Final Exam

Jungseul Ok
CSED/AIGS515 Machine Learning
POSTECH

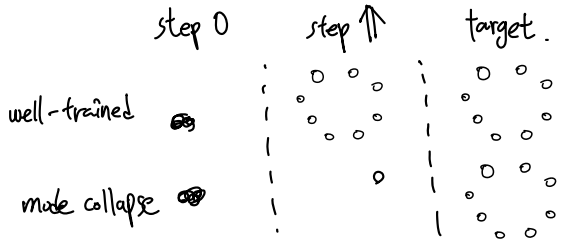
Dec 21, 2021

You have 7 hours in total, from 1 pm to 8 pm. Note that there are 8 problems with different difficulties, and manage your time wisely. The maximum score may not be 100.

- You need to provide proper justification for your answers.
- It may be helpful to understand that I leave sufficient blank space for your answer per problem, while, of course, you can use more if you need.
- You can write your answers in English or Korean.
- You need to upload your *handwritten* answers (e.g., handwriting on tablet/paper) in *a single PDF*, named `studentID_name.pdf` by 8 pm sharp. Note that no excuse will be accepted for any delayed submission, i.e., if you failed at submitting on time, you get 0 score. Any violation in the submission *format* may bring 5% penalty.
- It may be helpful to understand that I leave sufficient blank space for your answers.
- In case that you have questions regarding problems in exam, you can ask them to me in our webex classroom links on PLMS, in scheduled time slots:
  - 1:30pm - 1:40pm
  - 3:30pm - 3:40pm
- It is okay to refer lecture notes and books, or even use the Internet. However, any copy may cause failure of this course. Write what you are truly understanding. In addition, you cannot discuss with others.

Name: 심재윤 JAEYOON SIM

StudentID: 20170243



1. [Short Questions]

(a) [Training GAN; 4pt] With visual illustrations, explain <u>what</u> is mode-collapse in GAN training, and <u>why</u> it is often observed.

Mode-collapse란 GAN을 학습시킬때 발생하는 issue 중 하나이다. 예를들어 visual illustration을 보았을때 cluster가 3개가 존재한다고 가정해볼 것이다. GAN의 구조상 2개의 network를 학습시켜야 하며, 이 중 generator는 3개의 cluster에 대한 clustering 결과를 생성해야 할 것이다. 그러나 discriminator를 속이는데에는 1개의 cluster에 대한 data만 잘생성해도 가능하게 된다. 이 자체로도 real data와 일치하기에 틀린 사실이 아니게 된다. 즉, 특정 소수의 data class에만 충분히 만족스런 결과를 generator가 만들게 되면 discriminator를 속일 수 있고 이러한 현상은 학습의 불균형으로 한쪽이 상대적으로 학습이 잘되어 발생하는 issue이다.

(b) [Boosting; 3pt] Explain <u>what</u> is boosting method, and <u>why</u> it is often used.

Boosting method는 machine learning에서 ensemble 기법 중 하나로, 연속적으로 weak classifier들을 결합하여 분류하는 task 등에서 성능을 높일수 있는 알고리즘이다. 실제로 data를 통해서 model을 만들게 되는데, 이는 새로운 data에 대해서 task를 잘 수행하기를 원한다. 그렇기때문에 overfitting은 방지해야하며, 이를 위한 방법으로 여러 model을 결합해서 장점을 끌어내고자 하는것이 ensemble이 원하는 바 이다. Boosting method의 경우 ensemble의 아이디어에 연속적인 weak classifier 등을 이용해서 이전의 결과를 반영하는 식으로 진행이 된다. 대표적인 알고리즘으로는 AdaBoost 등이 있으며 이 경우에는 weighted majority vote를 가진 weak classifier를 갖으면서 결과적으로 성능을 올리고자 한다.

2. [Bayesian network] Consider the Bayesian network shown as the directed acyclic graph in Figure 1. All the variables are binary of support $\{0, 1\}$.

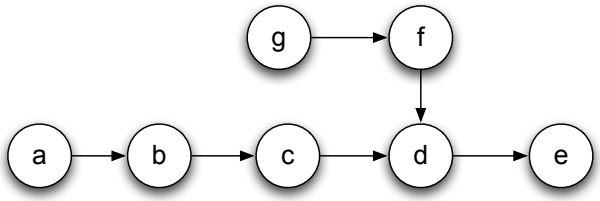
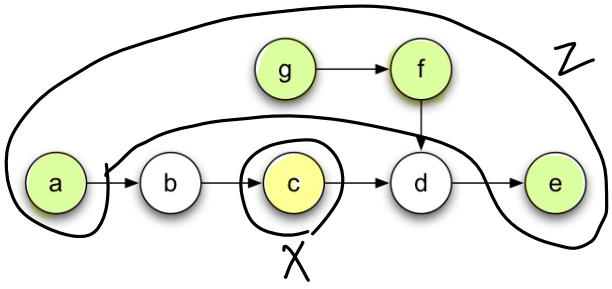


Figure 1: Bayesian network

(a) [2pt] Write the joint probability of $a, b, ..., g$ in the factorized form corresponding to Figure 1.

