به نام خدا

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

گزارش تمرین چهارم درس یادگیری ماشین

پاسخ سوالات تشریحی

استاد درس: دکتر احسان ناظرفرد

دانشجو: فاطمه غلام‌زاده

۹۹۱۳۱۰۰۳

نیم سال دوم ۱۳۹۹-۱۴۰۰

پاسخ سوالات تشریحی

سوال ۱)

الف)

۱. Mean یا میانگین : یکی ازمعایب این روش این است که به داده های نویز حساس است. این روش در Kmeans استفاده می‌شود.
۲. Centroid : بر اساس تفسیر فیزیکی (گاهی مرکز جرم جسمی است که توسط نقاط تعریف می شود) مرکز گاهی اوقات مرکز جرم یا مرکز تقسیم می شود. مانند میانگین، موقعیت مرکز نیز فاصله توان دو از نقاط دیگر را به حداقل می رساند.
۳. Median یا میانه: نسبت به قبلی حساسیت کمتری به داده های نویز دارد.
۴. Medoid: نقطه داده ای است که "بیشترین شباهت" را به سایر نقاط داده دارد. این روش شبیه میانگین است اما حتما باید یکی از خود اعداد داخل کلاستر را انتخاب کنیم در صورتی که در دو روش قبل اینطور نبود. در داده‌هایی مانند داده‌های گرافی که میانگین قابل تعریف نیست، می‌توان از medoid که یکی از خود داده‌هاست استفاده کرد.

ب)

معیارهای فاصله مختلفی می‌توانند در الگوریتم k-means مورد استفاده قرار بگیرند که ۳ تا از معروف‌ترین و رایج‌ترین آن‌ها عبارتند از :

۱. فاصله منهتن : برای دو نقطه p و q که در فضای n بعدی تعریف شده اند فاصله منهتن از رابطه زیر بدست می‌آید :

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

۱. فاصله اقلیدسی: برای دو نقطه p و q که در فضای n بعدی تعریف شده اند فاصله منهتن از رابطه زیر بدست می‌آید :

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

۲. فاصله کسینوسی: اگر A و B دو بردار (نقطه) باشند شباهت کسینوسی آن‌ها طبق رابطه زیر محاسبه می‌شود:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

فاصله کسینوسی به صورت ۱ منهای شباهت کسینوسی تعریف می‌شود.

مقایسه: به طور کلی انتخاب معیار فاصله بر خوشه بندی تأثیرگذار است، اما این تاثیر به مجموعه داده و هدف بستگی دارد که کدام معیار فاصله برای کاربرد خاص شما مناسب است. بنابراین به طور کلی نمی‌توان مشخص کرد که کدام معیار فاصله بهتر است. با این حال ، K-Means به طور ضمنی بر اساس فاصله اقلیدسی به صورت جفتی بین نقاط داده است و به همین دلیل در اکثر موارد از فاصله اقلیدسی استفاده می‌شود. از طرفی شباهت کسینوسی و شباهت اقلیدسی به صورت خطی به‌هم مرتبط هستند.

در جدول زیر مزایا و معایب این معیارهای فاصله آورده شده است :

