text{mean}(g_i) \times \log(\text{mean}(g_i)) \quad (4-7)$$

عبارت ۴-۷ را به تابع هزینه افزودیم تا از غلبه‌ی یک $L^{(i)}$ خاص روی \tilde{L} جلوگیری شود.

۴-۱-۲۲ نسخه ۵۱

از این نسخه به بعد از پکیج rpy۲ استفاده کردیم تا بخش‌هایی از کد که در محیط R باید اجرا می‌شدند در همان محیط پایتون در فضای colab قابل اجرا باشند. پیشتر بعد هربار آموزش شبکه، ماتریس embedding و \hat{L} به صورت فایل از محیط پایتون استخراج می‌شدند و به منظور تحلیل در محیط R دوباره بارگذاری می‌شدند که فرآیند زمان‌بری بود. این پکیج کمک کرد که کدهای R در فضای پایتون قابل اجرا باشند و همچنین این امکان را فراهم کرد که حتی در حین آموزش شبکه نیز تمام آنالیزها روی embedding و \tilde{L} به صورت لحظه به لحظه قابل نمایش باشد. معماری را نیز تغییر دادیم و با اقتباس از [۱۶] عبارت

$$\lambda \text{trace}(L^T \tilde{A}L) \quad (۸-۴)$$

را به تابع هزینه اضافه کردیم. L خروجی یک لایه MLP است و ابعادش به صورت $N \times c$ است که c تعداد خوشه‌های طیفی گراف با ماتریس لاپلاسی \hat{L} است. درواقع هر ستون این ماتریس، بردار معرف یکی از خوشه‌ها است. از آنجا که این بردار معرف‌ها باید متعامد یک γ باشند، عبارت

$$\mu(L^T L - I) \quad (۹-۴)$$

را هم به تابع هزینه افزودیم.

۴-۱-۲۳ نسخه ۵۲

به معماری نسخه قبلی رگولاریزشن ۱۲ روی وزن‌های بخش رمزگذار افزودیم و همچنین با اقتباس از [۲۲] روی ماتریس \hat{L} رگولاریزشن ۱۱ گذاشتیم. طبق توضیحی که در آن مقاله داده شده بود، تنک کردن ماتریس لاپلاسی موجب مشخص‌تر شدن خوشه‌ها می‌شود. یعنی یال‌های درون خوشه‌ای تقویت و یال‌های بین خوشه‌ای تضعیف می‌شوند.

۴-۱-۲۴ نسخه ۱ تا ۹

نسخه‌های ۱ تا ۹ ابتدایی‌ترین مدل‌ها بودند و کدشان به صورت مرتب آرشیو نشد. ولی ایده‌ی کلی بدین صورت بود که دقیقاً معماری کیف [۱۱] را پیاده‌سازی کردیم. بدین صورت که مثلاً داده‌ی

Orthonormal ^y

mirna و methy را که ابعادشان کمتر بود الحاق کردیم و به عنوان سیگنال ورودی به شبکه دادیم (به جای بردارهای وجود کلمات) و از L_{gene} در لایه‌های GCN استفاده شد (به جای ماتریس ارجاع‌دهی مقالات). علت ناموفق بودن این نسخه‌ها عدم ترکیب کافی مجموعه‌دادگان و تفاوت ماهیتی مسئله‌ی ما با مسئله‌ی کیف بود:

