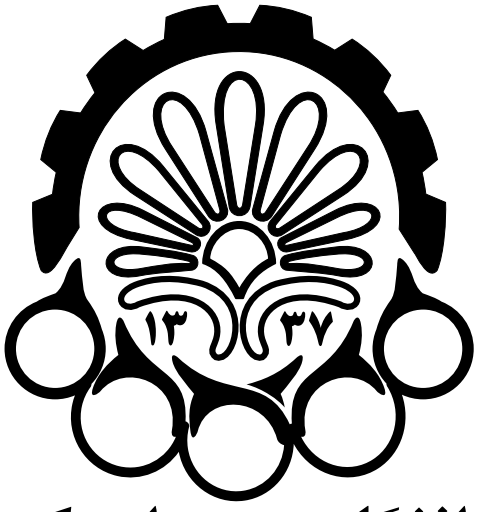


دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

درس : کلان داده

نام و نام خانوادگی : فاطمه توکلی

۴۰۰۱۳۱۰۱۶

گزارش پایانی

در ابتدا به صورت خلاصه به بیان مسئله و روش ارائه شده در آن میپردازیم:

خوشه بندی داده‌ها نامتوازن یک مشکل چالش برانگیز در یادگیری ماشین است. مشکل اصلی ناشی از عدم تعادل در اندازه خوشه و توزیع چگالی داده است. برای پرداختن به این مشکل، یک الگوریتم خوشه‌بندی جدید به نام LDPI را بر اساس پیک‌های چگالی محلی در این مقاله پیشنهاد شده‌است.

الگوریتم پیشنهادی LDPI شامل سه مرحله است:

۱) یک طرح تولید زیر خوشه اولیه طراحی شده است که می تواند به طور خودکار نقاط نویز و مراکز زیر خوشه اولیه را شناسایی کند. خوشه‌های فرعی با طبقه بندی نقاط در مراکز زیر خوشه بر اساس اصل نزدیکترین همسایه ایجاد می شوند. ۲) یک روش به روز رسانی زیرخوشه معرفی شده است که می تواند مراکز زیر خوشه کاذب را از مراکز زیر خوشه اولیه شناسایی کند. متعاقباً، طرح آنها را حذف می کند تا مراکز فرعی به روز شده و زیرخوشه های به روز شده مربوطه را به دست آورد.

۳) یک استراتژی ادغام زیرخوشه برای ادغام زیرخوشه‌های به روز شده و ایجاد خوشه‌های نهایی اعمال می شود.

۱) تعیین آستانه فاصله مناسب dc و چگالی

همانطور که قبلاً بحث شد، تخصیص دستی d_c در DPC معمولاً منجر به خوشه‌های نادرست، به ویژه با مجموعه داده‌های نامتعادل می‌شود. در راستای حل این مشکل برای یک مجموعه داده با n نقطه،اگر D_i فاصله بین نقطه i و نزدیکترین همسایه آن در نظر بگیریم، سپس مقدار حداکثر تمام فواصل بین هر نقطه و نزدیکترین همسایه آن به صورت زیر محاسبه می‌شود:

$$d_c = \max_{i=1}^{n}\{D_i\}$$

که این را به عنوان آستانه فاصله d_c در نظر میگیرد. موضوع دیگر، تعریف چگالی مناسب نقاط است.

در DPC چند نقطه ممکن است چگالی یکسانی داشته باشند و در نسخه بهبود یافته آن از معادله‌ای استفاده میکند که برای محاسبه چگالی هر نقطه بایستی تمامی نقاط در معادله گذاشته شوند و چگالی به دست آید.