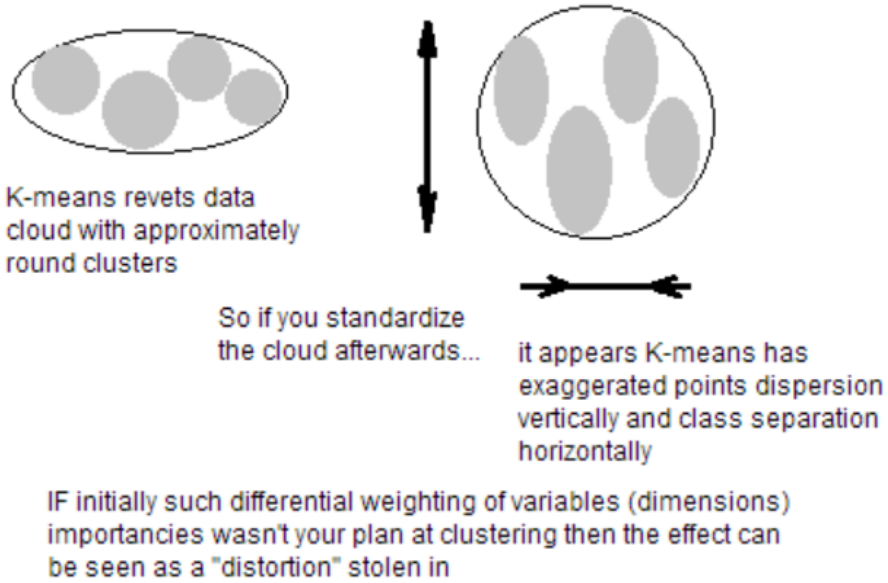
نام معیار	مزایا	معایب
اقلیدسی	بیشتر روش های بهینه سازی با در نظر گرفتن فاصله اقلیدسی طراحی شده اند و هزینه های محاسباتی می توانند به خوبی محدود شوند. فاصله اقلیدسی از نظر گرافیکی ساده است و برای اکثر مردم به خوبی قابل درک است.	اینکه فاصله‌ها را مربع می‌کنیم به طور قابل توجهی اثر داده‌های پرت را تقویت می کند.
منهتن	در برخی از مسائل امکان حرکت مورب بین نقاط وجود ندارد و فقط به صورت شبکه ای و در راستای یکی از محور ها می‌توان حرکت کرد. در چنین مسائلی نمی‌توان از فاصله اقلیدسی بهره گرفت و در چنین مواردی از فاصله منهتن استفاده می‌شود. این معیار فاصله به طور قابل توجهی در برابر داده‌های پرت مقاومت دارد.	الگوریتم های بهینه سازی در فاصله منهتن اغلب از نظر محاسباتی گران تر و پیچیده تر هستند.
کسینوسی	از آن‌جایی که فاصله کسینوسی و اقلیدسی به صورت خطی با هم ارتباط دارند می‌توان که فاصله کسینوسی هم از مزایای فاصله اقلیدسی برخوردار است	این معیار فاصله هنگامی استفاده می شود که اندازه بین بردارها مهم نیست بلکه جهت‌گیری بردارها مهم است. در برخی موراد هنگام خوشه‌بندی استفاده از فاصله کسینوسی نمی‌تواند مناسب باشد زیرا ممکن است نقاطی از مرکز خوشه فاصله یکسان داشته باشند ولی جهت مخالف باشند، با این حال در یک خوشه قرار می‌گیرند اما چون در جهت مخالف هستند فاصله کسینوسی آن‌ها بسیار زیاد می‌شود و به احتمال زیاد در یک گوشه قرار نمی‌گیرند.

ج)

اگر فیچرها از واحدهای غیرقابل مقایسه ای هستند (به عنوان مثال قد در سانتی متر و وزن در کیلوگرم) ، حتما باید نرمال‌سازی انجام شود. زیرا الگوریتم نباید نسبت به متغیرهایی با اندازه بالاتر بایاس داشته باشد. برای غلبه بر این مشکل از نرمال سازی بهره می‌گیریم.

حتی اگر متغیرها از واحد یکسانی باشند اما واریانس کاملاً متفاوتی را در راستاهای مختلف نشان دهند ، باز هم ایده خوبی است که قبل از K-means استاندارد شوند. در شکل‌های زیر می بینید که خوشه بندی K-mean در همه جهات فضا "ایزوتروپیک" است و بنابراین تمایل به تولید خوشه های کم و بیش گرد (و نه کشیده) دارد. در این شرایط ، گذاشتن واریانس های نابرابر معادل وزن دادن

بیشتر به متغیرهایی با واریانس کمتر است ، بنابراین خوشه ها تمایل دارند که در امتداد متغیرهای با واریانس بیشتر از هم جدا شوند.