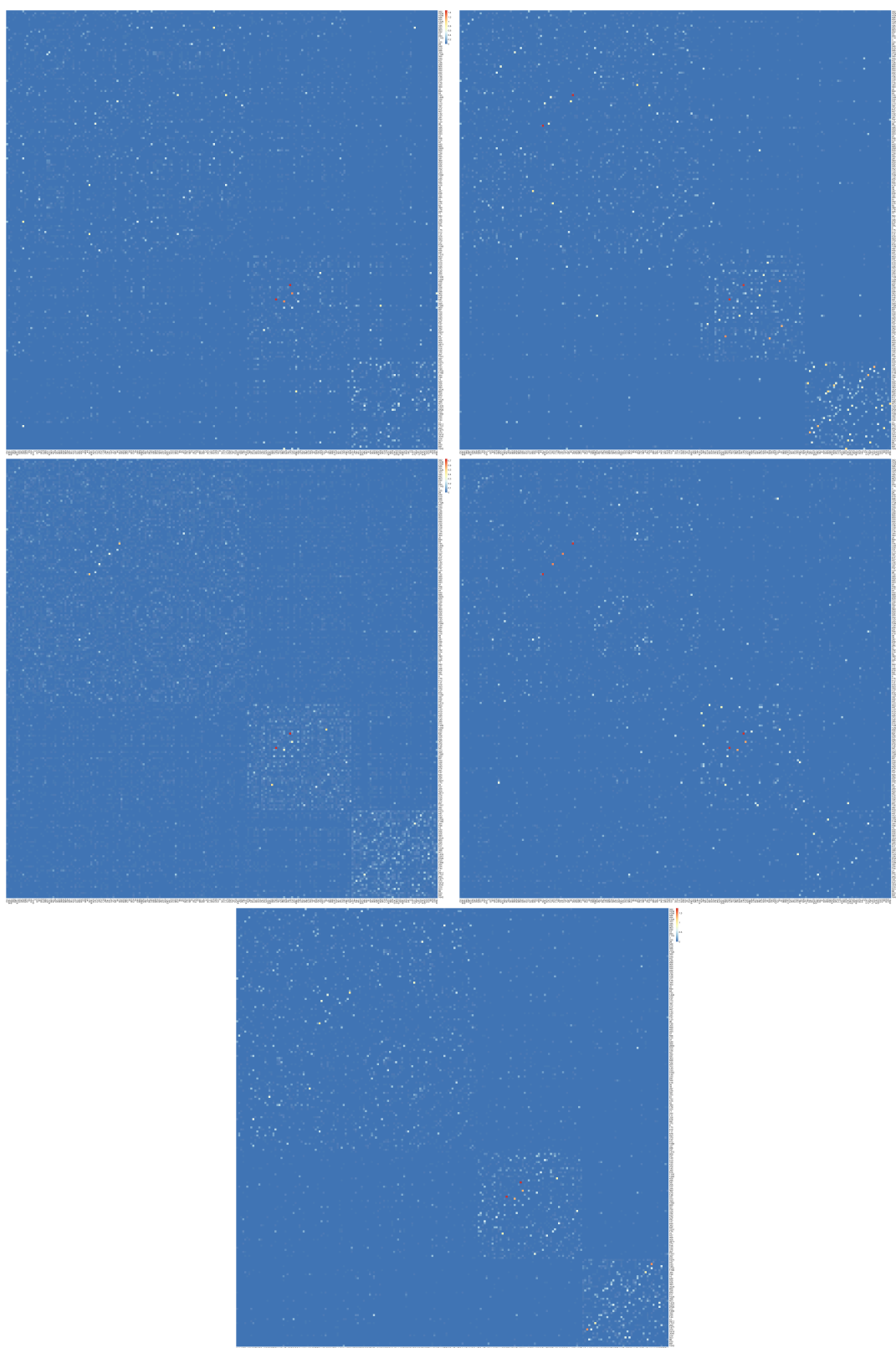
۱. در معماری کیف هدف ترکیب چندین مجموعه‌داده نبود بلکه می‌خواستیم یک ماتریس مجاورت ناقص را تکمیل کنیم. برای همین وقتی این معماری را روی مجموعه‌داده‌ی خودمان بکار بردیم در نهایت همان ماتریس مجاورت gene بازسازی شد. البته خود embedding را هم مطالعه کردیم ولی اطلاعات مفید و قابل اعتمادی نداشت.

۲. در معماری کیف ماتریس مجاورت تمامی یال‌هایش درست بود (یال غلط نداشت) و مشکلی صرفاً نداشتن تعدادی یال درست بود. اما در مسئله‌ی ما هر سه ماتریس‌های مجاورت هم دارای یال غلط و هم فاقد یال درست بودند.