در اینجا، برای غلبه بر اشکال روش بهبود یافته، به جای تمام نقاط موجود، چگالی محلی هر نقطه را با نقاط مجاور آن تعیین می‌کند، چگالی محلی هر نقطه را به صورت زیر محاسبه می‌کند

$$\rho_i = \sum_{j \in S/\{i\}, d_{ij} \le d_c} e^{-(\frac{d_{ij}}{d_c})^2}$$

۲) تعیین نقاط نویز توسط یک گراف تصمیم جدید

نقاط نویز نمودار تصمیم تاثیر منفی قوی بر نتایج خوشه بندی دارند. بنابراین باید آنها را شناسایی کرد. در DPC، نقاطی با چگالی کمتر و معیار upward distance بیشتر به عنوان نقاط نویز در نظر گرفته می شوند، که upward distance برای هر نقطه X_i صورت زیر تعریف می‌شود:

$$\delta_i = \begin{cases} \max_j\{d_{ij}\}, & if\ \rho_i > \rho_j\ for\ \forall j \\ \min_j\{d_{ij}|\rho_j > \rho_i\}, & otherwise. \end{cases}$$

اما با مجموعه داده های نامتعادل، برخی از مراکز خوشه کوچک به اشتباه به عنوان نقاط نویز اختصاص داده می شوند. برای شناسایی صحیح نقاط نویز در یک مجموعه داده نامتعادل، یک نمودار تصمیم گیری جدید معرفی شده که از تعداد معکوس نزدیکترین همسایگان هر نقطه استفاده می‌کند. و در صورتی که سه شرط زیر برقرار بود نقطه به عنوان نویز یا داده پرت در نظر گرفته می‌شود:

$$\begin{aligned} &1) \quad \delta_i > \mu(\delta) + \sigma(\delta). \\ &2) \quad \rho_i < \mu(\rho) - \sigma(\rho). \\ &3) \quad RNN_i < \mu(RNN) - \sigma(RNN). \end{aligned}$$

که μ و δ در آن به ترتیب میانگین و انحراف معیار تمام نقاط برای upward distance, densities ,RNN هستند.

۳) طرح اولیه تولید زیر خوشه

برای جلوگیری از تخصیص تعداد خوشه‌ها به صورت دستی و فعال کردن الگوریتم برای خوشه‌بندی خودکار روی مجموعه داده‌های نامتعادل، ابتدا تعداد نسبتاً زیادی از مراکز زیرخوشه را تنظیم کرده و سپس زیرخوشه‌های اولیه را ساختیم. به طور دقیق تر پس از حذف نقاط نویز، نقاط باقیمانده که فاصله آنها بیشتر $\mu(\delta) + \sigma(\delta)$ است به عنوان مراکز زیر خوشه اولیه انتخاب می شوند. از این طریق اطمینان حاصل شود که فواصل بین مراکز مختلف به اندازه کافی بزرگ است. پس از آن، هر مرکز زیر خوشه اولیه i به یک زیر خوشه اولیه C_i مربوط می شود. تمام نقاط باقیمانده به ترتیب نزولی چگالی آنها مرتب می شوند.و به نزدیک مرکز که چگالی آن از چگالی خود نقطه بیشتر است انتصاب می‌یابند.

۴) به روزرسانی زیرخوشه

ابتدا، زیرخوشه‌های کاذب را به صورت زیر شناسایی می‌کنیم: فرض کنید که iامین زیر خوشه اولیه C_i حاوی NC_i نقطه است، و همسایگی مرکز زیر خوشه i با شعاع dc حاوی Ndc_i نقطه است. اگر C_i حاوی کمتر از نیمی از نقاط Ndc_i، یعنی $NC_i < 0:5$ Ndc_i باشد،به این معنیست که چنین خوشه‌ای یک خوشه واقعی نیست و باید حذف شود. سپس، با تخصیص نقاط در زیر خوشه‌های نادرست به نزدیک‌ترین زیرخوشه‌های همسایه، زیرخوشه‌ها به روز می‌شوند.

۵) استراتژی ادغام خوشه‌های فرعی

ابتدا نقاط مرزی هر زیر خوشه مشخص می شود. نقاط مرزی به عنوان نقاطی تعریف می شوند که چگالی آنها کمتر از چگالی متوسط زیرخوشه داده شده‌است.