د)

دورترین فاصله یا پیوند کامل(Complete-Linkage)

شیوه محاسبه $\max\{d(a,b):a\in A,b\in B\}$

نزدیکترین فاصله یا پیوند تکی(Single-Linkage)

شیوه محاسبه $\min\{d(a,b):a\in A,b\in B\}$

پیوند میانگین (average link)

شیوه محاسبه $(1/|A|.|B|)*\sum a\in A\sum b\in B\ d(a,b)$

average :O(n^2 log n)
single : O(n^2 log n).
complete: O(n^2 log n)
پیچیدگی زمانی هر سه یکی است البته در روش سینگل با تغییراتی می‌توان پیچیدگی زمانی را به O(n^2) نیز رساند. از نظر حساسیت به داده پرت :

Single-link: به داده‌های پرت (outliers) حساس است. به عنوان دلیل می توان گفت که شاید خوشه هایی که به هم ماهیت نزدیکی ندارند به علت داده پرت با هم لینک شوند.

Complete-link: به شدتِ single-link به داده‌های پرت حساس نیست ولی باز هم حساس است. در این مورد هم می‌توان گفت شاید داده‌ها از نظر ماهیت به هم نزدیک باشند اما به علت فاصله زیاد داده‌های پرت به هم لینک نشوند.

Average-link: نسبت به دو مورد دیگر بسیار بسیار کمتر به داده‌های پرت حساسیت دارد، زیرا همواره میانگین فاصله را در نظر می‌گیرد.

ه)

خوشه بندی تجمعی (پایین به بالا) با هر شی در یک خوشه جداگانه شروع می‌شود. خوشه ها با گروه بندی اجسام به خوشه های بزرگتر و بزرگتر تشکیل می‌شوند. پیچیدگی زمانی برابر با O(n^3) و فضای مورد نیاز حافظه نیز برابر با O(n^2)است. بنابراین با افزایش حجم داده‌ها، سرعت و فضای حافظه برای اجرای عملیات خوشه‌بندی به شدت افزایش می‌یابد. به همین دلیل معمولا از این الگوریتم برای خوشه‌بندی کلان داده (Big Data)استفاده نمی‌شود. . در تحقیقات بازاریابی معمولاً از روش‌های پایین به بالا استفاده می شود.

خوشه بندی تقسیم بندی (بالا به پایین) با تمام اشیا گروه بندی شده در یک خوشه شروع می شود. خوشه ها تقسیم می شوند تا زمانی که هر جسم در یک خوشه جداگانه قرار گیرد. خوشه‌بندی بالا به پایین با یک خوشه که شامل تمامی مشاهدات می‌باشد شروع شده و سپس به‌طور بازگشتی یکی از خوشه‌های موجود را به خوشه‌های کوچکتری تقسیم می‌کند. یکی از برتری‌های این روش نسبت به خوشه‌بندی تجمعی برای مواردی است که بخواهیم داده‌ها را به تعداد کمی خوشه تجزیه کنیم. برای تجزیه هر خوشه می‌توان از الگوریتم‌های متفاوتی مانند K-Meansیا K-Medoids با k=2 استفاده کرد. به‌طور کلی هر الگوریتم خوشه‌بندی که حداقل دو خوشه خروجی تولید می‌کند در خوشه‌بندی تجزیه‌ای (بالا به پایین) قابل استفاده است.

الگوریتم خوشه‌بندی تقسیم بندی (بالا به پایین) بالا به پایین دقیق تر نیزاست. خوشه بندی تجمعی با در نظر گرفتن الگوهای محلی یا نقاط همسایه بدون در نظر گرفتن توزیع کلی داده ها تصمیم گیری می‌کند. این تصمیمات اولیه قابل لغو نیستند. در حالی که خوشه بندی تقسیم بندی (بالا به پایین) توزیع کلی داده را هنگام تصمیم گیری برای تقسیم بندی سطح بالا در نظر می‌گیرد.