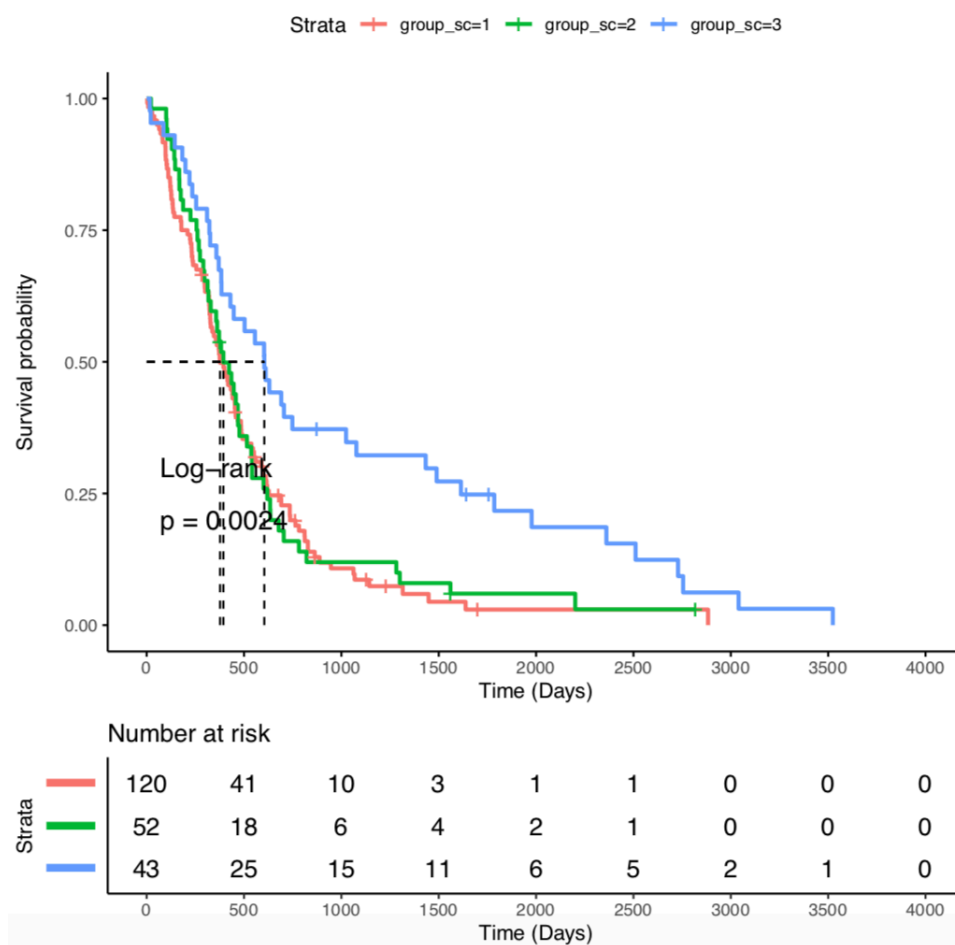
۴-۲ کد مربوط به آنالیزهای بعد آموزش شبکه

این کد نیز در آدرس اینترنتی که ارائه شد موجود است و به زبان R نوشته شده است. (تا قبل از اینکه پکیج $rp2y$ استفاده‌شود انواع آنالیزها در این محیط انجام می‌شد.) در ادامه قسمت‌هایی از آنالیزهای این کد روی معماری نسخه ۱۵ آورده شده است ($K = 30$ & $\mu = 0.3$ & $T = 17$). در مقاله‌ی [۷] به صورت تجربی فرض شده که داده‌ی GBM به صورت سه خوشه‌ای بوده و مقایسه‌ی p value برای خوشه‌بندی داده به سه خوشه انجام شده است. از این رو نتایجی که در ادامه آمده‌اند هم مربوط به خوشه‌بندی ۳ تایی هستند. هرچند ما هم embedding و هم ماتریس لاپلاسین حاصل از هر معماری را به خوشه‌های ۱ تا ۱۵ تایی خوشه‌بندی می‌کردیم تا حالات دیگر هم بررسی شوند کماآنکه در [۲۳] نشان داده شده است که خوشه‌بندی این مجموعه دادگان به ۱۵ خوشه p value های بهتری می‌دهد.^۸

