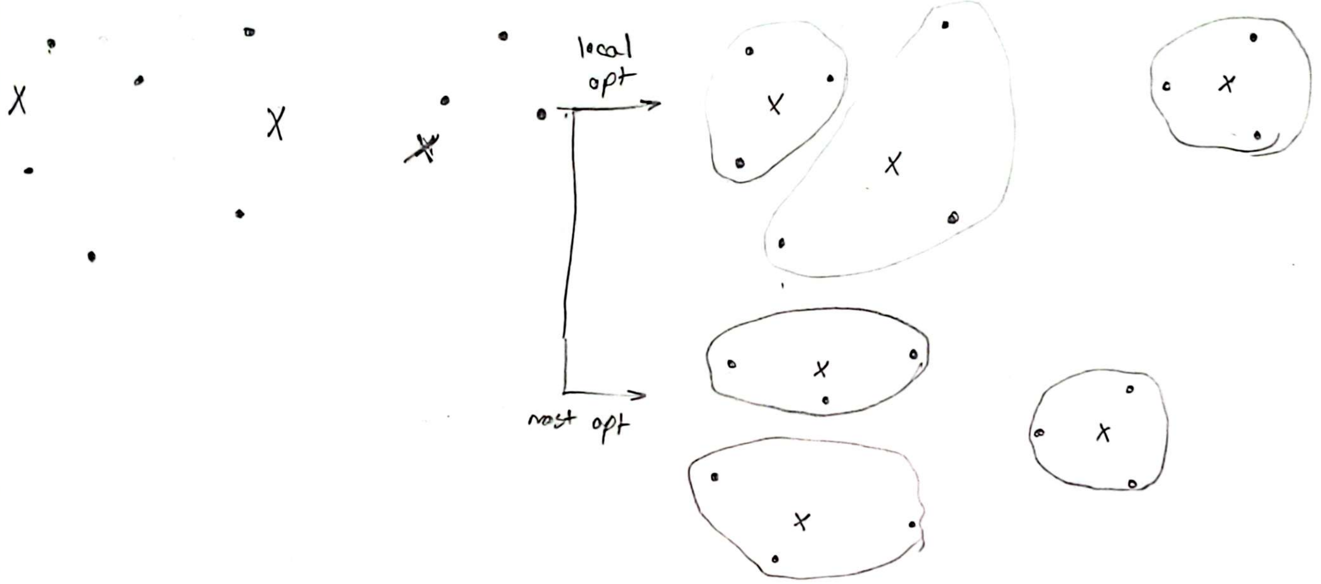
سوال ۱) الف) در اصل مفهوم تابع loss ما این است که فاصله هر نقطه از خوشه مورد نظرش را پیدا کند و آنها را باهم جمع بزند و حاصل مقدار همه خوشه ها را باهم جمع بزند تا تابع loss به وجود آید پس بدیهی است که اگر ما سعی کنیم تابع هزینه را به ازای هر خوشه مینیمم کنیم آنگاه به طور کلی حاصل جمع آن نیز مینیمم می‌شود و تابع loss مینیمم می‌شود.

ب) الگوریتم بدین گونه است که تعدادی node متناسب با تعداد خوشه ها انتخاب میکنیم و آنها را جایی قرار می‌دهیم (همین درست انتخاب کردن جای اولیه node ها باعث ایجاد local opt یا most opt می‌شود.) و در هر مرحله آن مرکز را با توجه به تابع loss جابجا میکنیم تا میزان loss کمتری داشته باشد و این را تا جایی ادامه می‌دهیم که دیگر مرکزی جابجا نشود پس همگرا می‌شود.

مثال:



ج) روش بدین گونه است که تابع loss را به ازای K های مختلف رسم میکنیم و به یک نمودار میرسیم مقدار K مورد نظر آن جایی است که در نمودار با کم کردن K مقدار زیادی از loss کم نشود مقدار K همان مقدار است. یک روش دیگر استفاده از silhouette score است که میزان شباهت یک نقطه با cluster ای که در آن است را با cluster های دیگر می‌سنجد و مورد دوم استفاده از gap statics است که کل تغییرات درون cluster ها را برای مقادیر مختلف K با یک مجموعه داده مرجع مقایسه می‌کند.