و)

یک راه برای تعیین epsilon این است که KNN را برای تخمین فاصله هر نقطه تا نزدیکترین همسایه در همان پارتیشن اعمال کنید. اکنون که مقدار eps را داریم ، بنابراین می توانیم همسایگان هر نقطه را در یک فاصله مشخص که epsilon neighborhood نامیده می‌شود حساب کنیم تا min_pts را پیدا کنیم. بنابراین ، یک نکته این است که هر زمان با الگوریتم های بدون نظارت مشکلی داشتیم، سعی کنیم به فکر برخی از الگوریتم های نظارت شده باشیم و از قدرت آن‌ها استفاده کنیم.

یک راه‌حل دیگر برای یافتن مقدار eps بهینه استفاده از فاصله اقلیدسی است. داده ها را مرتب می‌کنیم و سعی می‌کنیم فاصله بین همسایگان آن را پیدا کنیم تا حداقل فاصله بین آنها را پیدا کنیم و حداقل فاصله را ترسیم کنیم. مقادیر حداقل فاصله مقدار eps آنها به ما می دهد. یک راه دیگر برای یافتن مقدار min_pts نیز استفاده از معیار silhouette score است.

مزایای DBScan	معایب DBScan
مقاومت نسبت به نویز	برای سیستم های چند پردازنده قابل تقسیم نیست
می تواند خوشه هایی با اشکال و اندازه های مختلف را پیدا کند.	در صورت تغییر چگالی و اینکه مجموعه داده بسیار کم است ، خوشه را شناسایی نمی کند.
می‌تواند خوشه‌هایی را بیابد که کاملا توسط خوشه‌های مختلفی احاطه شده‌اند	حساس به پارامترهای خوشه بندی minPoints و EPS.
	مجموعه داده هایی که تراکم آنها تغییر می کند مشکل است.
	نمونه گیری بر معیارهای تراکم تأثیر می گذارد.

سوال۲)

الف:

۱. در یادگیری تقویتی نمونه‌های یادگیری بصورت ورودی/ خروجی مطرح نمی‌شوند بلکه بعد از اینکه عامل عملی را انجام داد پاداشی را دریافت می‌کند و به مرحله بعدی می‌رود. عامل هیچ گونه اطلاعی در مورد اینکه در هر حالت بهترین عمل چیست را ندارد. بلکه این وظیفه

عامل است که در طول زمان تجربه کافی در مورد حالت‌ها، عمل های ممکن، انتقال و پاداش جمع آوری نموده و عملکرد بهینه را یاد بگیرد.

۲. تفاوت دیگر در اینجاست که در یادگیری تقویتی سیستم باید کارائی آنلاین بالائی داشته باشد. زیرا اغلب ارزیابی سیستم بطور همزمان صورت می پذیرد.

یادگیری تقویتی	یادگیری بدون نظارت	یادگیری با نظارت
یک فیدبک یا تابع از محیط دریافت می‌کنیم به صورت سیگنال عددی است (Numeric reward signals)	هیچ label ای نداریم	برای هر ورودی label داریم
از روی تعامل با محیط یاد می‌گیریم	از روی داده‌هایی که برچسب ندارند، مثلا شباهتشان، یاد می‌گیریم	از روی مثال‌های برچسب خورده یاد می‌گیریم

ب:

۱) الگوریتم های Value iteration، تابع مقدار بهینه را جستجو می کنند که متشکل از مقدار بازگشتی حداکثر از هر حالت یا از هر جفت حالت عمل است. الگوریتم های Value iteration ، سیاست ها را با ساختن توابع مقدار آنها ارزیابی می کنند (به جای تابع مقدار بهینه) ، و از این توابع مقدار برای یافتن سیاست های جدید و بهبود یافته استفاده می کنند. بنابراین به یک معنا الگوریتم های Policy iteration مبتنی بر خط مشی‌ها هستند (که برای آنها تابع مقدار را محاسبه می کنید و سپس سیاست ها را بهبود می بخشید) ، در حالی که Value iteration بر اساس خود تابع مقادیر است (یعنی یافتن تابع مقدار بهینه).

