بسمه تعالیٰ

امتحان میان ترم درس داده کاوی

نیم‌سال دوم تحصیلی ۰۰-۱۳۹۹

نام: امیرحسین کارگران خوزانی شماره دانشجویی: ۹۹۲۰۱۱۱۹

دانشکدهٔ مهندسی کامپیوتر
مدرس: دکتر فضلی

سؤال ۱.

راه حل.

من راستش نمیدونستم هر ستونی دقیقا معنیش چیه و بیشتر اونهایی که میدونستم معنیش چیه و ممکنه کمک کنه را نگه داشتم. تعدادی از ستون ها را با ۰ پر کردم. و بعدش اونایی که مقدار NAN زیادی بیشتر از ۸۰.۰ داشت را حذف کردم. سپس بقیه را با ۰ دوباره جایگذاری کردم. GOALS از مهمترین ستون ها می باشد چرا که برای مثال دروازه زبان ها یا مدافعان تعداد گل کمتری به ثمر رساندند و این برای مهاجهم ها برعکس و مقدار زیادی است. یا مثلا CS فقط برای دروازه زبان است. کرنرها نیز موارد مهمی هستند که بازیکنان جلویی بیشتر آن را می‌زنن، همچنین سانت‌ها نیز. تقریبا بیشتر فیلدها مهم هستند و بیشتر مهم است که چگونه با ان ها رفتار شود و مقدار Nan آن ها با چه مقدار مناسبی پر شود.

بخش ۵) ابتدا اثبات می‌کنیم که k-means به صورت یکنواخت $\psi\left(X^{(t)}\right)=\frac{1}{n}\sum_{j=1}^{K}\sum_{i=1}^{n}\|x_i,c_j\|^2$ را کاهش می‌دهد که $X^{(t)}$ تقسیم بندی فعلی $X_1^{(t)},\ldots,X_K^{(t)}$ با centroid های $c_1^{(t)},\ldots,c_K^{(t)}$ است. در هر مرحله از اجرای الگوریتم k-means، $\mathcal{A}\left(x_i\right)$ به صورت $\mathcal{A}\left(x_i\right)\longleftarrow\underset{j\in\{1,\ldots,K\}}{\arg\min}\|x_i-c_j\|^2$ تعریف می‌شود. که به $\mathcal{A}^{(t)}$، تابع انتساب گفته می‌شود. در این صورت از آنجا که $\mathcal{A}\left(x_i\right)$ مقدار $\|x_i-c_j\|^2$ را روی همه $j\in\{1,\ldots,K\}$ کمینه می‌کند و $c_j^{(t+1)}$ نیز مقدار $\|x_i-c_j\|^2$ را روی همه $x_i\in X_j$ کمینه می‌کند، خواهیم داشت:

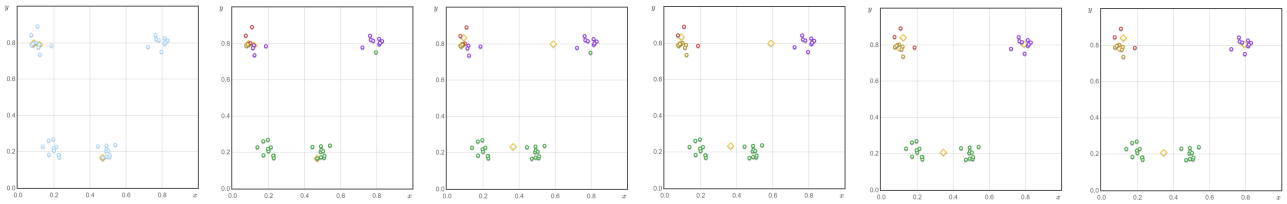
$$\begin{aligned}\psi\left(X^{(t)}\right)&\geqslant\sum_{j=1}^{K}\sum_{x_i\in X_j^{(t)}}\left\|x_i,c^{(t)}_{\mathcal{A}^{(t+1)}(x_i)}\right\|^2\\&\geqslant\sum_{j=1}^{K}\sum_{x_i\in X_j^{(t)}}\left\|x_i,c_j^{(t+1)}\right\|^2\\&\geqslant\psi\left(X^{(t+1)}\right)\end{aligned}$$

در این صورت اثبات می‌شود که این الگوریتم به طور یکنواخت مقدار $\frac{1}{n}\sum_{j=1}^{K}\sum_{i=1}^{n}\|x_i-c_j\|^2$ را کاهش می‌دهد.