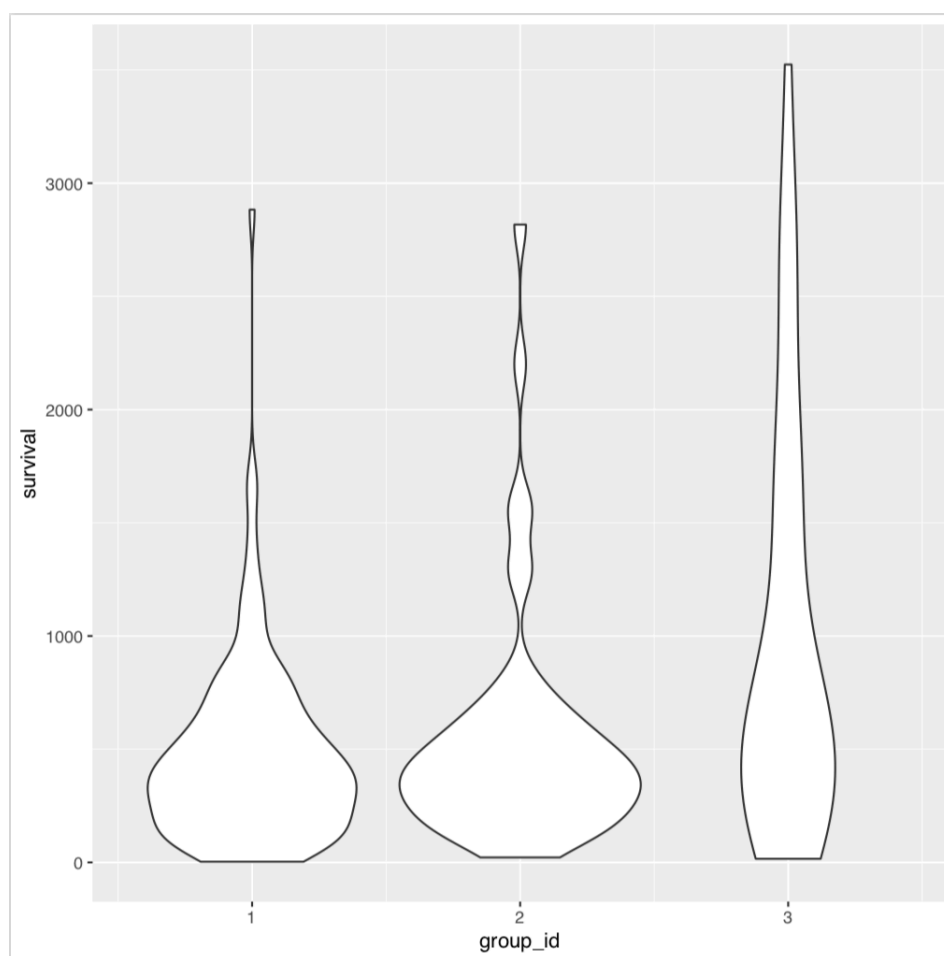
^۸ برای سرطان GBM $6/5 \times 10^{-6}$ نتیجه می‌دهد که از 2×10^{-4} در مقاله‌ی snf بهتر است.



شکل ۴-۳: تصاویر حاصل از اجرای خوشه‌بندی طیفی روی دیدهای مختلف. با ترتیب سطری از راست به چپ و از بالاترین سطر به پایین‌ترین: gene، methy، mirna، ترکیب سه دید با snftool و embedding حاصل از آموزش شبکه‌ی ۱۵.



شکل ۴-۴: نمودار Kaplan-Meier برای خوشه‌بندی طیفی embedding حاصل از معماری ۱۵



شکل ۴-۵: نمودار violin برای خوشه‌بندی طیفی embedding حاصل از معماری ۱۵

Gene	Methy	Mirna	Cross Diffusion	Embedding
0.00678	0.99879	0.26649	0.00680	0.00142

جدول ۴-۱: مقدار P Value برای Cox Partial Likelihood خوشه‌بندی روی دیدهای مختلف.

