دانشگاه صنعتی امیرکبیر
(پلیتکنیک تهران)
دانشکده مهندسی کامپیوتر

درس داده‌کاوی

تمرین سوم۳

امیرمهدی زرین‌نژاد

۹۷۳۱۰۸۷

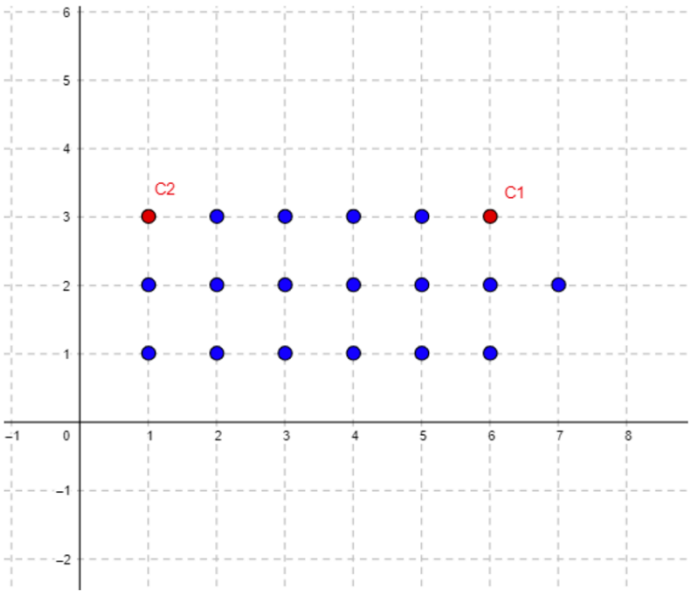
نیم‌سال دوم ۱۴۰۰

سوالات تشریحی

سوال اول)

الگوریتم kmeans را روی داده‌های شکل زیر با شرح مراحل اجرا کنید و خوشه‌ها را تعیین کنید.

- از نرم یک به عنوان معیار فاصله استفاده کنید.
- تعداد خوشه‌ها را ۲ در نظر بگیرید و c_1 و c_2 مراکز ابتدایی‌اند.



نرم۱ یعنی فاصله منهتن ==> $d(i, j) = |xi1 - xj1| + |xi2 - xj2| + ... + |xip - xjp|$

سنترویدها را داریم. پس ابتدا تعیین می‌کنیم که هر داده به کدام خوشه تعلق می‌گیرد. به این صورت که فاصله هر داده را با سنترویدها محاسبه می‌کنیم و داده را در خوشه با نزدیک‌ترین سنترویدها به آن داده قرار می‌دهیم:

داده‌ها:

```
data1:(1,1), data2:(2,1), data3:(3,1), data4:(4,1), data5:(5,1), data6:(6,1),

data7:(1,2), data8:(2,2), data9:(3,2), data10:(4,2), data11:(5,2), data12:(6,2),
data13:(7,2),

data14=C2:(1,3), data15:(2,3), data16:(3,3), data17:(4,3), data18:(5,3),
data19=C1:(6,3)
```

فاصله هر داده از ۲ سنترواید را محاسبه می‌کنیم:

data14=C2:(1,3), data19=C1:(6,3)

داده/مرکز	data1	data2	data3	data4	data5	data6
C1 (6,3)	6-1+3-1 =7	6-2+3-1 =6	6-3+3-1 =5	6-4+3-1 =4	6-5+3-1 =3	6-6+3-1 =2
C2 (1,3)	1-1+3-1 =2	2-1+3-1 =3	3-1+3-1 =4	4-1+3-1 =5	5-1+3-1 =6	6-1+3-1 =7

داده/مرکز	data7	data8	data9	data10	data11	data12	data13
C1 (6,3)	6-1+3-2 =6	6-2+3-2 =5	6-3+3-2 =4	6-4+3-2 =3	6-5+3-2 =2	6-6+3-2 =1	7-6+3-2 =2
C2 (1,3)	1-1+3-2 =1	2-1+3-2 =2	3-1+3-2 =3	4-1+3-2 =4	5-1+3-2 =5	6-1+3-2 =6	7-1+3-2 =7