روش silhouette score: مقدار silhouette را برای تمامی داده های به ازای مقادیر مختلف K حساب می‌کند پس به ازای هر K مقدار میانگین این امتیاز ها را میگیرد و وقتی که این امتیاز بیشینه شود K مورد نظر یافت می‌شود.

روش gap statics: مقدار $W_K = \sum_{j=1}^{K} \frac{1}{2|S_j|} \sum_{x,y \in S_j} \|x-y\|^2$ را برای هر داده داخل cluster ها به ازای K های مختلف پیدا میکند و یک dataset هم سایز با dataset اصلی تولید میکند و میزان اختلاف W_K ها را پیدا میکند که بدین گونه است: $gap(K) = E[\log(W_K^{random})] - \log(W_K^{data})$ و حال K طوری انتخاب می‌شود که این اختلاف ماکسیمم شود یا میزان بهبود زیادی مشاهده شود.

د) می‌توان تابع loss را بدین گونه تعریف کرد: $L' = L + \lambda K$

مزایا: احتمال overfit شدن را کم میکند با محدود کردن تعداد cluster ها و ممکن است با محدودیت های دنیای واقعی بهتر هماهنگ شود.

معایب: احتمال underfit شدن به علت کم کردن تعداد cluster ها و نیاز به تنظیم پارامتر λ دارد.

سوال 2) خیر به دلایل زیر:

① مرکز یک میانگین وزن از تمام نقاط خوشه است که به آن از نظر آماری نماینده نقاط است، حرکت نقاط به سمت مرکز تضمین می‌کند که آنها در منطقه تعریف شده خوشه باقی می‌مانند.

② اگر همه نقاط را به میزان ثابتی به مرکز نزدیک کنیم آنگاه دیگر تشخیص مرکز ممکن است مبهم باشد.

③ چون الگوریتم cure نسبت به noise و نقاط دور افتاده مقاوم است که در این حالت ممکن است که این حالت رخ ندهد و اگر برای نقاط دور افتاده پروسه کنیم میزان کاهش فاصله نسبی بهتر عمل می‌کند.

سوال 3) الف) $U^TU = I$ $\quad V^TV = I$ $\quad \Sigma = \Sigma^T$ $\quad A = U\Sigma V^T \longrightarrow A^T = V\Sigma^T U^T = V\Sigma U^T$

$$AA^T = (U\Sigma V^T)(V\Sigma^T U^T) = U\Sigma V^TV\Sigma^TU^T = U\Sigma\Sigma^TU^T = U\Sigma^2U^T$$

$$A^TA = (V\Sigma^TU^T)(U\Sigma V^T) = V\Sigma^TU^TU\Sigma V^T = V\Sigma^T\Sigma V^T = V\Sigma^2V^T$$

مقادیر ویژه A و A^T یکی هستند و مقادیر ویژه AA^T و A^TA برابر هستند و توان 2، A و A^T هستند.

ب) در قسمت قبل دیدیم:

$$A^TA = V\Sigma^2V^T \xrightarrow{V^T\times \quad \times V} V^TA^TAV = V^TV\Sigma^2V^TV = \Sigma^2$$

حال با توجه به این تساوی می‌توان گفت که $\quad Q^TQ = I,\ \Lambda = \Sigma^T\Sigma \longrightarrow Q^TA^TAQ = \Lambda = \Sigma^2$; $\quad A^TA = Q\Lambda Q^T$

A^TA را می‌توان به صورت Q^TA^TAQ نوشت که Q یک ماتریس متعامد است.

سوال 4) الف | از آنجایی که احتمال sampling به توان 2 هر ستون وابسته است پس ستون‌های با نرم فروبنیوس بیشتر احتمال انتخاب بالاتری دارند و میزان خطای بین A و CC^+A براین اساس است که چقدر ماتریس C فضای ستون ماتریس A را نمایش می‌دهد و از آنجایی که rank خیلی کمتر از A است پس حتما اطلاعاتی از دست می‌دهیم. حال چون کلا 10 ستون درمی داریم می‌توان گفت:

$$\|A - CC^+A\|_F \simeq \epsilon\|A\|_F = 0.1\times 50 = 5$$

ب) مانند قسمت قبل ستون‌های انتخاب می‌شوند که نرم فروبنیوس بیشتری دارند. میزان خطای AB به نرم‌های جداگانه A و B، که با s اسکیل شده است مرتبط است پس:

$$\|AB - CR\|_F \leq \sqrt{5}\,\|A\|_F\|B\|_F\,\delta \longrightarrow \|AB - CR\|_F \leq 283$$