برای ادغام زیرخوشه‌های به روز شده، اگر S مجموعه داده را نشان دهد، $No(X)$ مجموعه‌ای از نقاط نویز را در S و $In(X)$ نشانگر زیرمجموعه ای از X به استثنای تمام نقاط مرزی X باشد. شعاع ادغام r با معادله زیر تعریف می شود:

$$r = \begin{cases} d_c, & if\ No(S) = \emptyset \\ \max_{x_i \in S/No(S)}\{D_i\}, & otherwise. \end{cases}$$

و با استفاده از این شعاع و شروط زیر برای ادغام به خوشه بندی نهایی میرسد :

برای ادغام زیر خوشه های Cm و Cn، باید نشان دهیم $d(C_m, C_n) = \min\{d_{ij} \mid x_i \in C_m, x_j \in C_n\}$ و دو مورد باید در نظر گرفته شود:

۱) اگر $d(C_m, C_n) > r$; سپس $Cn\ Cm$ و Cn دور هستند و ادغام نمی شوند.

۲) در غیر این صورت

i) اگر $d(C_m, C_n) = \min\{d_{ij} \mid x_i \in C_m, x_j \in C_n\}$، $d(C_m, C_n) = \min\{d_{ij} \mid x_i \in In(C_m), x_j \in In(C_n)\}$، سپس Cm و Cm بسیار نزدیک هستند و می توانند مستقیماً ادغام شوند.

ii) در غیر این صورت،

الف) $(x_t, x_s) = \text{argmin}_{x_i \in C_m, x_j \in C_n}\left(d_{ij}\right)$ نشان می دهد. اگر مجموع چگالی x_t و x_s از میانگین چگالی مراکز زیر خوشه آنها بزرگتر باشد، یعنی اگر $\rho_t + \rho_s > \rho_{C_{mn}}$ سپس C_m و C_n ادغام می‌شوند.

ب) در غیر این صورت، دو خوشه فرعی با هم ادغام نمی شوند.

این فرآیند ادغام تا زمانی تکرار می شود که هیچ جفتی از زیر خوشه‌ها شرایط ادغام را برآورده نکنند. پس از فرآیند ادغام، اگر نقاط نویز باقی نماند، خوشه بندی کامل می شود. در غیر این صورت، هر نقطه نویز به نزدیکترین خوشه خود اختصاص داده می‌شود.

در نهایت با پرداختن به جزئیات بیان می‌کند که حل این مسئله با پیچیدگی زمانی $o(n^2)$ ممکن است.

در نتیجه به عنوان مزایا میتوان به این موارد اشاره کرد که :

۱) نیازی به هیچ پارامتر ورودی ندارد.

۲) قادر است به طور خودکار مراکز خوشه‌بندی و تعداد خوشه‌ها را تعیین کند.

۳) برای مجموعه داده‌های متنوع و نامتوازن با شکل و توزیع دلخواه، مناسب است.

و همچنین به عنوان معایب میتوان به موارد زیر اشاره کرد:

۱) پیچیدگی محاسبانی هنگاهی که مجموعه داده ما بسیار بزرگ است،مراحل محاسبه چگالی و تخصیص خوشه شامل مقایسه هر نقطه داده با همسایگان آن است که با افزایش اندازه مجموعه داده می تواند از نظر محاسباتی هزینه بر شود.

۲) در مواردی که خوشه ها به طور قابل توجهی همپوشانی دارند، ممکن است با چالش هایی روبرو شود، زیرا ممکن است پیک های چگالی به خوبی تعریف نشده باشند، که منجر به طبقه بندی نادرست بالقوه یا نتایج خوشه بندی غیربهینه می شود.

۳) نتایج خوشه بندی تولید شده توسط الگوریتم ممکن است در موارد خاص به راحتی قابل تفسیر نباشد. از آنجایی که الگوریتم تعداد خوشه ها را تعیین می کند و آنها را بر اساس ویژگی های داده تطبیق می دهد، ساختار خوشه ای حاصل ممکن است همیشه با انتظارات شهودی یا دانش حوزه همسو نباشد

۴) یکی از چالش های مطرح در حین پیاده سازی این مسئله نگاشت بین خوشه های به دست آمده و خوشه‌بندی ابتدایی مجموعه داده برای ارزیابی بود که روشی برای آن ارائه نشده است.

