به نام خدا

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پاسخ تمرین سری چهارم داده کاوی

نیمسال اول ۰۰-۹۹

سوال ۱-

الف) در این حالت استفاده از PCA پیشنهاد می شود زیرا داده های زیادی برای هر کلاس نداریم ، بنابراین در این حالت LDA براساس ماتریس های کوواریانس درون کلاس[۱] قابل اعتماد نیست. درختان تصمیم برای ویژگی های اعداد حقیقی چندان مناسب نیستند ، زیرا سوالات تک متغیره ممکن است ویژگی هایی را که در ترکیب با سایر ویژگی ها سودمند هستند را نادیده بگیرد.

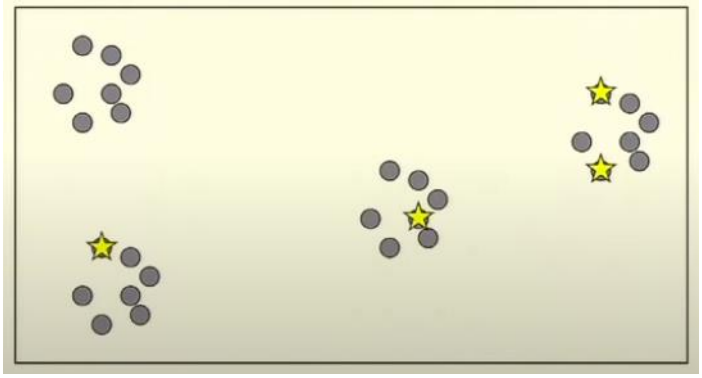
ب) ابتدا از PCA یا LDA برای کاهش ابعاد subvector با مولفه های عددی استفاده شود (به علت آنکه ابعاد بردار بزرگ است) ، و سپس از درختان تصمیم با این مولفه ها استفاده شود تا یاد گرفته شود که کدام یک از عناصر categorical در ترکیب با ویژگی های عددی کاهش یافته، مفید هستند.

ج) برای کاهش ابعاد از PCA بر روی داده های بدون برچسب استفاده شود و سپس از داده های دارای برچسب برای انتخاب اندازه مناسب بردار ویژگی استفاده شود یا با استفاده از LDA به کاهش ابعاد بیشتر پرداخته شود.

