(1) (الف)

PCA → Eigen-Decomposition: Normal way

PCA → Singular-Value-Decomposition: Our Way

X: Data Matrix $X_{n \times p}$

فرض کنیم که X در واقع روی تک تک ستون ها میانگین صفر دارد این کار با یک نرمال سازی ساده قابل انجام است: $X' = X - \mu_{mean}$ حال ماتریس کوواریانس X را حساب کنیم و چون centered است تفاضل از میانگین لازم است

$$C = \frac{X^T X}{n-1} = \underbrace{V}_{\text{eigenVectors}} \underbrace{L}_{\text{eigenValues (sorted)}} V^T$$

چون متقارن است می‌توان diagonalize شود

خب حال ماتریس V در واقع ستون های اصلی (Principal Component) ها را در بر دارد.

حال اگر یک SVD روی X انجام هم داریم

$$X = USV^T \Rightarrow C = \cancel{US}\, \frac{VSU^T USV^T}{n-1}$$

$$\xRightarrow{U^T = U^{-1}} V \frac{S^2}{n-1} V^T$$

این بیان معنی است که ستون های V همان جهت های اصلی هستند. در واقع اجزای اصلی توسط فرمول زیر قابل تولید هستند:

$$XV = USV^T V = US$$

۱) ب) فرض کنیم تعداد نمونه‌ها N و تعداد ویژگی‌ها D باشد داریم:

PCA $\rightarrow$ SVD $\Rightarrow$ order$(\min(N^2D, ND^2))$

PCA $\rightarrow$ Eigen Decomposition $\rightarrow$ ~~Order~~ Covariance Matrix Order$(ND\min(N,D))$

$\rightarrow$ ~~Eig~~ eigen Decomposition Order(D^3) +

$O(D^3 + ND\min(N,D))$

حال تکلیف را مشخص کنیم ~~سؤال~~ $D \gg N$

$O(N^2D)$ $\cancel{O(ND^2)}$ $\cancel{O(D^3)}$ $=$ SVD

$\cancel{O(ND^2)}$ $O(D^3)$ $\cancel{\text{Of}}$ $=$ Eigen

$\Downarrow$

$O(N^2D) \ll O(D^3) \Rightarrow$ SVD is better !

۲) الف) چرا K-means همگرا می‌شود؟! K-mean سعی می‌کند تابع زیر را min کند.

$$\min \sum_{k=1}^{K} \sum_{x_i \in k} \|x_i - \overset{(\mu_k)}{\eta_k}\|^2$$

این کار با الگوریتم Coordinate Descent انجام می‌شود که گاهی نسبت به متغیر μ_k بهینه می‌شود

و یکبار نسبت به نقاط بهینه می‌شود که نقاط اندیس‌های خود تغییر بکنند و چون هر دو مرحله در حال کاهش ~~[illegible]~~ ~~[illegible]~~

تابع خواهند:

$$\mu_k \leftarrow \frac{1}{|S_k|} \sum x_n \, , \quad S_k \leftarrow \{x_n : \|x_n - x_k\| \leqslant \text{all } \|x_n - \cancel{x}_k\|\}$$

علت دوم: حداکثر K^N روش مختلف دسته‌بندی وجود دارند که به هر کدام clustering گوییم

در هر بار یک کلاستر جدید بر حسب کلاستر قبلی ایجاد می‌شود:

۱) اگر کلاستر قبلی همان جدید است کلاستر بعدی هم همان خواهد بود ← Convergence

۲) اگر کلاستر قبلی متفاوت است با کلاستر بعدی:

چون یک تکرار روی تابعی داریم که تعداد اعضای آن محدود است ~~[illegible]~~

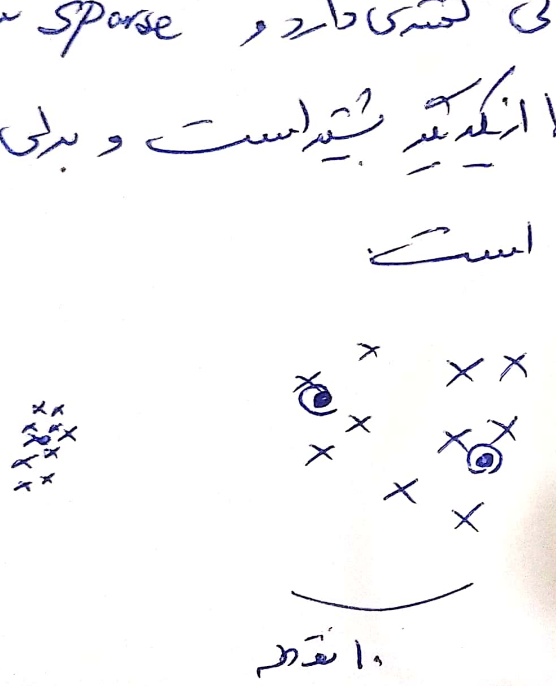
~~[illegible]~~ و هر بار هم تابع کم می‌شود پس در جایی نهایتاً Converge خواهیم کرد.

