داده کاوی

کیوان ایپچی حق – ۹۸۳۱۰۷۳

سوال اول

برای این امر می‌توان از الگوریتم های متعددی – بر اساس تابع heuristic تعریف شده - استفاده کرد و مشکل چندانی پیش نخواهد آمد. ولی در اینجا الگوریتم Density-based Clustering انتخاب خوبی است زیرا می‌توان معیار ۱ تا ۳ امتیاز شباهت را به راحتی تعریف کرد. به گونه ای که ویژگی ها را برداری کرده و روی scatter plot نمایش می‌دهیم (برای اینکار می‌توان از PCA یا روش های دیگر استفاده کرد). سپس حیوانات را کشیده و به مرکز هر حیوان یک radius به شعاع قرار دادی می‌کشیم (اندازه شعاع را ما تعریف می‌کنیم). در نهایت حیواناتی که اشتراک ۳ یا بیشتر ویژگی دارند را ۳ گرفته، اشتراک ۲ ویژگی را ۲ امتیاز و ۱ ویژگی را ۱ امتیاز می‌دهیم (همچنین می‌توان تعریف کرد بالای 2n/3 ویژگی مشترک ۳ امتیاز، بالای n/3 ویژگی مشترک ۲ امتیاز و کمتر از آن ۱ امتیاز). سپس خوشه بندی را بدون نظارت انجام می‌دهیم.

سوال دوم

برای اثبات این موضوع باید به خواص دو تابع مجاورت بپردازیم:

الف) نرم ۱)

$$f(c, d) = |d - c|$$

در الگوریتم خوشه بندی با استفاده از تابع بالا فاصله بین دو نقطه برابر با مجموع قدر مطلق های آنهاست:

$$d(A, B) = |A_1 - B_1| + |A_2 - B_2| + \dots + |A_n - B_n|$$

حال می‌خواهیم نشان دهیم اگر C نقطه مرکزی خوشه انتخاب شود، نقاطی که مجاورت نرم ۱ آنها با C کمتر از نقاط دیگر است داخل دسته C قرار می‌گیرند. برای مثال نقطه D در دسته C قرار میگیرد به طوری که d(C, D) کمتر از d(E, D) برای مجموعه نقاط دیگر است. پس برای هر بعد i داریم:

$$|C_i - D_i| \leq |C_j - D_j| \ (j \neq i)$$

سپس دو طرف تساوی را بر یک عدد مشترک تقسیم می‌کنیم (چون هر دو مثبت هستند):

$$(C_i - D_i) / |C_i - D_i| \leq (C_j - D_j) / |C_j - D_j| \ (j \neq i)$$

همچنین می‌دانیم a<=b اگر و تنها اگر a/b<=1 باشد. پس می‌توان از این فرض برای مسئله خودمان استفاده کنیم:

$$|C_i - D_i| / |C_j - D_j| \leq 1 \ (j \neq i)$$

و در ادامه:

$$|C_i - D_i| + |C_j - D_j| \leq |C_j - D_j| \ (j \neq i)$$

نتیجه گیری آن روابط بالا این است که برای هر بعد i برای D داریم که نسبت به C کمترین فاصله را دارد.

ب) مطابق بخش الف عمل می‌کنیم:

$$f(c, d) = |d - c|$$

$$d(A, B) = \sqrt{((A_1 - B_1)^2 + (A_2 - B_2)^2 + ... + (A_n - B_n)^2)}$$

$$(C_i - D_i)^2 \leq (C_j - D_j)^2 \ (j \neq i)$$

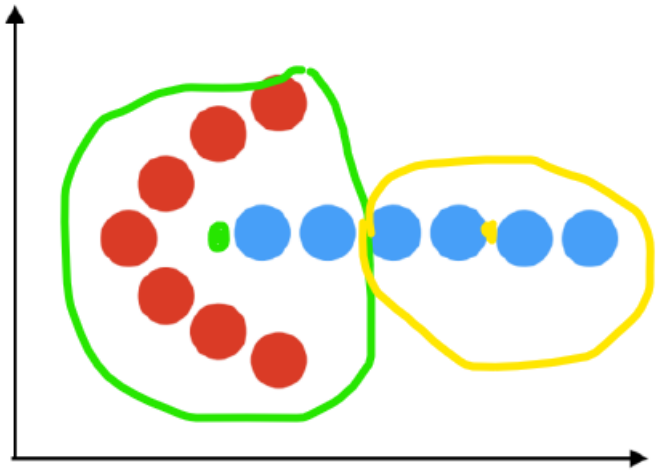
$$(C_1 - D_1)^2 + (C_2 - D_2)^2 + ... + (C_n - D_n)^2 \leq (C_j - D_j)^2 \ (j \neq i)$$