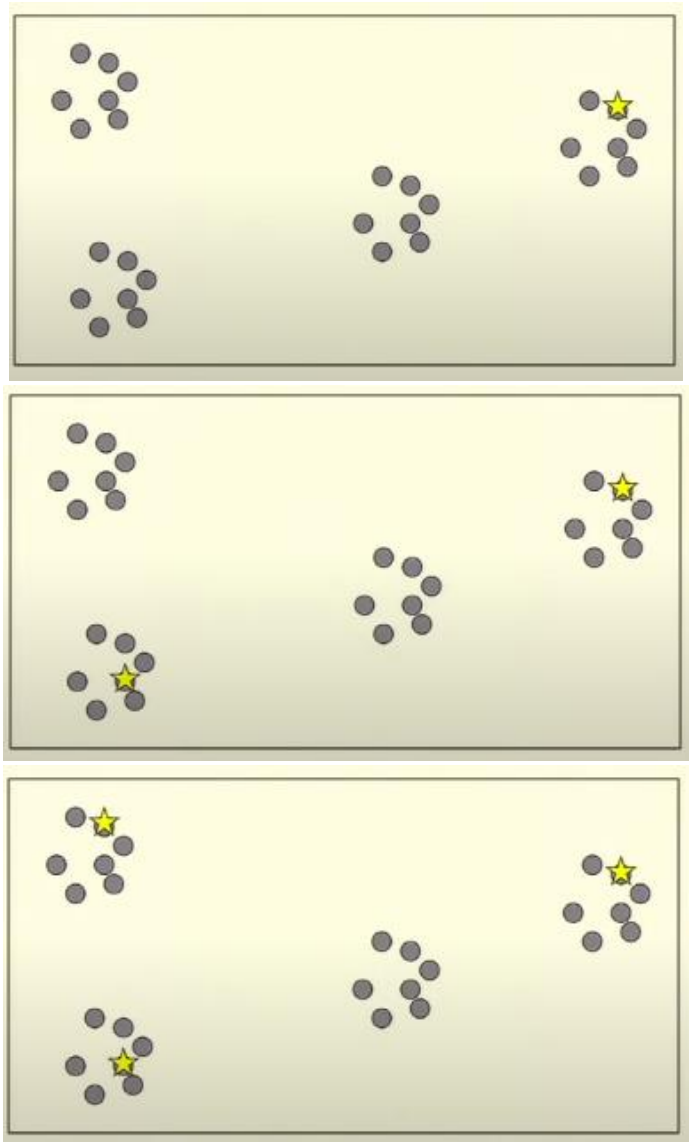
سوال ۲-

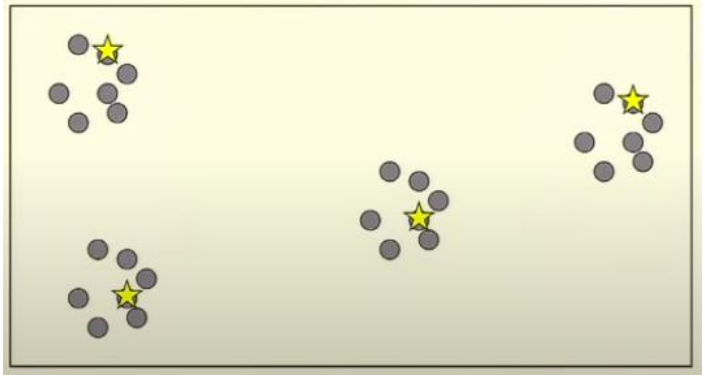
تفاوت اصلی الگوریتم kmeans++ با kmeans در مرحله انتخاب مراکز اولیه است. در الگوریتم kmeans تمامی k مرکز به صورت تصادفی انتخاب می‌شوند. اما در الگوریتم kmeans++ مراکز به صورت مرحله به مرحله و با توجه به فاصله‌ی نقاط از مراکز انتخاب‌شده تا آن مرحله انتخاب می‌شوند. در اولین گام الگوریتم kmeans++ یک مرکز به صورت تصادفی از بین داده‌ها انتخاب می‌شود. سپس فاصله تمامی نقاط تا این مرکز محاسبه می‌شود. احتمال انتخاب هر داده به عنوان مرکز در مرحله بعد با مربع فاصله آن نقطه تا مرکز مرحله قبلی رابطه مستقیم دارد. بعد از انتخاب مرکز دوم دوباره برای هر داده نزدیک‌ترین فاصله‌اش با یکی از مرکزها را مشخص می‌کنیم و دوباره با همان معیار احتمالی مرکز بعدی را انتخاب می‌کنیم و این کار را تا انتخاب شدن k مرکز ادامه می‌دهیم. در این حالت مراکز انتخاب‌شده به صورت پخش‌شده خواهند بود و کیفیت و سرعت همگرایی خوشه‌بندی بهتر خواهد شد.

[۱] within-class covariance matrices



مثال بالا یک حالت احتمالی است که در آن برای تعیین مراکز اولیه از الگوریتم kmeans ساده استفاده شده است که به صورت رندوم نقاطی به عنوان مرکز انتخاب شده. همانطور که مشخص است در این حالت الگوریتم به اشتباه دو خوشه سمت چپ را به عنوان یک خوشه در نظر می‌گیرد و نتیجه مورد نظر به دست نمی‌آید.





مراحل بالا حالتی است که مراکز اولیه با الگوریتم kmeans++ تعیین شده است. همانگونه که مشخص است الگوریتم kmeans++ در این حالت هم به کیفیت دسته‌بندی و هم به همگراشدن سریع‌تر کمک می‌کند.

برای جزئیات بیشتر می‌توانید به مقاله زیر مراجعه کنید.

“K-means++: The advantages of careful seeding” by D. Arthur and S. Vassilvitskii.