۲) ب) در واقع در اینجا در طرفی که چگالی کمتری دارد و sparse تر است

تعداد خوشه‌های بیشتری وجود دارد زیرا فاصله نقاط از یکدیگر بیشتر است و برای کمینه کردن

فاصله به نقطه خوشه بیشتری مورد نیاز است

مثلا اگر k=3 باشد داریم



لیست سریالهای فرم سپرده سرمایه گذاری طرح ...

ردیف	از سریال	تا سریال	آخرین سریال
۱	۱۰۶۰۰۱	۱۰۶۰۵۰	۱۰۶۰۴۴

2 ج) این همان الگوریتم k-means++ است که ایده اصلی آن این است که مرکز

اگر نقاط اولیه پخش شوند و فاصله بیشتری داشته باشند بهتر است

سرعت همگرایی: با این که این روش به نظر در ابتدا محاسبات بیشتری را دارد اما الگوریتم

k-means بعد از انتخاب مراکز، این روش خیلی سریع تر همگرا می شوند و در واقع زمان محاسبه اش

خیلی کمتر می شود.

در مورد خطا نیز، خطای این روش خیلی خیلی کمتر از k-means است و جایی که در بعضی جاها

ادعا شده است که خطا تا ۱۰۰۰ برابر کاهش داشته است.

همان ~~بیشتر بالای الگوریتم~~ اما یک ایراد دارد که ممکن است که به بهترین

نقطه نرسیم چون یک init داریم که ممکن است خوب نباشد و در بهینه محلی گیر کنیم

۳ الف الف

0.5	0.5	0
0.5	0.5	0.5
0	0.5	0

init Q-values

0	0	-1
0	0	0
-1	0	1

rewards

البته جدول بهتر بود شکلی کشیده می شد که جهت نیز مشخص بود و تغییری نکرد

فرمول آپدیت Q-value ها:

$$Q(s,a) = Q(s,a)(1-\alpha) + \alpha\left[r(s,a,s') + \gamma \max_{a'} Q(s',a')\right]$$

$\alpha \leftarrow$ پارامتر (learning rate)

$\gamma \leftarrow$ discount factor

$< \gamma < 1$

$$S_{11} \xrightarrow{R} S_{12} \xrightarrow{R} S_{13}$$

$$Q(S_{11}, right) = Q(S_{11}, right)(1-\alpha) + \alpha\left[r(S_{11}, right, S_{12}) + \gamma \max_{a'}(S_{12}, a')\right]$$

$$= 0.5(1-\alpha) + \alpha[0 + 0.5\gamma] = \frac{1}{2}\alpha\gamma - \frac{1}{2}\alpha + \frac{1}{2} = \frac{1}{2} + \overbrace{\frac{1}{2}\alpha(\gamma-1)}^{\text{منفی}} < \frac{1}{2}$$

$$Q(S_{12}, right) = Q(S_{12}, right)(1-\alpha) + \alpha\left[r(S_{12}, right, S_{13}) + \gamma \max(S_{13}, a')\right]$$

$$= 0.5(1-\alpha) + \alpha[-1 + 0 \times \gamma] = \frac{1}{2} - \frac{3}{2}\alpha < \frac{1}{2}$$

۳) الف | ب

$$S_{11} \xrightarrow{R} S_{12} \xrightarrow{R} S_{22} \xrightarrow{D} S_{32} \xrightarrow{R} S_{33}$$

✓ $Q(S_{11}, right) = (1-\alpha)\, Q(S_{11}, right) + \alpha\left[r(S_{11}, right, S_{12}) + \gamma \max_{a'} (S_{12}, a')\right]$

$= (1-\alpha)\left(\frac{1}{2}\alpha\gamma - \frac{1}{2}\alpha + \frac{1}{2}\right) + \alpha\left[0 + 0.5\gamma\right] = \alpha\gamma - \frac{1}{2}\alpha + \frac{1}{2} - \frac{1}{2}\alpha^2\gamma + \frac{1}{2}\alpha^2 - \frac{1}{2}\alpha$

$= -\frac{1}{2}\alpha^2\gamma + \frac{1}{2}\alpha^2 - \alpha + \alpha\gamma + \frac{1}{2}$

✓ $Q(S_{12}, down) = (1-\alpha)\, Q(S_{12}, down) + \alpha\left[r(S_{12}, down, S_{22}) + \gamma \max_{a'} (S_{22}, a')\right]$

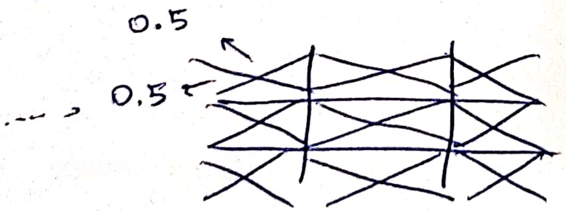
$= (1-\alpha)\, 0.5 + \alpha\left[0 + \gamma \times 0.5\right] = \frac{1}{2}\alpha\gamma + \frac{1}{2}\alpha + \frac{1}{2}$