$$\sqrt{((C_1 - D_1)^2 + (C_2 - D_2)^2 + ... + (C_n - D_n)^2)} \leq \sqrt{((C_j - E_j)^2)} \ (j \neq i)$$

$$d(C, D) \leq d(C, E) \ (j \neq i)$$

سوال سوم

الف) یک دسته شامل تمامی نقاط قرمز و دو نقطه آبی سمت چپ به مرکز اولین نقطه آبی چپ بوده و دسته دیگر ۴ نقطه آبی به مرکز وسط نقطه دوم و سوم از راست است. بدین شکل:



دلیل این پیشبینی، در نظر گرفتن نقاط مرکزی است به گونه ای که بتواند بیشترین پوشش را بدهد. دو نقطه آبی و زرد در نظر گرفته شده بیشترین پوشش را دارند به طوری که فقط دو دسته داشته باشیم. اجرای الگوریتم نیز این پیش بینی را تایید کرد.

ب) قطعا. روش DBSCAN که بر اساس Density-based Clustering عمل کرده اما به صورت Spatial، قادر است بین دو دسته نقاط قرمز و آبی تمایز قائل شده و هر کدام را یک دسته کند، که نتیجه مطلوب ما است. این الگوریتم بر خلاف K-Means به صرفا بر اساس فاصله Centroid ها عمل می‌کند، چگالی را اندازه گرفته که heuristic بهتری در مسئله مطرح شده است.

داده کاوی

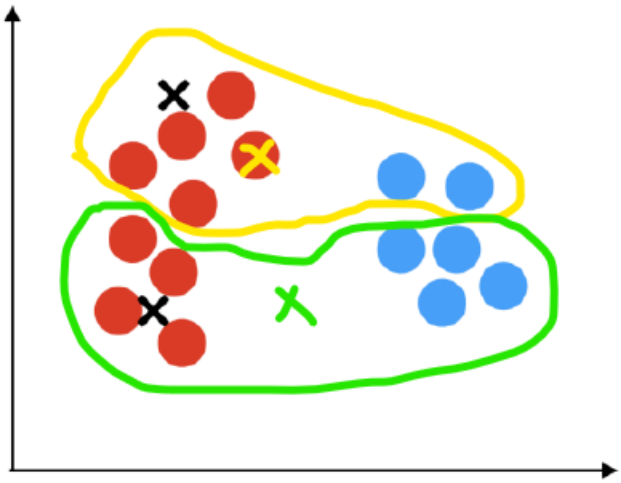
کیوان ایپچی حق – ۹۸۳۱۰۷۳

ج) در صورتی که مجموعه داده در بخش های مختلف چگالی های متنوع و بسیار گسترده ای داشته باشد الگوریتم Density-based ها نمی‌توانند دسته بندی درستی داشته باشند. یا زمانی که داده تعداد بعد زیادی داشته باشد (High-Dimensional Data). همچنین وجود برخی noise ها و outlier ها در داده می‌تواند کارکرد آن را دچار مشکل کند.

برای مثال مجموعه داده ای داریم که دو دسته دارد که مقدار خوبی noise و outlier داشته باشد. دسته اول چگالی بالایی داشته و نقاط آن نزدیک هم باشد و در دسته دوم نقاط پراکنده باشند. در این صورت الگوریتم density-based ها با چندین چالش تشخیص outlier ها، تعیین میزان density threshold و ترکیب cluster ها روبرو می‌شوند.

سوال چهارم

الف) با اجرای تخمینی الگوریتم با K=2 میبینیم دو دسته زرد و سبز بوجود می‌آیند زیرا تابع heuristic ما برای الگوریتم K-Means فاصله اقلیدوسی است (با اجرای الگوریتم در شبیه سازی پیشبینی تایید شد). دلیل بوجود آمدن centroid نهایی سبز رنگ، وسط بودن بین چهار نقطه آبی و چهار نقطه قرمز رنگ است و دلیل بوجود آمدن centroid نهایی زرد، پنج نقطه قرمز و چهار نقطه آبی است که منجر می‌شود مرکز به چپ متمایل شود. دلیل درست دسته بندی نکردن این الگوریتم انتخاب نقاط اولیه نامناسب است.