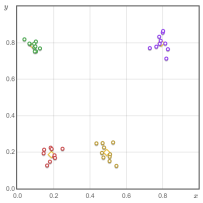
حال به اثبات صورت سوال گفته شده می‌پردازیم و تنها کافی است که نشان دهیم الگوریتم k-means پس از تعداد محدودی step متوقف می‌شود. از انجا که تعداد محدودی تقسیم‌بندی وجود دارد و تعداد آن برابر $\binom{N}{k}$ است، در نتیجه دنباله $\psi\left(X^{(t)}\right)_{t\in\mathbb{N}}$ تعداد محدودی مقدار خواهد داشت. این بدین معناست که حالتی وجود دارد که $\psi\left(X^{(t+1)}\right)=\psi\left(X^{(t)}\right)$ باشد و در نتیجه در step شماره t داریم: $X^{(t+1)}=X^{(t)}$ در غیر این صورت برخی از element ها اشتباه دسته‌بندی شده است. در این اثبات صرفا یک باند نمایی برای مساله پیدا کردیم که نشان دهیم حتما در تعداد محدودی بار الگوریتم متوقف می‌شود.

بخش ۶) از آنجا که استفاده از منابع اینرنتی در طول امتحان مجاز است من از مثال این link استفاده کردم. هدف این لینک آن است که نشان دهد الگوریتم k-means در چه مثال‌هایی خوب عمل می‌کند و در چه مثال‌هایی بد. اما من مثالی که در آن این صفحه از آن به عنوان مثال خوب یاد کرده است صرفا استفاده کردم و نتیجه‌گیری ها کاملا مشاهدات و

استدلال‌های خودم است و تنها از اجرای آنلاین ان استفاده کردم. همانگونه که در این صفحه گفته شده این مثالی است که در آن k-means به خوبی عمل می‌کند. اما اگر چندین بار الگوریتم را ران کنیم تا نقاطی که در ابتدا به عنوان نقاط تصادفی مرکز centroid ها initial می‌شود به گونه نامناسبی قرار گیرد و در نتیجه در ادامه اجرای الگوریتم در یک مینیمم محلی گیر می‌کند. برای مثال در این شکل ۱ ما انتظار داشتیم که ۴ کلاستر به شکل زیر ۲ در آید اما انتخاب نقاط اولیه باعث شده در یک مینیمم محلی گیر کنیم. و در نتیجه دو کلاستر پایینی با یکدیگر ادغام شده و یکی از کلاسترها به دو کلاستر تقسیم شود. دلیل این امر نیز آن است که الگوریتم روی همه تقسیم‌بندی‌های یک تابع محدب نیست و در نتیجه خروجی به مقداردهی اولیه بستگی دارد. به همین منظور توصیه می‌شود گه چندین بار الگوریتم ران شود و خروجی مطلوب به عنوان جواب نهایی برداشته شود.



شکل ۱: اجرای الگوریتم کلاسترینگ از سمت چپ به راست



شکل ۲: کلاسترینگ مطلوب

■

سؤال ۲.

راه حل.

در نوتبوک به پیوست الگوریتم مقاله پیاده سازی شده است. در ادامه بحث می‌کنیم که الگوریتم این مقاله چگونه کار می‌دهد. اگر فرض کنیم که U_1, U_2 و ... به صورت iid بین ۰ و ۱ انتخاب شده باشند آنگاه اولین n ای که $S_n = \sum_{i=1}^{n} U_i > 1$ باشد را داشته باشیم. انگاه $E(N)$ روی تمام n هایی که از این روش بدست امده بگیریم برابر e خواهد بود. بنابراین یک تابع نوشتیم که n را برامون بدست بیاره و در نتیجه تعداد زیادی از اون‌ها را با هم جمع کنه از طریق reduce و در نهایت هم تقسیم بر تعداد جمع این اعداد را بکنیم.