داده/مرکز	data14	data15	data16	data17	data18	data19
C1 (6,3)	6-1+3-3 =5	6-2+3-3 =4	6-3+3-3 =3	6-4+3-3 =2	6-5+3-3 =1	6-6+3-3 =0
C2 (1,3)	1-1+3-3 =0	2-1+3-3 =1	3-1+3-3 =2	4-1+3-3 =3	5-1+3-3 =4	6-1+3-3 =5

با مقایسه فاصله‌های منهتن از سنترویدها هر داده را به نزدیک‌ترین سنترواید نسبت می‌دهیم که در جدول با رنگ سبز و نارنجی نشان داده شده‌اند. (سبزها به C1 و نارنجی‌ها به C2 تعلق گرفتند)

حال باید مراکز را آپدیت کنیم. مرکز جدید هر خوشه برابر است با میانگین داده‌های آن خوشه:

C1new: $x = \frac{4+5+6+4+5+6+7+4+5+6}{10} = \frac{52}{10} = 5.2,\ \ y = \frac{1+1+1+2+2+2+2+3+3+3}{10} = \frac{20}{10} = 2$ ➔ (5.2, 2)

C2new: s $x = \frac{1+2+3+1+2+3+1+2+3}{9} = \frac{18}{9} = 2,\ \ y = \frac{1+1+1+2+2+2+3+3+3}{9} = \frac{18}{9} = 2$ ➔ (2, 2)

حال دوباره باتوجه به مراکز جدید و فاصله داده‌ها از آن‌ها، تعلق داده‌ها را به خوشه‌ها بررسی می‌کنیم:

داده/مرکز	data1	data2	data3	data4	data5	data6
C1 (5.2,2)	5.2-1+2-1 =5.2	5.2-2+2-1 =4.2	5.2-3+2-1 =3.2	5.2-4+2-1 =2.2	5.2-5+2-1 =1.2	6-5.2+2-1 =1.8
C2 (2,2)	2-1+2-1 =2	2-2+2-1 =1	3-2+2-1 =2	4-2+2-1 =3	5-2+2-1 =4	6-2+2-1 =5

داده/مرکز	data7	data8	data9	data10	data11	data12	data13
C1 (5.2,2)	5.2-1+2-2 =4.2	5.2-2+2-2 =3.2	5.2-3+2-2 =2.2	5.2-4+2-2 =1.2	5.2-5+2-2 =0.2	6-5.2+2-2 =0.8	7-5.2+2-2 =1.8
C2 (2,2)	2-1+2-2 =1	2-2+2-2 =0	3-2+2-2 =1	4-2+2-2 =2	5-2+2-2 =3	6-2+2-2 =4	7-2+2-2 =5

داده/مرکز	data14	data15	data16	data17	data18	data19
C1 (5.2,2)	5.2-1+3-2 =5.2	5.2-2+3-2 =4.2	5.2-3+3-2 =3.2	5.2-4+3-2 =2.2	5.2-5+3-2 =1.2	6-5.2+3-2 =1.8
C2 (2,2)	2-1+3-2 =2	2-2+3-2 =1	3-2+3-2 =2	4-2+3-2 =3	5-2+3-2 =4	6-2+3-2 =5

همانطور که مشخص است با تغییر در مراکز، تعلق داده‌ها به هرخوشه تغییر نکرده است.

حال دوباره مراکز را آپدیت میکنیم:

C1new: $x = \frac{4+5+6+4+5+6+7+4+5+6}{10} = \frac{52}{10} = 5.2,\ y = \frac{1+1+1+2+2+2+2+3+3+3}{10} = \frac{20}{10} = 2$ ➔ $(5.2, 2)$

C2new: s $x = \frac{1+2+3+1+2+3+1+2+3}{9} = \frac{18}{9} = 2,\ y = \frac{1+1+1+2+2+2+3+3+3}{9} = \frac{18}{9} = 2$ ➔ $(2, 2)$

می‌بینیم باتوجه به اینکه داده‌های هر خوشه تغییری نکرده است، مراکز هم تغییر نکردند.