من در این پروژه از سه مجموعه‌داده Guassian,Thryoid,Vote استفاده کردم که در ادامه پیش پردازش های لازم برای هرکدام ذکر شده است:

ابتدا داده و لیبل‌ها را تفکیک کردیم، ابعاد مجموعه گاوسین به این صورت است:

```
the shape of gaussian dataframe (2000, 2)
```

بررسی کردیم در هر سطر مقدار null داریم یا خیر:

```
0    0
1    0
dtype: int64
```

سپس آن هارا نرمالایز کردیم و خروجی به صورت زیر می‌باشد:

	0	1
0	0.229078	0.889672
1	0.150356	0.565184
2	0.316992	0.680032
3	0.106285	0.334825
4	0.361934	0.728777
...	...	...

برای مجموعه داده thyroid نیز همین کار را تکرار کردیم:

```
the shape of gaussian dataframe (215, 6)
```

```
0    0
1    0
2    0
3    0
4    0
5    0
dtype: int64
```

	0	1	2	3	4	5
0	1	107	10.1	2.2	0.9	2.7
1	1	113	9.9	3.1	2.0	5.9
2	1	127	12.9	2.4	1.4	0.6
3	1	109	5.3	1.6	1.4	1.5
4	1	105	7.3	1.5	1.5	-0.1

و برای مجموعه داده vote به صورت زیر داریم:

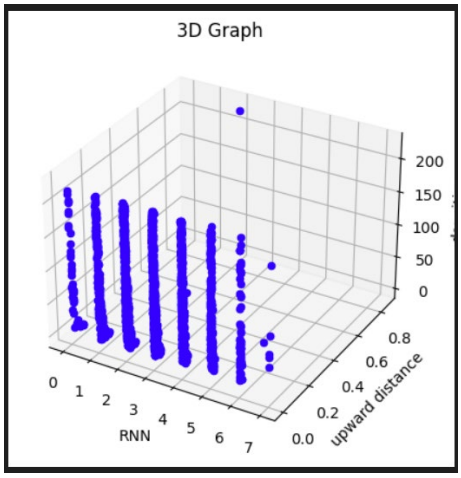
در ابتدا مقادیر null را با مقدار mode هر ستون پر کرده و سپس با استفاده از pd.factorize هرستون را به مقادیر عددی متمایز تبدیل کردیم در ادامه خواهیم داشت:

```
the shape of gaussian dataframe (435, 34)
```

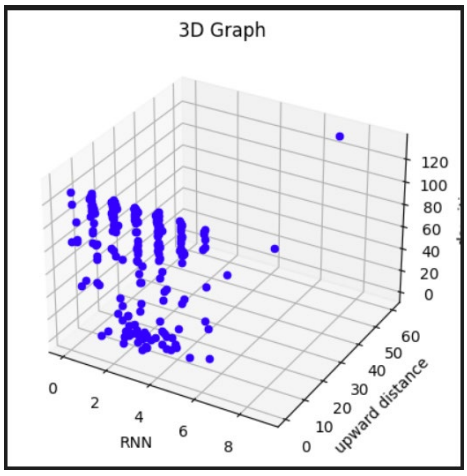
	handicapped-infants_n	handicapped-infants_y	water-project-cost-sharing_n	water-project-cost-sharing_y	adoption-of-the-budget-resolution_n	adoption-of-the-budget-resolution_y	physician-fee-freeze_n	physician-fee-freeze_y	el-salvador-aid_n	el-salvador-aid_y	...	superfund-right-to-sue_n	superfu right su
0	0	0	0	0	0	0	0	0	0	0	...	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	
2	0	0	0	0	1	1	1	1	0	0	...	0	
3	0	0	0	0	1	1	1	1	0	0	...	0	
4	1	1	0	0	1	1	1	1	0	0	...	0	

در این قسمت نیازی به نرمالایز کردن نیست و فقط داده را از لیبل جدا کرده و ادامه میدهیم.