■

سؤال ۳.

راه حل.

حال اثبات می‌کنیم که چرا $E(N) = e$ می‌شود. ابتدا تابع احتمال متغیر N را تعیین می‌کنیم. به طور واضح N احتمال غیر صفر بر روی مجموعه $T = \{۲, ۳, ۴, ۵, \ldots\}$ دارد و داریم: $\Pr(N = n) = \Pr(S_n > ۱ \ n \in T, \Pr(S_{n-۱} < ۱) - \Pr(S_n < ۱)$
آخرین معادله به راحتی می‌تواند با نوشتن E_n برای رویداد $(n = ۱, ۲, ۳, \ldots)$ $\{S_n < \mathrm{I}\}$ و با توجه به $\Pr(E_n^c \cap E_{n-۱}) =$ $\Pr(E_{n-۱}) - \Pr(E_n)$ تایید شود.
حال نیاز است که مقدار $\Pr(S_n < ۱$ برای $n = ۱, ۲, ۳$، ... محاسبه شود. این احتمال مقدار ۱ را برای $n =$ ۱ دارد. و اثبات می‌شود که $(n = ۱, ۲, ۳, \ldots) \Pr(S_n < ۱) = ۱/n!$ است. در حالیکه می‌توان گفت $\Pr(N =$ $E(N) = \Sigma_{n=۲}^{x} n \Pr(N = n) = e$ است. بنابراین خواهیم داشت: $n) = (n - ۱)/n!(n = ۲, ۳, ۴, \ldots)$.

قسمت اول) ابتدا به تعاریف می‌پردازیم و از روی تعاریف به نتیجه می‌رسیم. چه موقع یک FIS ماکسیمال است؟ اگر هیچ کدام از سوپرست های آن FIS دیگر پر تکرار نباشد و چه موقع یک FIS را بسته می‌نامیم؟ زمانی که هیچ سوپرستی از ان مقدار ساپورت برابر یا بزرگتری نداشته باشد. پس یک مجموعه بسته می‌تواند سوپرستی پر تکرار داشته باشد اما شرط بسته بودن نقض نشود به زبانی دیگر سوپرست پرتکراری می‌تواند داشته باشد ولی آن سوپرست مقدار ساپورت برابر یا بزرگتری نداشته باشد. پس FIS ماکسیمال زیر مجموعه FIS بسته است. همچنین اگر از طریق برهان خلف نیز به این مساله نگاه شود و در نظر بگیریم که FIS ماکسیمال زیر مجموعه FIS های بسته نیست پس باید شرط آن را نقض کند در صورتی که مجموعه FIS ماکسیمالی نمی‌توان یافت که سوپرست آن از آن مقدار ساپورت برابر یا بزرگتری داشته باشد چرا که با ماکسیمال بودن FIS در تناقض است.
قسمت دوم) همانگونه که این موضوع در کلاس نیز بحث شد اگر به فرمول confidence نگاه کنیم کاملا متوجه آن خواهیم شد:

$$confidence(\{A \to B\}) = \frac{support(\{A \to B\})}{support(\{A\})} = \frac{freq(A, B)}{frq(A)} = \Pr(B|A)$$

اگر آیتمی از A به B اضافه شود آنگاه صورت کسر تغییر نخواهد کرد. اما ممکن است باعث شود که مخرج کسر بزرگتر شود (چرا که یک آیتم از مجموعه آن حذف شده است) و این موضوع باعث می‌شود مقدار confidence کاهش یابد و اگر تغییر نکند ثابت ماند که همان گزاره صورت سوال است.
قسمت سوم) بله از هر دو آن‌ها می‌توان استفاده کرد. با توجه به قسمت اول می‌توان گفت که اگر سمت چپ rule یک عبارت ماکسیمال پرتکرار داشته باشیم آنگاه دیگر نیازی به بررسی سوپرست‌های آن نیست چرا که دیگر پرتکرار نیستند تا در سمت چپ قرار گیرند. با توجه به قسمت دوم می‌توان rule هایی را که انتقال کالا باعث کم شدن confidence می‌شود را در صورتی که مقدار confidence آن‌ها از ترشولد کمتر می‌شود را در نظر نگرفت.