فصل ۵

نتیجه‌گیری

در این نوشتار مجموعه‌ای از تلاش‌ها و ایده‌های مبتنی بر واپخشی و خودرمزگذار گراف برای ترکیب چندین مجموعه داده‌ی ژنتیکی ارائه کردیم تا خوشه‌بندی مناسبی از جهت برزیستی ایجاد کنیم. متأسفانه نتوانستیم به روش جدید و صحیحی دست یابیم که با اطمینان آن را ارائه دهیم. اما به هرحال نگارش این تلاش‌ها ممکن است برای پژوهش‌های آتی در این زمینه ایده‌بخش و مفید باشد. در طی این پژوهش متوجه شدیم آموزش شبکه‌های دارای لایه‌ی GCN فرآیند آموزش خودرمزگذارها را دشوارتر از آن‌چه که هست می‌کند و تعداد ایپوک‌های بسیار بیشتری طول خواهد کشید تا به دقت مطلوب برسد، مخصوصاً اگر از بیشتر از ۲-۳ لایه متوالی استفاده شود. مشکل بعدی ناشی از تفاوت هدف شبکه‌ها با تحلیل ما بود. عمده‌ی شبکه‌هایی که طراحی شدند به دنبال ترکیب دیدها و ارائه‌ی embedding بودند. این درحالی است که هدف نهایی ما خوشه‌بندی بود و ما به دنبال معماری بودیم که اگر هرچندبار از اول آن را روی مجموعه‌ی داده‌گان آموزش دهیم به خوشه‌بندی تقریباً یکسان و با P Value بهتری از خط مبنا ^۱ ۰/۰۰۰۲ دست یابد. اما همواره در روش‌های ما خوشه‌بندی یک آنالیز پسا آموزشی بود و روی تابع هزینه و نحوه‌ی یادگیری شبکه تأثیر نداشت (بجز نسخه‌های اخیر مثل ۵۰ و ۵۱ که از [۱۶] ایده‌برداری کردند). اما مقایسه‌ی embedding خودرمزگذارها کار چالش برانگیزی است چراکه با هربار آموزش دادن آن‌ها روی یک مجموعه‌ی داده‌گان ثابت یک embedding کاملاً جدید به دست می‌آید و نباید انتظار داشت که خوشه‌های آن ثابت بماند. به همین دلیل توصیه می‌شود که پژوهش‌های آتی روی این مسئله به دنبال شبکه‌های مبتنی بر خوشه‌بندی باشند و مفهوم خوشه‌بندی را به صورت مستقیم در معماری و

Baseline^۱

تابع هزینه وارد کنند تا در طی آموزش embedding مناسبی برای این منظور ایجاد شود. مثلاً [۲۴] یک معماری از این نوع است و تبدیل آن به یک شبکه‌ی چنددیدگاهی^۲ می‌تواند یک مسیر پژوهشی برای آینده باشد. یکی از پژوهش‌هایی که این ایده را پیش گرفت، [۲۵] بود اما نواقصی دارد:

۱. در رابطه‌ی (۳) از آن مقاله که تابع هزینه تعریف شده امکان دارد ضرایب مربوط به چندتا از دیدها نزدیک صفر و تعدادی دیگر بسیار بزرگ شود و عملاً ترکیب دیدها به مساوات صورت نگیرد لذا افزودن عباراتی همچون

$$\lambda \sum_v a^{(v)} \log a^{(v)}$$

از این مشکل جلوگیری می‌کند

۲. تابع هزینه‌ی این پژوهش به دنبال یافتن $Y \in \mathbb{R}^{N \times c}$ ای است که بیانگر بهترین خوشه‌بندی در مجموع برای تمام $L^{(v)}$ ها باشد. لذا ایده‌ی ترکیب کردن دیدها در آن به صورت توافق همگانی است و نه تکمیل کردن یکدیگر. تلاش برای ترکیب دادگان به وجهی که یکدیگر را تکمیل کنند می‌تواند موضوع پژوهش آتی باشد.