سوال 5) الف / جدول بدین گونه می‌شود :

	h	g	f	e	d	c	b	a
A	0	1	0	0	1	0	1	1
B	0	0	0	0	1	1	1	0
C	1	1	1	0	1	0	0	0

برای مثال :

$$D_j(g,h) = 1 - \frac{|g\cap h|}{|g\cup h|} = 1 - \frac{1}{2} = \frac{1}{2}$$

$h = (0,0,1)$

$g = (1,0,1)$

$g\cap h = (0,0,1) \longrightarrow 1 \text{ sum}$

$g\cup h = (1,0,1) \longrightarrow 2 \text{ sum}$

میزان شباهت آنها بدین گونه است .

Jaccard Distance

h	0							
g	0.5	0						
f	1	1	0					
e	1	1	1	0				
d	0.5	0.5	0.5	1	0			
c	1	0.5	0.5	1	0.5	0		
b	0.5	0.5	1	1	0.5	0.5	0	
a	0.5	0.5	1	1	0.5	1	0.5	0
	h	g	f	e	d	c	b	a

حال باید از بین این عناصر دو تا دو تا استون را merge کنیم برحسب Jaccard Distance آنها. برای مثال g و h را merge می‌کنیم و آنها را یک عضو می‌گیریم و با بقیه مقایسه می‌کنیم. پس این cluster را با بقیه Jaccard Distance می‌کنیم و پس d و c را merge می‌کنیم پس بدین گونه می‌شود {a} ، {b} ، {d,c} ، {e} ، {f} ، {h,g}

از محاسبات جدول‌ها صرف نظر می‌کنیم چون مانند قسمت اول است .

پس در این حالت نیز دوباره یکی از بیشانسه‌ها را با توجه به Jaccard Distance ، merge می‌کنیم که b به (h,g) می‌پیوندد پس clusterهای نهایی بدین گونه هستند .

{h,g,b} , {f} , {e} , {d,c,a}

ب) حال مقادیر جدول را باید بدون در نظر گرفتن فیلتر روی داده‌ها انجام بدهیم .

	h,g,b	d,c,a	f	e
A	$\frac{2+3+5}{3} = 3.33$	$\frac{5+4}{2} = 4.5$	—	1
B	$\frac{1+3}{2} = 2$	$\frac{3+4}{2} = 3.5$	2	1
C	$\frac{3+5}{2} = 4$	$\frac{3+1+2}{3} = 2$	4	—

ج)

$$\text{similarity}(A,B) = \frac{\vec{A}\cdot\vec{B}}{\|\vec{A}\|\,\|\vec{B}\|}$$

$u_A = (3.33, 4.5, 0, 1) \longrightarrow \|\vec{u}_A\| = \sqrt{3.33^2 + 4.5^2 + 1^2} \approx 5.7$

$u_b = (2, 3.5, 2, 1) \longrightarrow \|\vec{u}_b\| = \sqrt{2^2 + 3.5^2 + 2^2 + 1^2} = 4.6$

$u_c = (4, 2, 4, 0) \longrightarrow \|\vec{u}_c\| = \sqrt{4^2 + 2^2 + 4^2} = 6$

$\vec{u}_A \cdot \vec{u}_b = 3.33\times 2 + 4.5\times 3.5 + 2\times 0 + 1\times 1 = 23.41$

$\vec{u}_A \cdot \vec{u}_c = 3.33\times 4 + 4.5\times 2 + 0\times 4 + 1\times 0 = 22.32$

$\vec{u}_b \cdot \vec{u}_c = 2\times 4 + 3.5\times 2 + 2\times 4 + 1\times 0 = 23$

$\text{sim}(u_A, u_b) = \frac{23.41}{5.7\times 4.6} = 0.89$ $\quad$ $\text{sim}(u_b, u_c) = \frac{23}{4.6\times 6} = 0.83$ $\quad$ $\text{sim}(u_A, u_c) = \frac{22.32}{5.7\times 6} = 0.65$