ب) مزایا و معایب:

- **استفاده از medoid به جای median**: از مزایا این روش می‌توان به کاهش تاثیر outlier ها اشاره کرد به طوری که medoid ها به outlier ها حساسیت پایین تری دارند. همچنین قابلیت فهم و استدلال بالاتری می‌دهد زیرا medoid ها در اصل نقاط واقعی ما هستند در صورتی که mean و median یک نقطه جدید خارج داده اصلی هستند. همچنین برای داده غیر عددی استفاده از medoid مناسب تر است.

از معایب آن، میتوان اشاره کرد این کار محاسبه medoid ها را بیشتر کرده و اصطلاحا computationally expensive است. در این حال محدودیتی بر روی dissimilarity measures داریم.

- **انتخاب نقاط اولیه به شکلی که بیشترین فاصله را از هم داشته باشند:** از مزایا آن دیرتر converge شدن centroid ها می‌توان اشاره کرد تا در نقاط بهینه محلی گیر نیفتیم. همچنین برای مجموعه داده هایی با توزیع نامتعارف انتخاب خوبی است.
از معایب آن: به outlier ها بشدت حساس خواهیم شد و محاسبات ما بیشتر میشود زیرا به تعداد iteration بیشتری برای converge شدن نیاز داریم. در نهایت هم برای دسته هایی که همپوشانی داشته باشند می‌تواند مشکل ساز باشد.
- **انتخاب نقاط اولیه بر اساس توزیع داده ها:** خوبی این روش به مقدار دهی اولیه بهتر برای دسته ها است به طوری که شروع خوبی داریم (لااقل!) و حساسیت به outlier ها کاهش میابد. از طرفی برای داده های با چگالی نامتعارف این روش بهتر جواب می‌دهد.
از طرفی میزان محاسبات به طور قابل توجهی افزایش میابد و مثل روش قبل دسته های با همپوشانی قابل توجه را نمیتوان به خوبی دسته بندی کرد.
- **انتخاب چند باره مراکز اولیه برای رسیدن به جواب مناسب:** بهترین روش است زیرا با افزایش تکرار این امر خطا کاهش میابد و outlier ها را کمرنگ می‌کنیم و مشکلات حالت های پیشین به طور قابل توجهی کاهش میابد.
از طرفی میزان محاسبات ما بیش تر از حالت های پیش افزایش پیدا کرده و باید روشی برای اتمام آن (چه وقتی converge را قبول می‌کنیم و یا threshold ما برای اینکار چیست؟) (مگر اینکه تعداد مشخص داشته باشیم برای انتخاب چند باره مراکز) داشته باشیم و این می‌تواند زمان اجرا را افزایش دهد.

د) بهتر است K را معادل ۲ و یا ۳ قرار دهیم. انتخاب ۳ بهتر است زیرا دسته های بیشتری داریم در ازای مقداری خطا و انتخاب ۲ شاید خیلی هم خوب نباشد (با اعداد و ارقام می‌توان بهتر گفت ^_^). معیار انتخاب این عدد طول خط های عمودی است که هرچه بیشتر باشد یعنی دسته ها از هم متمایز تر هستند (مثل عدد K در الگوریتم K-Means)

سوال پنجم

ماتریس اولیه:

$$initial\ matrix = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ -1 & -1 \\ -1 & -2 \\ -2 & -1 \end{bmatrix}$$

1. در قدم اول باید داده را center کنیم به صورتی که میانگین را از تمامی اعضا کم کرده و یک ماتریس جدید تشکیل دهیم:

$$mean_{column\ one}: \frac{1+1+2-1-1-2}{6} = 0$$

$$mean_{column\ two}: \frac{1+2+1-1-2-1}{6} = 0$$

داده کاوی

کیوان ایپچی حق – ۹۸۳۱۰۷۳

$$centered\ matrix = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ -1 & -1 \\ -1 & -2 \\ -2 & -1 \end{bmatrix}$$