پس الگوریتم همینجا به پایان می‌رسد و سنترویدها و داده‌های هر خوشه به شرح زیر هستند:

C1: (5.2, 2)

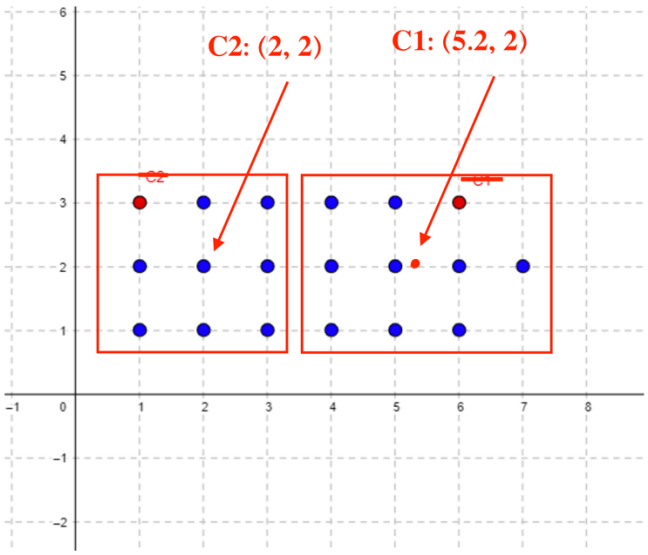
Cluster1: data4, data5, data6, data10, data11, data12, data13, data17, data18, data19

➔ Cluster1: (4, 1), (5, 1), (6, 1), (4, 2), (5, 2), (6, 2), (7, 2), (4, 3), (5, 3), (6, 3)

C2: (2, 2)

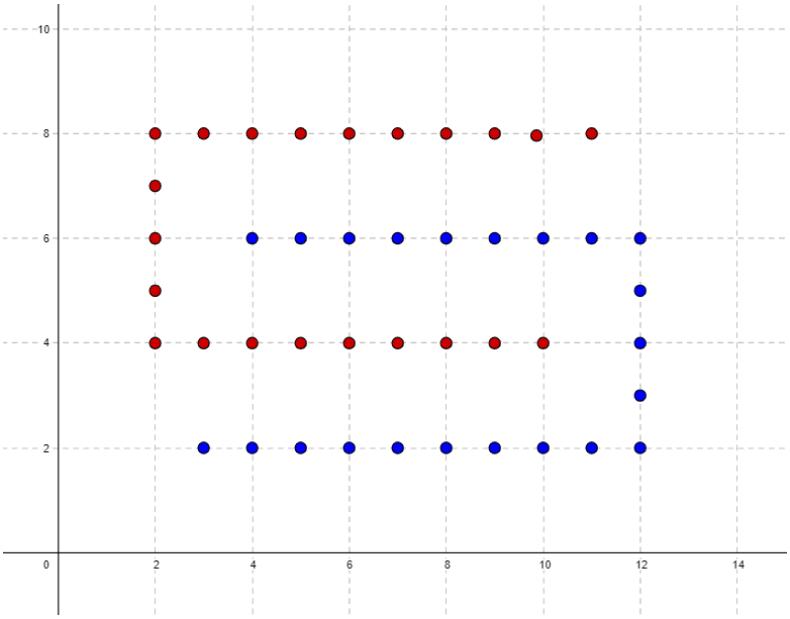
Cluster2: data1, data2, data3, data7, data8, data9, data14, data15, data16

➔ Cluster1: (1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)



سوال دوم)

یک الگوریتم مناسب برای خوشه‌بندی مجموعه داده‌ی زیر پیشنهاد دهید و توضیح دهید که به چه دلیل آن را انتخاب کردید. سپس با تعیین پارامترهای آن، عملکرد این الگوریتم را روی این داده‌ها تحلیل کنید.