سوال ۳-

الف) صحیح. شعاعی که با عنوان eps معرفی میشود فاصله از نقطه core یا هسته را نشان میدهد و تمام همسایه های هسته باید در این فاصله یا کمتر باشند. همچنین ۲ نقطه همسایه محسوب میشوند اگر از طریق یک زنجیره از نقاطی که ۲ به ۲ همسایه هستند به یکدیگر برسند. در نتیجه تعداد زیادی نقطه هسته (core) در یک خوشه قرار دارد و هر نقطه در آن خوشه باید در فاصله کمتر یا مساوی eps از یک نقطه هسته باشد.

ب) غلط. پیچیدگی زمانی الگوریتم DBSCAN در بدترین حالت $O(n^2)$ (برای چک کردن همسایه های مجاور با فاصله eps) و در حالت بهینه O(nlogn) میباشد.

ج) صحیح. از ویژگی های مهم الگوریتم DBSCAN مقاومت آن در برابر داده ها پرت یا outlier ها می-باشد. در حقیقت نقاط outlier معمولا دارای تعداد کمی نقطه مجاور خود در فاصله eps یا کمتر می-باشد و بنا بر مقادیر minPts و eps می-توان حساسیت الگوریتم را نسبت به داده-های پرت تغییر داد. به زبان ساده تر الگوریتم DBSCAN به دنبال تراکم داده ها میگردد و نقاطه که متراکم نیستند را جزو خوشه بندی محسوب نمی-کند.

د) صحیح. تعداد خوشه ها جزو پارامتر های ورودی این الگوریتم نمی-باشد و این تعداد با تغییر پارامتر های minPts و eps تغییر می-کند.

سوال ۴-

4.

-Single link HAC dendrogram

	A	B	C	D	E	F
A	0					
B	0.12	0				
C	0.51	0.25	0			
D	0.84	0.16	0.14	0		
E	0.28	0.77	0.70	0.45	0	
F	0.34	0.61	0.93	0.20	0.67	0

Distance matrix 0

	A,B	C	D	E	F
A,B	0				
C	0.25	0			
D	0.16	0.14	0		
E	0.28	0.70	0.45	0	
F	0.34	0.93	0.20	0.67	0

Distance matrix 1

	A,B	C,D	E	F
A,B	0			
C,D	0.16	0		
E	0.28	0.45	0	
F	0.34	0.20	0.67	0

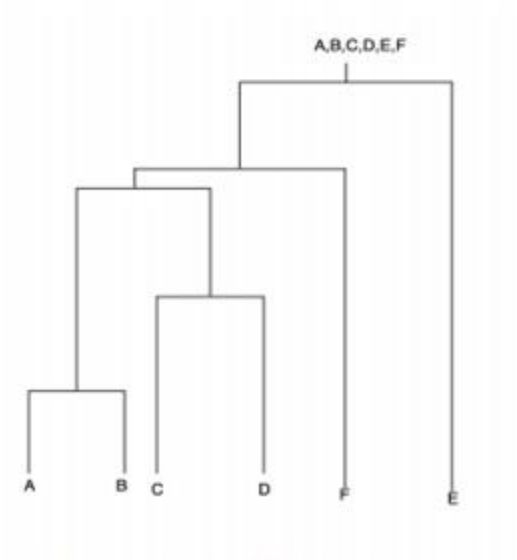
Distance matrix 2

	A,B,C,D	E	F
A,B,C,D	0		
E	0.28	0	
F	0.20	0.67	0

Distance matrix 3

	A,B,C,D,F	E
A,B,C,D,F	0	
E	0.28	0

Distance matrix 4



Dendrogram of HAC with single link

-Complete link

	A	B	C	D	E	F
A	0					
B	0.12	0				
C	0.51	0.25	0			
D	0.84	0.16	0.14	0		
E	0.28	0.77	0.70	0.45	0	
F	0.34	0.61	0.93	0.20	0.67	0