2 . قدم دوم محاسبه ماتریس کوواریانس است (فرمول محاسبه آن: $\frac{matrix * matrix.T}{n-1}$)

$$covariance\ matrix = \begin{bmatrix} 2.4 & 2 \\ 2 & 2.4 \end{bmatrix}$$

3 . در قدم بعد باید مقدار Eigenvalues و Eigenvectors را محاسبه کنیم

(با حل رابطه مقابل: $CovarianceMatrix * EigenVectors = EigenValues * EigenVectors$)

$$EigenValues = [4.4, 0.4] \qquad EigenVector = \begin{bmatrix} 0.7 & -0.7 \\ 0.7 & 0.7 \end{bmatrix}$$

4 . در گام بعد باید مقادیر PC را با استفاده از Eigenvalues و Eigenvector بدست آوریم که معادل ستون اول میشود:

$$PC = [0.7, 0.7]$$

5 . سپس باید dot product برای ماتریس های PC و Centered Matrix را بدست آوریم که مطابق زیر است:

$$projected\ matrix = [\ 1.4\ \ 2.1\ \ 2.1, -1.4, -2.1, -2.1]$$

اعداد بالا با استفاده از اسکریپت پایتون که در کنار همین فایل ضمیمه شده نیز تست شده و اعداد یکسان در آمده!

سوال ششم

1. Choosing the item sets

Itemset-1 (minsup= 0.3):

Item	Frequency	Support
Bread	5	0.71
Milk	5	0.71
Diaper	6	0.85
Beer	4	0.57
Eggs	1	0.14
Coke	3	0.42

Itemset2 (minsup= 0.3):

Item	Frequency	Support
Bread, Milk	3	0.42
Bread, Diaper	4	0.57
Bread, Beer	3	0.42
Bread, Coke	1	0.14

Milk, Diaper	4	0.57
Milk, Beer	2	0.28
Milk, Coke	3	0.42
Diaper, Beer	4	0.57
Diaper, Coke	3	0.42
Beer, Coke	2	0.28

Itemset3 (minsup= 0.3):

Item	Frequency	Support
Bread, Milk, Diaper	2	0.28
Bread, Beer, Diaper	3	0.42
Milk, Diaper, Coke	3	0.42

بر اساس اجرای الگوریتم بالا برای یافتن itemset، خروجی نهایی سبز رنگ شده است.

2. Finding Associative Rules:

Item	Confidence
Milk -> Diaper	0.66
Milk -> Coke	0.6
Diaper -> Milk	0.6
Diaper -> Coke	0.5
Coke -> Milk	1
Coke -> Diaper	1
Milk, Diaper -> Coke	0.75
Coke -> Milk, Diaper	1
Milk, Coke -> Diaper	1
Diaper -> Milk, Coke	0.5
Diaper, Coke -> Milk	1
Milk -> Diaper, Coke	0.6
Bread -> Beer	0.6
Beer -> Bread	0.75
Bread -> Diaper	0.8
Diaper -> Bread	0.66
Beer -> Diaper	1
Diaper -> Beer	0.66
Bread, Beer -> Diaper	1
Diaper -> Beer, Bread	0.5
Bread, Diaper -> Beer	0.75
Beer -> Bread, Diaper	0.75
Beer, Diaper -> Bread	0.75
Bread -> Beer, Diaper	0.6

سوال هفتم

الف) مانند سوال قبلی عمل می‌کنیم.

Itemset-1 (Support=0.33):

داده کاوی

کیوان ایپچی حق – ۹۸۳۱۰۷۳

Item	Frequency	Support
A	5	0.62
B	5	0.62
C	5	0.62
D	3	0.37
E	2	0.25

Itemset-2 (Support=0.33):

Item	Frequency	Support
A, B	3	0.37
A, C	3	0.37
A, D	2	0.25
B, C	4	0.5
B, D	2	0.25

Itemset-3 (Support=0.33):

Item	Frequency	Support
A, B, C	3	0.37

ب)

Item	Confidence
A -> B	0.6
B -> A	0.6
B -> C	
C -> B	
A -> C	0.6
C -> A	0.6
A, B -> C	1
C -> A, B	0.6
A, C -> B	1
B -> A, C	0.6
B, C -> A	0.75
A -> B, C	0.6

بخش عملی

ماتریس ها شباهت:

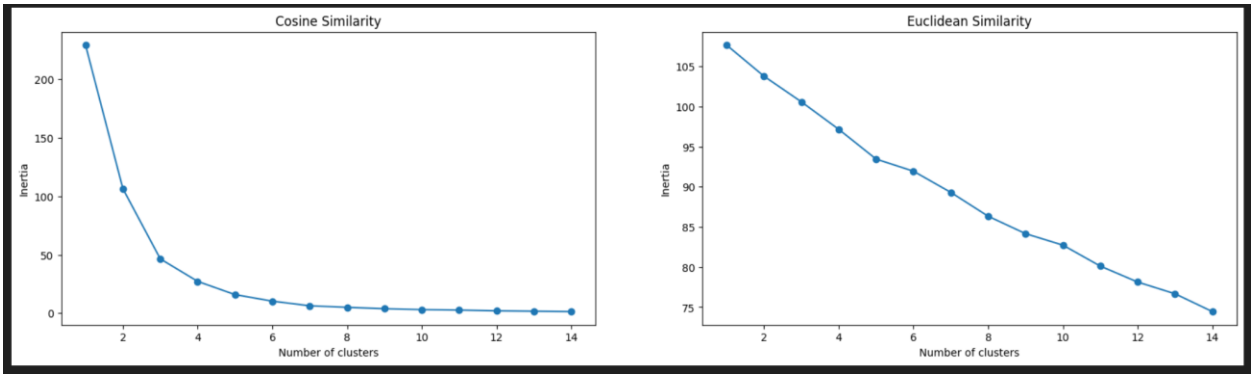
⚒ داده کاوی

کیوان ایپچی حق – ۹۸۳۱۰۷۳

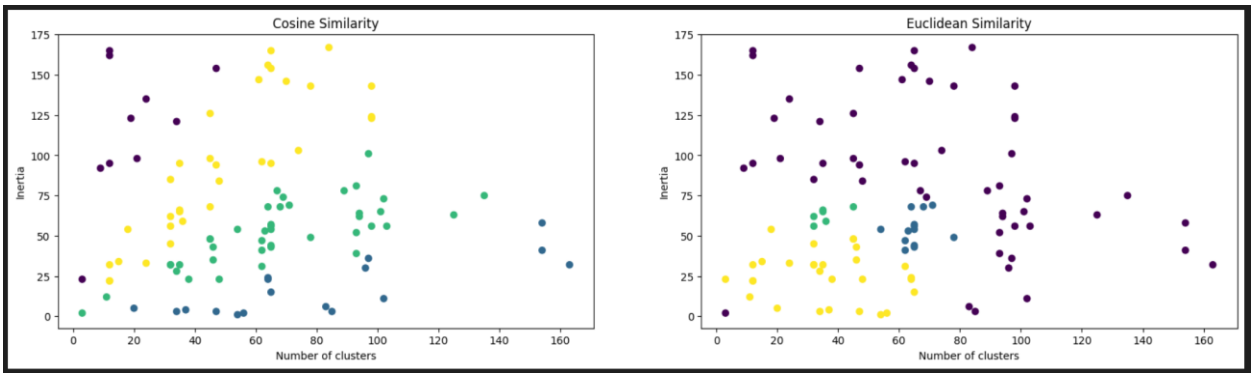
```
- Cosine Similarity Matrix:
[[1.         0.99985238 0.99604997 ... 0.94266984 0.72007789 0.68925746]
 [0.99985238 1.         0.99742859 ... 0.94826473 0.73189397 0.70160425]
 [0.99604997 0.99742859 1.         ... 0.96857935 0.77884743 0.75086792]
 ...
 [0.94266984 0.94826473 0.96857935 ... 1.         0.91036648 0.89153278]
 [0.72007789 0.73189397 0.77884743 ... 0.91036648 1.         0.99905616]
 [0.68925746 0.70160425 0.75086792 ... 0.89153278 0.99905616 1.        ]],


- Euclidean Similarity Matrix:
[[1.         0.30901699 0.01998777 ... 0.03834375 0.02563238 0.02203387]
 [0.30901699 1.         0.02088896 ... 0.04096782 0.02536864 0.02123245]
 [0.01998777 0.02088896 1.         ... 0.02429415 0.01349853 0.01086892]
 ...
 [0.03834375 0.04096782 0.02429415 ... 1.         0.02942488 0.01806962]
 [0.02563238 0.02536864 0.01349853 ... 0.02942488 1.         0.03333333]
 [0.02203387 0.02123245 0.01086892 ... 0.01806962 0.03333333 1.        ]]
```

نمودار Elbow:



دسته بندی:



همچنین تصاویر در نوت بوک موجود هستند.