الف) یک الگوریتم مناسب برای خوشه‌بندی این داده‌ها می‌تواند الگوریتم‌های مبتنی بر چگالی DBSCAN باشد. الگوریتم‌های خوشه‌بندی مانند k-means باتوجه به این که برحسب فاصله داده‌ها از یکدیگر و مراکز خوشه‌بندی را انجام می‌دهند؛ نمی‌توانند به درستی عمل کنند. زیرا همان‌طور که از توزیع داده‌ها و دسته‌ها برمی‌آید، صرف فاصله معیار مناسبی برای خوشه‌بندی نیست و داده‌هایی که در یک میانگین فاصله‌ای قرار می‌گیرند لزوما ارتباط بیش‌تری ندارند و در یک دسته قرار نگرفته‌اند. (و اینکه خوشه‌ها را نمی‌توان با دایره یا کره یا بیضی یا ... جداسازی کرد و این کاری است که عموما در kmeans و معیار فاصله صورت می‌گیرد. چراکه فاصله درحالت ساده‌اش عموما به صورت شعاعی اطراف داده است)

اما الگوریتم‌های مبتنی بر چگالی می‌توانند بهتر عمل کنند زیرا برحسب چگالی و میزان تراکم داده‌ها عمل می‌کنند و نه فاصله خام یا ... که این باعث می‌شود بتوانند شکل خوشه‌های مختلفی را تشخیص و تشکیل دهند. مانند همین نمونه داده که داده‌های مرتبط و مربوط به یک خوشه، تراکم بیش‌تری دارند و چگال‌تر هستند.

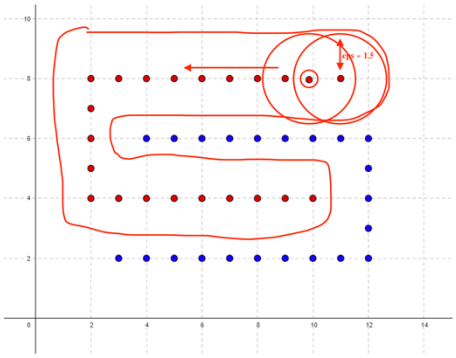
این الگوریتم ۲ ابرپارامتر epsilon(شعاع همسایگی) و minpoints(حداقل تعداد داده که باید در شعاع همسایگی وجود داشته باشد تا بتوان کلاستر را از آن داده شروع کرد و یا core محسوب شود) دارد. Core داده‌ای است که در فاصله شعاع همسایگی‌اش حداقل به تعداد minpoints داده وجود داشته باشد و می‌توان خوشه را از آن شروع کرد. الگوریتم مبتنی بر چگالی از یک نقطه به صورت رندم شروع می‌کند و اگر core بود باتوجه به شعاع

همسایگی و نقاط موجود در شعاع، نقطه‌های همسایه را پیدا می‌کند و با پیدا کردن coreهای بعدی از میان همسایگان گسترش می‌یابد و coreهای بعدی را پیدا می‌کند. (داده‌هایی که core نیستند اما در همسایگی core هستند و از طریق core می‌توان به آن‌ها دست یافت را border می‌گویند و داده‌هایی که نه core هستند و نه از طریق core می‌توان به آن‌ها دست یافت نیز outlier هستند)

این روش پیداکردن core و رشد دادن یک خوشه باعث می‌شود که الگوریتم بتواند مسیری که داده‌های یک دسته درآن قرار دارند را به خوبی تشخیص دهد و دنبال کند و نهایتا خوشه را به همین شکلی که هست گسترش دهد و ایجاد کند.

هم‌چنین این الگوریتم ویژگی‌های مثبت دیگری دارد که آن را به انتخاب مناسبی تبدیل می‌کند. از جمله اینکه نواحی چگال و متراکم را به خوبی تشخیص می‌دهد و از نواحی غیرچگال مشخص می‌کند. در این الگوریتم نیازی نیست که تعداد خوشه‌ها را از قبل داشته باشیم. این الگوریتم به داده‌های پرت و یا نویز حساسیت کمی دارد و کمتر عملکردش تحت تاثیر این موارد قرار می‌گیرد و عملکرد بهتری نشان می‌دهد.