■

سؤال ۴.

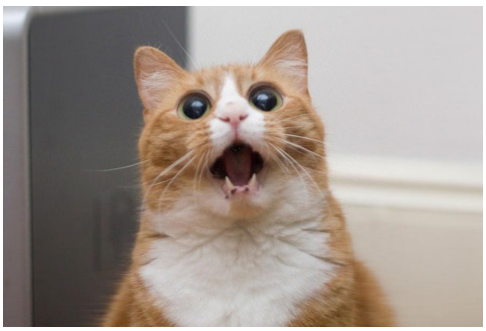
راه حل.

در قسمت ۱ با استفاده از کتابخانه pillow اقدام به بارگزاری عکس کردم و اندازه ان را چاپ کرده و کانال های مختلف رنگ را از یکدیگر نیز جدا کردم. در قسمت ۲ تابعی نوشتم که بتواند حاصلضرب اجزای مختلف svd را برای یک کانال حساب کند تا تصویر فشرده شده بدست آید. در قسمت ۳ برای بعد ۱۰۰ اقدام به محاسبه هر کانال کردم و در قسمت ۴ با استفاده از قسمت ۳ عکس کامل را ساختم این عکس را برای این پارامتر می‌توانید در شکل ۳ مشاهده کنید. این کار موجب

شد عکس از ۷۱ به ۳۸ کیلوبایت فشرده شود. سپس در قسمت ۵، با گام ۱۰ از ۱ تا ۲۰۰ حرکت کردم و عکس‌های مختلف را ساختم. همانگونه که در نوتبوک میبینید دو عکس مربوط به پارامتر ۵۱ و ۶۱ را گذاشتم که با یکدیگر تفاوت خاصی (چشمی) ندارند و هر دو از کیفیت مناسبی برخوردار هستند. بقیه عکس ها را می‌توانید در شکل ۴ مشاهده کنید. در قسمت ۶ حجم تک تک عکس ها را با کتابخانه os خواندم و نمودار مربوطه را کشیدم. همانجور که میبینید با اضافه شدن بعد حجم عکس نیز افزایش می‌یابد. دلیل ان نیز کم تر حذف شدن دادگان در تابع فشرده ساز است که پارامتر dim ان را کنترل می‌کند. در واقع می‌توان به جای این که مقدار ۳ $* m * n$ تعیین کننده سایز نسبی عکس باشد، مقدار ۳ $* (۱ + m + n) * dim$ برابر سایز نسبی (بایت) عکس فشرده شده باشد. اولین عبارت حجم اصلی تصویر با ۳ کانال است که طول در عرض آن ضرب شده است. اما در دومی تنها برای ۳ کانال و ۳ ماتریس U, Z, V اعداد ذخیره می‌شوند که تعداد اعداد غیر صفر آن‌ها برابر عبارت دوم است. بدین صورت با افزایش dim این مقدار افزایش می‌یابد که در نمودار نیز مشخص است. اما از یه جایی به می بینید که این نمودار شیبی کمتری پیدا می کند و از این فرمول حجم نسبی گفته شده کاملا پیروی نمی کند این به آن دلیل است که خود عکس وقتی به صورت jpg ذخیره می‌شود توسط jpg از یک نوع فشرده‌ساز استفاده می‌کند و نمودار خود jpg خود نیز به شکل جذری است .