مراجع

- [1] M. Soleymani. Variational auto-encoder (vae) [slideset]. 2019.
- [2] M. Soleymani. Generative models [slideset]. 2019.
- [3] A. N. Yifeng Li, Fang-Xiang Wu. A review on machine learning principles for multi-view biological data integration. 2019.
- [4] R. S. Lenore Cowen, Trey Ideker.
- [5] C. Z. Zonghan Wu, Shirui Pan. A comprehensive survey on graph neural networks. 2013.
- [6] P. V. David I Shuman, Sunil K. Narang. The emerging field of signal processing on graphs. 2013.
- [7] A. G. Bo Wang, Aziz M Mezlini. Similarity network fusion for aggregating data types on a genomic scale.
- [8] Z. T. Bo Wang, Jiayan Jiang. Unsupervised metric fusion by cross diffusion. 2012.
- [9] Z. T. Bo Wang, Jiayan Jiang. Unsupervised metric fusion by cross diffusion. 2012.
- [10] X. Z. Yang Wang, Wenjie Zhang. Unsupervised metric fusion over multiview data by graph random walk-based cross-view diffusion. 2017.
- [11] M. W. Thomas N. Kipf. Variational graph auto-encoders. 2016.
- [12] M. W. Thomas N. Kipf. Semi-supervised classification with graph convolutional networks. 2017.
- [13] F. W. Tengfei Ma, Cao Xiao. Drug similarity integration through attentive multi-view graph auto-encoders. 2018.

-
- [14] N. K. Martin Simonovsky. Graphvae: Towards generation of small graphs using variational autoencoders. 2018.
 - [15] L. X. G. Kumardeep Chaudhary, Olivier B. Deep learning based multi-omics integration robustly predicts survival in liver cancer. 2017.
 - [16] E. P. S. B. Bo Wang, Junjie Zhu. Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. 2017.
 - [17] I. D. Xiang Wang, Buyue Qian. On constrained spectral clustering and its applications. 2012.
 - [18] I. D. Hongjing Zhang, Sugato Basu. A framework for deep constrained clustering - algorithms and advances. 2019.
 - [19] J. Z. Inci M. Baytas, Cao Xiao. Patient subtyping via time-aware lstm networks. 2019.
 - [20] J. H. Yeqing Li, Feiping Nie. Large-scale multi-view spectral clustering via bipartite graph. 2016.
 - [21] A. A. M. Mehmet Go"nen. Localized data fusion for kernel k-means clustering with application to cancer biology. 2014.
 - [22] S. Y. Canyi Lu and Z. Lin. Convex sparse spectral clustering: Single-view to multi-view. 2018.
 - [23] N. P. Nora K. Speicher. Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. 2015.
 - [24] H. L. Uri Shaham, Kelly Stanton. Spectralnet: Spectral clustering using deep neural networks. 2018.
 - [25] J. L. Zhenyu Huang, Joey Tianyi Zhou. Multi-view spectral clustering network. 2019.

واژه‌نامه

الف	
Spectral Clustering خوشه‌بندی طیفی	Concatenate الحاق
Baseline خط مبنا	Epoch ایپاک
ب	
View دید	Unsupervised بدون سرپرستی
برزیستی	
Survival	Overfitting بیش برازش
ت	
Voting رای‌گیری	Consensus توافق همگانی
Encoder رمزگذار	
Decoder رمزگشا	
ط	
Spectral طیفی	چ
چنددیدگاهی	
Multi-view	
خ	
Similarity Matrix ماتریس شباهت	Autoencoder خودرمزگذار
Orthonormal متعامد یکه	Variational Autoencoder خودرمزگذار متغیر
خودرمزگذار نویزدا	
Denoising Autoencoder	
ن	

Multivariate Normal..... نرمال چندبعدی

هـ

Smoothness..... همواری

و

Diffusion..... واپخش

Random Walk..... ولگشت