ب) دراین توزیع داده شده، داده‌های مجاور در یک خوشه در فاصله تقریبا ۱واحد از یکدیگر قرار گرفته‌اند و حداقل فاصله بین دو داده مجاور از خوشه‌های متفاوت برابر ۲ است. پس می‌توان eps را برابر ۱.۵ و minspoint را ۱ یا ۲ درنظر گرفت و به اینصورت خوشه‌بندی را انجام داد و داده‌های در فاصله ۱.۵ را به عنوان همسایه برای گسترش خوشه درنظر گرفت. این مقادیر در تفکیک خوشه‌ها نیز به خوبی عمل می‌کند زیرا داده‌های مجاور از دو خوشه متفاوت حداقل به اندازه ۲واحد از یکدیگر فاصله دارند. پس در شعاع همسایگی core قرار نمی‌گیرند(۲ از eps بیش‌تر است و در همسایگی core خوشه دیگر تشخیص داده‌نمی‌شود) و به این صورت داده‌های خوشه‌های مختلف به خوبی از یکدیگر جدا می‌شوند.



برای مثال در این تصویر میبینم که روند به چه صورت است. از یک نقطه رندم مانند داده قرمز در سمت راست شروع کردیم و در شعاع همسایگی‌اش یک داده دیگر وجود دارد(پس core است و خوشه شروع شده است) و به همین صورت برای همسایگانش هم انجام می‌دهیم و خوشه گسترش می‌یابد. هم‌چنین می‌بینیم باتوجه به اینکه شعاع همسایگی به درستی انتخاب شده، داده‌های آبی در خوشه داده‌های قرمز قرار نمی‌گیرند زیرا فاصله حداقل ۲واحد دارند که از شعاع همسایگی بیش‌تر است.

این روند برای داده‌های آبی هم متقابلا انجام می‌شود و در یک خوشه قرار می‌گیرند.

سوال ۳)

توضیح دهید که چگونه با ذخیره‌ی closed frequent itemsets می‌توان پشتیبانی مربوط به frequent itemset ها را تعیین کرد.

یک itemset در یک مجموعه داده closed است اگر؛ super-itemset از آن itemset وجود نداشته باشد که دارای تعداد پشتیبانی مشابه با آن itemset باشد. Closed frequent itemset یک frequent itemset است که هم closed است و هم support آن بزرگتر یا مساوی minsup است.

باتوجه به ویژگی‌های ذکر شده می‌توان به support یک frequent itemset از طریق closed frequent itemsetها دست پیدا کرد. به این صورت که اگر همه closed frequent itemsetها را ذخیره داشته باشیم و بخواهیم support یک آیتم F را پیدا کنیم، support آیتم F برابر می‌شود با support کوچک‌ترین super-itemsetای که F را در خود دارد و یک closed frequent itemset است. یعنی بین closed frequent itemsetها باید کوچک‌ترین super-itemsetای که F را درخود دارد پیدا کنیم و ساپورت F مساوی می‌شود با ساپورت این closed frequent itemset.

سوال ۴)

الگوریتم apriori را برروی تراکنش‌های زیر اجرا کنید. تمامی مراحل تولید مجموعه آیتم‌های کاندید را نشان دهید و درنهایت مجموعه آیتم‌های پرتکرار را بدست آوردید. همچنین تمامی قواعد انجمنی قابل تولید از مجموعه آیتم‌ها را نوشته، آنهایی که مطمئن هستند را مشخص کرده و براساس میزان اطمینان مرتب کنید (آستانه پشتیبانی را ۳۳٪ و آستانه اطمینان را ۶۰٪ در نظر بگیرید).

Minsup = 0.33 ➔ $\frac{\text{count(تعداد)}}{\#\text{transactions}} \geq \frac{1}{3}$ (0.33) باید باشه

#transactions = 6 ➔ support count = 2 ➔ count $\geq$ 2(support count) باید باشه و می‌دانیم

شماره تراکنش	آیتم‌ها
۱	سیب، پرتقال، موز
۲	انار، موز
۳	سیب، پرتقال، موز
۴	انار، پرتقال
۵	سیب، نارنگی
۶	سیب، نارنگی، انار

تعداد	آیتم-تکی
۴	سیب
۳	پرتقال
۳	موز
۳	انار
۲	نارنگی

چون همگی تعداد(supcount) بیش‌تر مساوی ۲ دارند؛ minsup را برآورده می‌کنند و برای آیتم‌ست‌های دوتایی و بیش‌تر می‌توانند مورد استفاده قرار گیرند. درواقع برابر supcount(۲) یا بیش‌تر از آن هستند است و supportشان بزرگ‌تر مساوی 2/6 (support = #itemset-count / #transactions) است که از 0.33 (minsup) بیش‌ترمساوی می‌شود.