$Q(S_{22}, down) = (1-\alpha)\, Q(S_{22}, down) + \alpha\left[r(S_{22}, down, S_{32}) + \gamma \max_{a'} (S_{32}, a')\right]$

$= (1-\alpha)\, 0.5 + \alpha\left[0 + \gamma \times 0.5\right] = \frac{1}{2}\alpha\lambda - \frac{1}{2}\alpha + \frac{1}{2}$

$Q(S_{32}, right) = (1-\alpha)\, Q(S_{32}, right) + \alpha\left[r(S_{32}, right, S_{33}) + \gamma \max_{a'} (S_{33}, a')\right]$

$(1-\alpha)\, 0.5 + \alpha\left[1 + \gamma \times 0\right] = \frac{1}{2}\alpha + 1$



جدولی که در ابتدا کشیدیم نیز بهتر بود به این صورت کشیده می‌شد

۳) ب) نکته مهم این است که در سیاست بهینه α تأثیر ندارد.

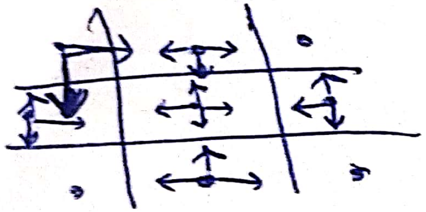
حال اگر بخواهیم از S_{12} به S_{33} برویم ۳ عمل نیاز است که بنابراین روی مقدار $Q(S_{33},0)$ ۳ بار γ اعمال شده است تا به S_{12} برسد پس می‌توان گفت

$$Q_1^*(S_{11},R) \propto \gamma_1^3, \quad Q_2^*(S_{11},R) \propto \gamma_2^3 \xRightarrow{\div} \left(\frac{\gamma_1}{\gamma_2}\right)^3 = \frac{Q_1^*(S_{11},R)}{Q_2^*(S_{11},R)} = \frac{0.9}{0.7} \Rightarrow \frac{\gamma_1}{\gamma_2} = \sqrt[3]{\frac{9}{7}}$$

۳ ب ۴

~~Model based~~

۳ ب ۴ برای یادگیری Q-value ها باید به تعداد شکل مقابل پارامتر یاد بگیریم ،



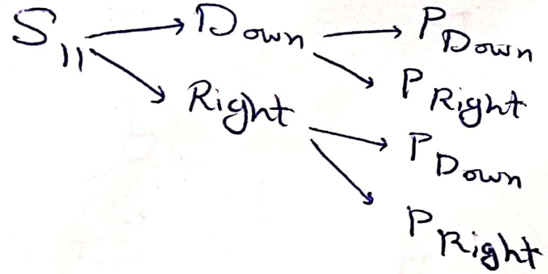
۴ + ۴×۳ + ۲ = ۱۸

که این روش اصطلاحا Model-free گویند چون مدل یاد نگرفتیم و یک سری سیاست با همان Q-value را یاد گرفتیم .

اما اگر از روش model-base استفاده کنیم تعداد پارامترها بیشتر خواهد بود زیرا به ازای هر (state , action) باید احتمال

یاد بگیریم که قطعاً مدل با تعداد پارامترهای بیشتری خواهد شد .

۳ ب ۲



$\underline{4 \times 2}$

یک همچنین generalization ی هم از سوال داریم

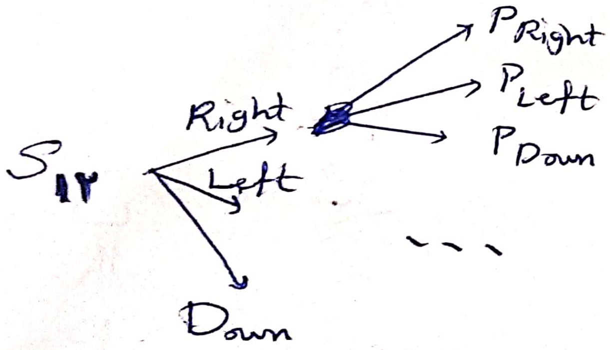
که [illegible] تعداد اکشن های هر حالت به توان ۲

برسد و ضرب در ۲ شود

~~در شکل مشخص است اگر به هر حالتی برویم~~

علت نیز آن است که اگر بخواهیم به چپ مثلا برویم

احتمال دارد به راست یا بالا یا چپ یا پایین برویم!



$9 \times 2 = \underline{18}$

4 تا خونه

$4 \times 2 + (9 \times 2) \times 4 + 16 \times 2 = 8 + 72 + 32 = \boxed{112}$

حال از لحاظ بهتر یا بدتر بودن نیز بستگی به خود مسئله دارد اگر تعداد حالت ها کم باشد روش Model-based بهتر است وگرنه model-free