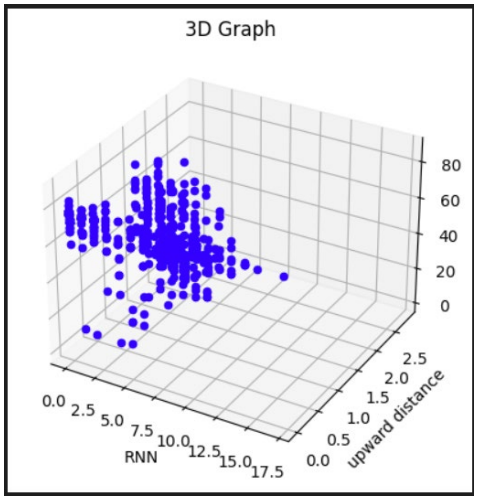
پس از این ها به تعریف توابع پیدا کردن مقدار استانه یافتن مقدار چگالی و سپس upward distance و سپس RNN را پیدا کرده که بر اساس آن ها بتوان نقاط نویزی را پیدا کرد decision graph و رسم کرده :



gaussian



Thyroid



Vote

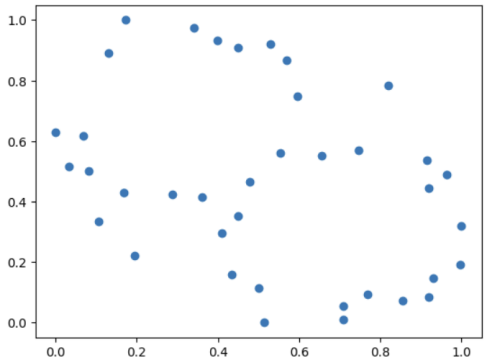
در این قسمت در محاسبه upward distance میبایستی چگالی خود نقطه را حذف میکردیم تا بتوان فواصل را به درستی پیدا کرد.

پس از آن نویز را با توجه به شروط داده شده به دست آوردیم و مراکز اولیه را مشخص کرده و در انتها باقی نقاط را به کلاسترهای موجود انتصاب دادیم.

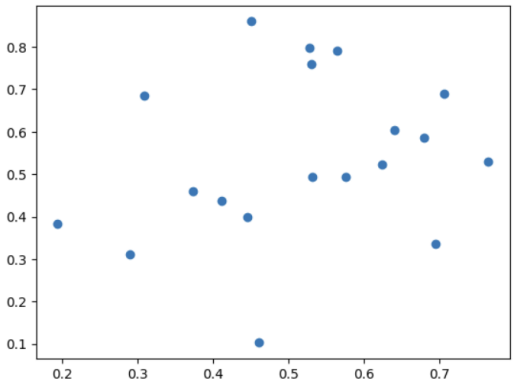
حاصل اعمال الگوریتم ۱ روی تمامی دیتاست ها به صورت زیر است:

Gaussian

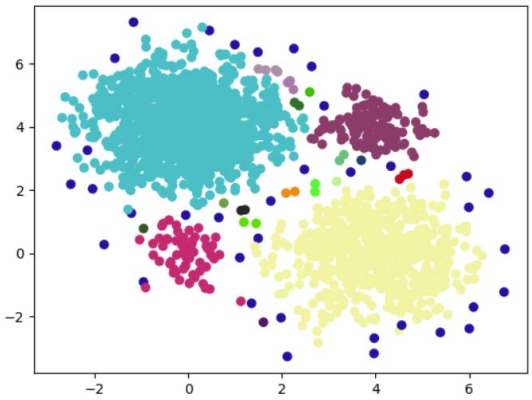
نقاط نویز:



مراکز:



خوشه بندی حاصل از مرحله اول :



در مجموعه داده های Thyroid, Vote به علت ابعاد داده‌ها نمیتوان نتیجه‌ی آن‌ها به صورت بصری مشاهده کرد.

کلاسترهای یکتا به دست آمده پس از اعمال الگوریتم ۲ و به روز رسانی زیرکلاسترها به صورت زیر در آمدند:

Gaussian

```
[0, 1, 2, 10, 14]
```

Thyroid

```
[0, 1, 4, 5, 9, 11, 13, 14, 15]
```

Vote

```
[0, -1, 39]
```

پس از این مرحله به دنبال کلاسترهایی هستیم که بتوان آن ها را ادغام کرد و سپس برای ادغام آن ها با توجه به شرایط گفته شده باید اقدام کرد و در نهایت نویز ها را نیز به نزدیک کلاستر انتصاب میدهیم:

همانطور که در مقاله نیز ذکر شده است برای مجموعه داده گاوسین درمرحله پیش کلاسترینگ به درستی انجام شده و در اینجا لیستی برای ادغام برنمیگرداند الگوریتم:

```
mapped_clusters = sub_cluster_merging(df_gaussian, c, densities, centers)
```

✓ 0.0s

```
[]
```

نتایج نهایی برای هرکدام از دیتاست ها به صورت زیر است:

```
Result  Vote
Accuracy: 0.5586
NMI: 0.0015
Recall: 0.5586
```

```
Result  Thryoid
Accuracy: 0.6977
NMI: 0.1722
Recall: 0.6977
```

```
Result  Gaussian
Accuracy: 0.9300
NMI: 0.8392
Recall: 0.9300
```

که برای دو مورد اول نزدیک به نتایج مقاله است و برای دیتاست vote خیر فکر میکنم یکی از دلایل به روزرسانی این دیتاست و و افزایش تعداد ستون ها و نمونه ها یکی از دلایل کاهش دقت می‌باشد

یکی از چالش هایی که در این مسیر با آن روبرو شدیم هنگام ارزیابی دقت بود که کلاسترهای به دست آمده با شاخص هر کلاستر مشخص میشد درحالیکه در لیبل ها ترتیب خاصی داشت، در این مسئله ما به صورت دستی این نگاشت را انجام

دادیم و به دنبال راه حل های دیگری نیز بودیم که دقت مطلوبی به ما نداد و این یکی از نقاط ضعف مسئله است اگر بخواهیم واقعا در طول پیاده سازی الگوریتم ها کاری به لیبل واقعی نظیر هر مرکز نداشته باشیم.

در حوزه یادگیری ماشین و ارزیابی عملکرد الگوریتم‌ها، معیارهای مختلفی برای اندازه‌گیری عملکرد مدل‌ها وجود دارد. سه معیار متداول برای اندازه‌گیری عملکرد مدل‌ها عبارتند از Accuracy (دقت)، NMI (معیار انطباق اطلاعات نرمالیزه) و Recall (بازخوانی). در زیر تعاریف هرکدام از این معیارها را بررسی می‌کنیم:

Accuracy :

دقت یا Accuracy به میزان تعداد نمونه‌هایی اشاره دارد که به درستی تشخیص داده شده‌اند را نسبت به کل تعداد نمونه‌ها. به عبارت دیگر، دقت مدل برابر است با تعداد پیش‌بینی‌های درست تقسیم بر تعداد کل نمونه‌ها. این معیار بیان‌گر صحت و قدرت پیش‌بینی مدل است.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

NMI :

معیار انطباق اطلاعات نرمالیزه (Normalized Mutual Information) یک معیار برای اندازه‌گیری شباهت بین دو دسته‌بندی مختلف است. این معیار بر پایه اطلاعات مشترک بین دو دسته‌بندی تعیین می‌شود و به عنوان یک معیار کیفیت دسته‌بندی استفاده می‌شود. این معیار مقدار بین ۰ تا ۱ دارد، که مقادیر نزدیک به ۱ نشان‌دهنده انطباق بالا بین دو دسته‌بندی است.

- Normalized Mutual Information:

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]}$$

where,

1) Y = class labels
2) C = cluster labels
3) H(.) = Entropy
4) I(Y;C) = Mutual Information b/w Y and C

Recall :

بازخوانی یا Recall نشان‌دهنده توانایی مدل در شناسایی تمام نمونه‌های مثبت موجود در دسته‌ی مثبت است. به عبارت دیگر، بازخوانی برابر است با تعداد پیش‌بینی‌های درست تقسیم بر تعداد کل نمونه‌های مثبت. این معیار برای مسائلی که تشخیص نمونه‌های مثبت از اهمیت بالایی برخوردارند، مفید است.

$$\text{Recall} = \frac{TP}{TP + FN}$$

معیارهای دیگری نیز برای اندازه‌گیری عملکرد مدل‌ها وجود دارند، اما این سه معیار بسیار متداول هستند و در بسیاری از مسائل استفاده می‌شوند.