تعداد	آیتم‌ست-۲تایی
۲	سیب، پرتقال
۲	سیب، موز
۲	پرتقال، موز
۱	انار، موز
۱	انار، پرتقال
۲	سیب، نارنگی
۱	سیب، انار
۱	نارنگی، انار

باتوجه به اینکه ۱ از ۲ کمتر است؛ آیتم‌ست‌های {انار، موز}، {انار،پرتقال}، {سیب، انار} و {نارنگی، انار} supcount و نتیجتا minsup را برآورده نمی‌کنند. درواقع countشان برابر۱ و supportشان برابر 1/6 است (support = #itemset-count / #transactions) که از 0.33 (minsup) کمتر می‌شود. پس این آیتم‌ست‌ها را کنار می‌گذاریم و برای مراحل بعد استفاده نمی‌کنیم. (این‌ها هرس می‌شوند. اما بقیه (2/6)=support بزرگ‌تر مساوی 0.33 دارند که minsup را برآورده می‌کنند)

تعداد	آیتم‌ست-۳تایی
۲	سیب، پرتقال، موز

آیتم‌ست ۳تایی ممکن در این مرحله {سیب، پرتقال، موز} است که تعدادش ۲ است و supcount و نتیجتا minsup را برآورده می‌کنند و support=2/6 از 0.33 بزرگ‌تر مساوی می‌شود.
هم‌چنین باتوجه به این که آیتم‌ست بزرگ‌تری در تراکنش‌ها وجود ندارد، همینجا تشکیل آیتم‌ست‌ها را پایان می‌دهیم و قوانین انجمنی را استخراج می‌کنیم.

{سیب، پرتقال، موز}

⬅ قوانین انجمنی ممکن:

۱- {پرتقال، موز} <== {سیب}

$$\text{Confidence} = \frac{\#\{\text{سیب ،پرتقال، موز}\}}{\#\{\text{سیب}\}} = \frac{2}{4} = 0.5$$

۲- {سیب، موز} <== {پرتقال}

$$\text{Confidence} = \frac{\#\{\text{سیب ،پرتقال، موز}\}}{\#\{\text{پرتقال}\}} = \frac{2}{3} = 0.66$$

۳- {سیب، پرتقال} <== {موز}

$$\text{Confidence} = \frac{\#\{\text{سیب ،پرتقال، موز}\}}{\#\{\text{موز}\}} = \frac{2}{3} = 0.66$$

۴- {موز} <== {سیب، پرتقال}

$$\text{Confidence} = \frac{\#\{\text{سیب ،پرتقال، موز}\}}{\#\{\text{سیب، پرتقال}\}} = \frac{2}{2} = 1.00$$

۵- {پرتقال} <== {سیب، موز}

$$\text{Confidence} = \frac{\#\{\text{سیب ،پرتقال، موز}\}}{\#\{\text{سیب، موز}\}} = \frac{2}{2} = 1.00$$

۶- {سیب} <== {پرتقال، موز}

$$\text{Confidence} = \frac{\#\{\text{سیب ،پرتقال، موز}\}}{\#\{\text{سیب، پرتقال}\}} = \frac{2}{2} = 1.00$$

باتوجه به اینکه آستانه اطمینان 0.60 است؛ قانون انجمنی اول اطمینانش آستانه را برآورده نمی‌کند و به عنوان قانون پذیرفته نمی‌شود زیرا مقدارش(0.50) از آستانه(0.60) کم‌تر است. اما بقیه اطمینان کافی را دارند و از آستانه بیش‌تر مساوی هستند و قوانین انجمنی ما را تشکیل می‌دهند. (زیرا مقادیر 0.66 و 1 دارند که از آستانه اطمینان بیش‌تر هستند)

🡸 قوانین انجمنی:

{سیب، موز} <== {پرتقال}

{سیب، پرتقال} <== {موز}

{موز} <== {سیب، پرتقال}

{پرتقال} <== {سیب، موز}

{سیب} <== {پرتقال، موز}

سوال ۵)

از ماتریس فاصله در جدول زیر برای انجام خوشه‌بندی سلسله مراتبی با لینک تک و کامل (min, max) استفاده کنید. نتایج خود را با کشیدن یک دندروگرام نشان دهید. در رسم باید به روشنی ترتیب ادغام نقاط نشان داده شود.