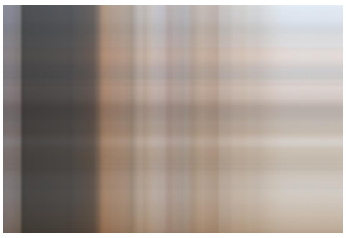
برای قسمت ۷، طبق اسلاید باید انرژی ماتریس sigular-value در هر یک از حالت ها محاسبه شود و پارامتر بعد مناسب به نحوی انتخاب شود که حداقل ۹۰ درصد از انرژی را در خود داشته باشد. کد این قسمت نیز مانند بقیه قسمت ها در نوتبوک زده شده است و مقدار این فاصله انرژی با حد ۰٫۰۰۱ مقدار انرژی محاسبه شده است. این مقدار dim برای هر یک از کانال ها به صورت مجزا حساب شده و سپس max آن در نظر گرفته شده است تا مقدار حد مورد نظر تضمین شود.

همچنین راهکار دیگری که به ذهنم می‌رسد این است که ما دوست داریم حجم فایل ها کم شود در عین حالت ارور بین عکس فشرده شده و عکس اصلی نیز کم باشد. با افزایش مقدار dim مشاهده می‌کنیم که این خطا کاهش می یابد اما در عوض حجم افزایش پیدا می کند (شکل خطا در بخش اخر نوتبوک کشیده شده است). در این صورت می توان یک تابع هدف تعریف کرد که با توجه به مقدار خطا و مقدار حجم فایل برای ما مقدار dim مناسب را پیدا کند. این تابع هدف دارای دو پارامتر مقدار خطا و حجم فایل در هر dim است که با استفاده از ۲ ضریب مثبت باید کمینه شود. هر کدام از ضریب ها در جه اهمیت مد نظر ما را برای هر پارامتر بیان می‌کند.

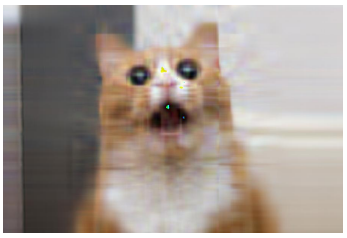


شکل ۳: بعد ۱۰۰

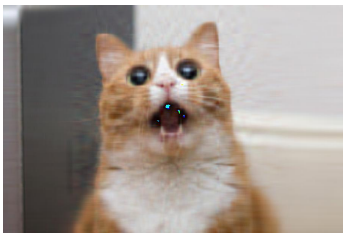
■



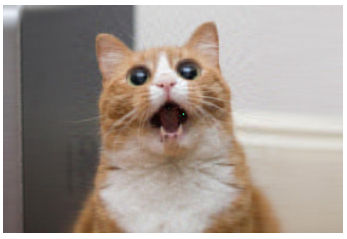
(آ) ابعاد ۱



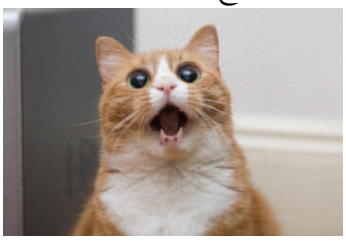
(ب) ابعاد ۱۱



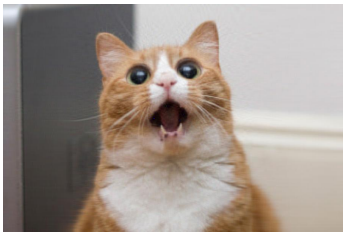
(ج) ابعاد ۲۱



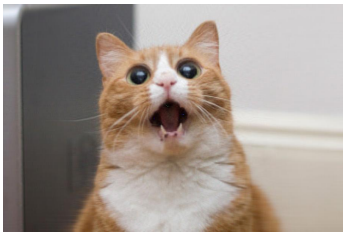
(د) ابعاد ۳۱



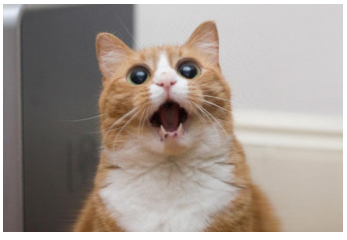
(ه) ابعاد ۴۱



(و) ابعاد ۵۱



(ز) ابعاد ۶۱



(ح) ابعاد ۷۱

شکل ۴: شکل ها با پارامترهای مختلف بعد