۲) Value iteration به سیاست اولیه قابل قبول نیاز دارد در حالی که Value iteration نیازی به آن ندارد.

۳) Value iteration راه حل پشتیبان گیری کامل نامیده می‌شود و می تواند محاسبات قابل توجهی را انجام دهد در حالی که Value iteration راه حل پشتیبان گیری جزئی نامیده می‌شود و محاسبات کمتری را انجام می‌دهد.

۴) Value iteration یک روش بازگشتی است و از سیاست های قبلی برای به روزرسانی خط مشی جدید استفاده می‌کند.

Policy Iteration:

$$\pi_0 \xrightarrow{\text{E}} V^{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} V^{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi^* \xrightarrow{\text{E}} V^*$$

- دو سیاست متوالی با هم مقایسه می‌شوند اگر متفاوت بود، دوباره مراحل تکرار می‌شوند و در غیر این صورت الگوریتم پایان می‌یابد. شبه کد آن به این صورت است:

1. Initialization
   $V(s) \in \Re$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Repeat
   $\Delta \leftarrow 0$
   For each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s'} \mathcal{P}_{ss'}^{\pi(s)} \left[\mathcal{R}_{ss'}^{\pi(s)} + \gamma V(s')\right]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   $b \leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s'} \mathcal{P}_{ss'}^{a} \left[\mathcal{R}_{ss'}^{a} + \gamma V(s')\right]$
   If $b \neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop; else go to 2

Value Iteration:

- برای کوتاه کردن برنامه از این روش استفاده می‌شود:

$$\begin{aligned} V_{k+1}(s) &= \max_a E\left\{r_{t+1} + \gamma V_k(s_{t+1}) \mid s_t = s, a_t = a\right\} \\ &= \max_a \sum_{s'} \mathcal{P}_{ss'}^{a} \left[\mathcal{R}_{ss'}^{a} + \gamma V_k(s')\right], \end{aligned}$$

شبه کد به این صورت تغییر می‌کند:

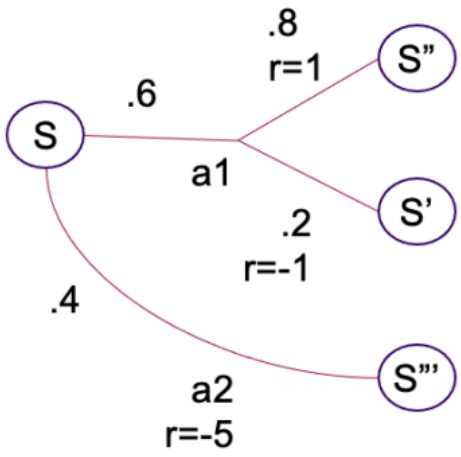
Initialize V arbitrarily, e.g.,$V(s) = 0$, for all $s \in \mathcal{S}^+$

Repeat
  $\Delta \leftarrow 0$
  For each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \max_a \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V(s')\right]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$ (a small positive number)

Output a deterministic policy, π, such that
  $\pi(s) = \arg\max_a \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V(s')\right]$

سوال ۳)

الف) در شکل زیر تابع مقدار را بدست آورید.



با توجه به شکل داریم:

$$P_{ss'}^{a_1} = 0.2$$
$$P_{ss''}^{a_1} = 0.8$$
$$P_{ss'''}^{a_2} = 1$$
$$\pi(s, a_1) = .6$$
$$\pi(s, a_2) = .4$$
$$V^{\pi}(s) = .6*(.8*1 + .2*-1) + .4*(1*-5) = -1.64$$

ب) چرا گاما در محاسبات تاثیر نداشت؟

زیرا فقط ۱ مرحله پیش می‌رویم.