	P1	P2	P3	P4	P5
P1	0.00	0.10	0.41	0.55	0.35
P2	0.10	0.00	0.64	0.47	0.98
P3	0.41	0.64	0.00	0.44	0.85
P4	0.55	0.47	0.44	0.00	0.76
P5	0.35	0.98	0.85	0.76	0.00

لینک تک: single link ==> کوتاه‌ترین فاصله را درنظر می‌گیریم(min)

ابتدا p1 و p2 انتخاب می‌شوند که کوتاه‌ترین فاصله را دارند:

باهم دریک خانه قرار می‌گیرند و فاصله ترکیب‌شان از دیگر خانه‌ها با minimumگیری آپدیت می‌شود:

d(P1P2, P3) = min(d(P1, P3), d(P2, P3)) = min (0.41, 0.64)

d(P1P2, P4) = min(d(P1, P4), d(P2, P4)) = min (0.55, 0.47)

d(P1P2, P5) = min(d(P1, P5), d(P2, P5)) = min (0.35, 0.98)

	P1P2	P3	P4	P5
P1P2	0	0.41	0.47	0.35
P3	0.41	0	0.44	0.85
P4	0.47	0.44	0	0.76
P5	0.35	0.85	0.76	0

حال P1P2 با P5 کم‌ترین فاصله را دارند و برای ادغام انتخاب می‌شوند:

d(P1P2P5, P3) = min(d(P1P2, P3), d(P5, P3)) = min (0.41, 0.85)

d(P1P2P5, P4) = min(d(P1P2, P4), d(P5, P4)) = min (0.47, 0.76)

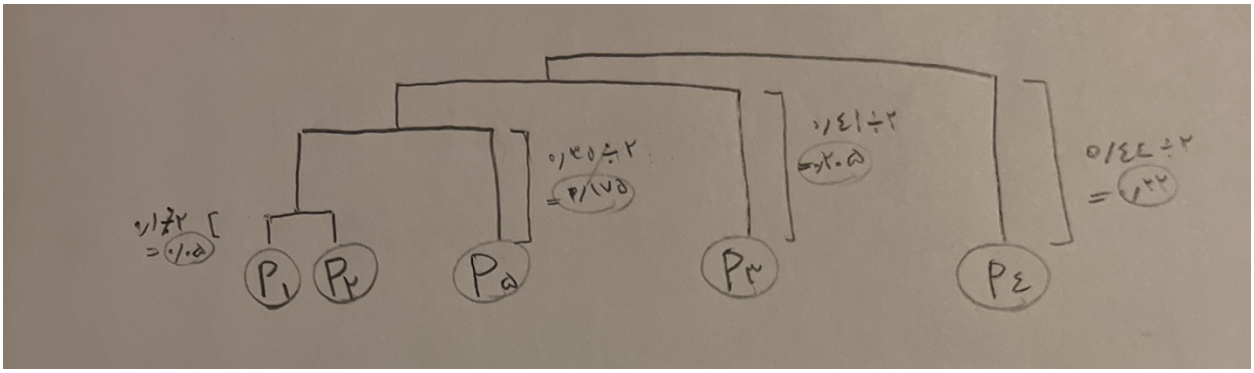
	P1P2P5	P3	P4
P1P2P5	0	0.41	0.47
P3	0.41	0	0.44
P4	0.47	0.44	0

حال P1P2P5 با P3 کم‌ترین فاصله را دارند و برای ادغام انتخاب می‌شوند:

d(P1P2P5P3, P4) = min(d(P1P2P5, P4), d(P3, P4)) = min (0.47, 0.44)

	P1P2P5P3	P4
P1P2P5P3	0	0.44
P4	0.44	0

Dendogram:



لینک کامل: complete link ==> بلندترین فاصله را برای آپدیت کردن فاصله تا خانه‌های ادغام شده درنظر می‌گیریم(max)

	P1	P2	P3	P4	P5
P1	0.00	0.10	0.41	0.55	0.35
P2	0.10	0.00	0.64	0.47	0.98
P3	0.41	0.64	0.00	0.44	0.85
P4	0.55	0.47	0.44	0.00	0.76
P5	0.35	0.98	0.85	0.76	0.00

ابتدا p1 و p2 انتخاب می‌شوند که کوتاه‌ترین فاصله را دارند:

باهم دریک خانه قرار می‌گیرند و فاصله ترکیب‌شان از دیگر خانه‌ها با minimumگیری آپدیت می‌شود:

d(P1P2, P3) = max(d(P1, P3), d(P2, P3)) = max (0.41, 0.64)

d(P1P2, P4) = max(d(P1, P4), d(P2, P4)) = max (0.55, 0.47)

d(P1P2, P5) = max(d(P1, P5), d(P2, P5)) = max (0.35, 0.98)

	P1P2	P3	P4	P5
P1P2	0	0.64	0.55	0.98
P3	0.64	0	0.44	0.85
P4	0.55	0.44	0	0.76
P5	0.98	0.85	0.76	0

حال P3 با P4 کم‌ترین فاصله را دارند و برای ادغام انتخاب می‌شوند:

d(P3P4, P1P2) = max(d(P3, P1P2), d(P4, P1P2)) = max (0.64, 0.55)

d(P3P4, P5) = max(d(P3, P5), d(P4, P5)) = max (0.85, 0.76)

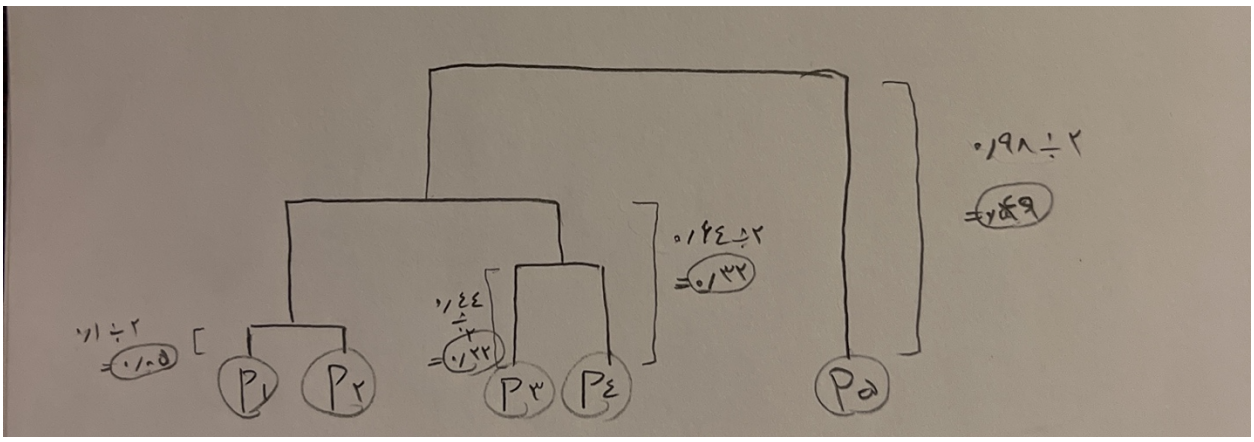
	P1P2	P3P4	P5
P1P2	0	0.64	0.98
P3P4	0.64	0	0.85
P5	0.98	0.85	0

حال P1P2 با P3P4 کم‌ترین فاصله را دارند و برای ادغام انتخاب می‌شوند:

d(P1P2P3P4, P5) = max(d(P1P2, P5), d(P3P4, P5)) = max (0.98, 0.85)

	P1P2P3P4	P5
P1P2P3P4	0	0.98
P5	0.98	0

Dendogram:



بخش برنامه‌نویسی

در google colab و با استفاده از jupyter notebook پیاده‌سازی انجام شده که فایل‌ها ضمیمه شده‌اند.