Distance matrix 0

	A,B	C	D	E	F
A,B	0				
C	0.51	0			
D	0.84	0.14	0		
E	0.77	0.70	0.45	0	
F	0.61	0.93	0.20	0.67	0

Distance matrix 1

	A,B	C,D	E	F
A,B	0			
C,D	0.84	0		
E	0.77	0.70	0	
F	0.61	0.93	0.67	0

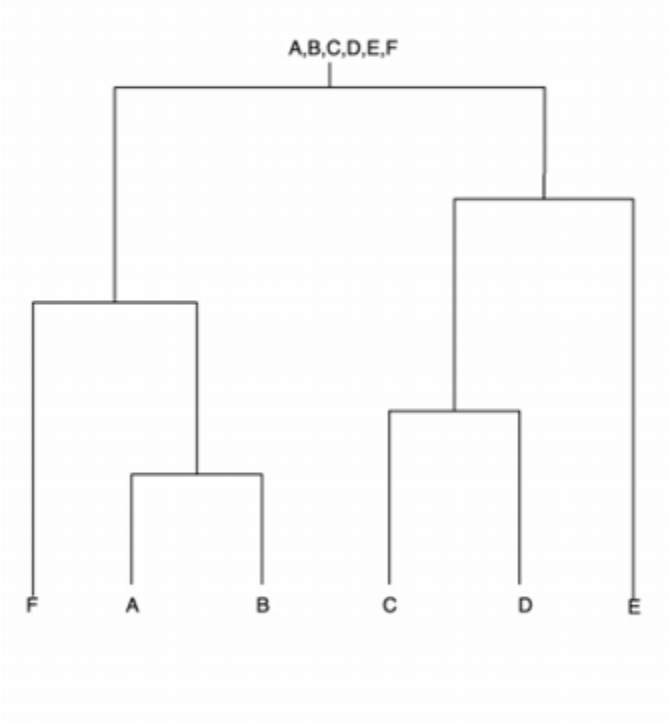
Distance matrix 2

	A,B,F	C,D	E
A,B,F	0		
C,D	0.93	0	
E	0.77	0.70	0

Distance matrix 3

	A,B,F	C,D,E
A,B,F	0	
C,D,E	0.93	0

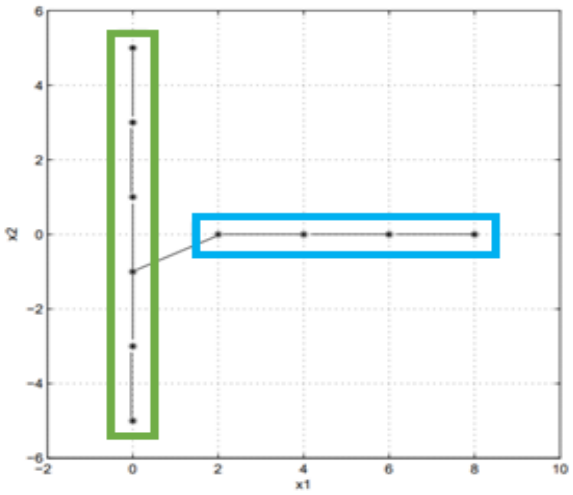
Distance matrix 4



Dendrogram of HAC with complete link

سوال ۵-

الف) همانطور که قابل مشاهده است، حرکت random walk ناشی از وزن دهی داده شده، می تواند بین نقاط $(0,-1)$ و $(2,0)$ جا به جا شود. از آنجا که وزن ها با فاصله بیشتر کاهش می یابند، وزن های مربوط به انتقالات داخل خوشه ای بیشتر از انتقالات میان خوشه ای خواهند بود. در نتیجه خوشه ها به صورت نهایی زیر می باشند.



ب) خیر. در الگوریتم خوشه بندی k-means، داده ها به نزدیک ترین میانگین (مرکز خوشه یا centroid) نسبت داده می شوند. همانطور که میبینیم، مراکز خوشه های سمت چپ و راست شکل داده شده به ترتیب $(0,0)$ و $(5,0)$ هستند. در نتیجه برای مثال نقطه ی $(2,0)$ به مرکز خوشه ی چپ یعنی داده ی$(0,0)$نزدیکتر است و به دسته ی سمت راست نسبت داده نخواهد شد و عضو خوشه ی چپ خواهد بود. پس خوشه بندی مشخص شده در قسمت قبل، نتیجه ی k-mean نمی باشند.