$p(a,b,c,d,e,f,g) = p(a)p(g)\,p(b|a)\,p(c|b)\,p(f|g)\,p(e|d)\,p(d|c,f)$

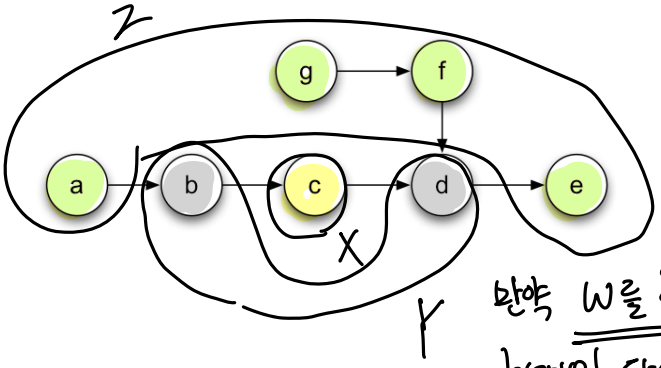
(b) [2pt] Let $X = \{c\}, Y = \{b, d\}, Z = \{a, e, f, g\}$. Is $X \perp Z|Y$? If yes, explain why. If no, show a path from X to Z that is not blocked.

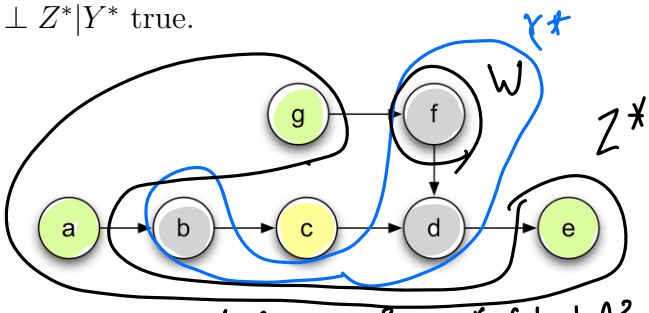


No. $X \not\perp Z|Y$ (By D-seperation)

path 중에서 variable ⓓ의 양옆 ⓒ, ⓕ와 head-to-head 관계에 해당하고 observed 되었기 때문에 conditional independence 하지 않고 dependent 하다. 그래서 X와 Z는 Y가 주어졌을 때 dependent 하게 된다.

(c) [2pt] For a subset $W \subset Z$, define $Z^* = Z \setminus W$ and $Y^* = Y \cup W$. Find the smallest set W which makes $X \perp Z^* | Y^*$ true.

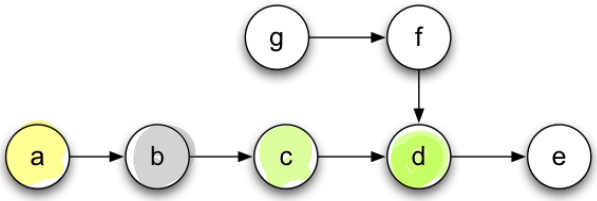




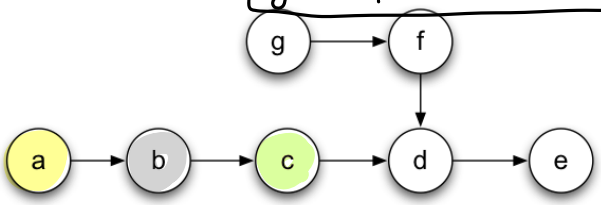
W = {f}

만약 W를 {f}라 한다면, $Z^* = \{a, e, g\}$, $Y^* = \{b, d, f\}$ 가 되어 Y^*가 observed 되었을때 D-seperation에 의해 a-b-c path는 head-to-tail을 만족하고, c-d-e도 d가 head-to-tail로 만족하게 된다. c-d-f-g에서 f가 head-to-tail로 만족.

(d) [2pt] From the graph in Figure 1, we can see that $a \perp c, d | b$. Using probability theory (not graphical inspection), prove that this implies $a \perp c | b$.



$\Rightarrow$



goal: $p(a, c | b) = p(a|c)\, p(b|c)$

$a \perp c, d \mid b$

$$p(a, c, d | b) = \frac{p(a, c, d, b)}{p(b)}$$

$$= \frac{p(a)\, p(b|a)\, p(c, d | b)}{p(b)}$$

$$= p(a|b)\, p(c, d | b)$$

$a \perp c \mid b$

$$p(a, c, d | b) = p(a|b)\, p(c, d | b)$$

$$\sum_d p(a, c, d | b)\, p(d) = p(a|b) \sum_d p(c, d | b)\, p(d)$$

$$p(a, c | b) = p(a|b)\, p(c|b)$$

$\therefore \; a \perp c \mid b$

3. [Belief Propagation] Consider an integer programming (IP) of $x_1, ..., x_5$ with linear objective and constraints in the followings:

$$\begin{aligned} &\underset{x_1,...,x_5 \in \{0,1\}}{\text{maximize}} \quad \sum_{i=1}^{5} x_i \\ &\text{subject to} \quad \sum_{i \in \{1,2\}} x_i \leq 1 \,, \quad \sum_{i \in \{1,4\}} x_i \leq 1 \,, \quad \sum_{i \in \{2,3\}} x_i \leq 1 \,, \\ &\qquad\qquad \sum_{i \in \{3,4\}} x_i \leq 1 \,, \quad \text{and} \quad \sum_{i \in \{4,5\}} x_i \leq 1 \,. \end{aligned}$$

In order to solve the IP, we can formulate a maximum a posterior (MAP) problem of the joint probability of $x_1, ..., x_5$ in the following factorized form:

$$p(x_1, ..., x_5) = \frac{1}{Z} \prod_{i=1}^{5} f_i(x_i) \prod_{I \in \{\{1,2\},\{1,4\},\{2,3\},\{3,4\},\{4,5\}\}} F_I(x_I) \,,$$

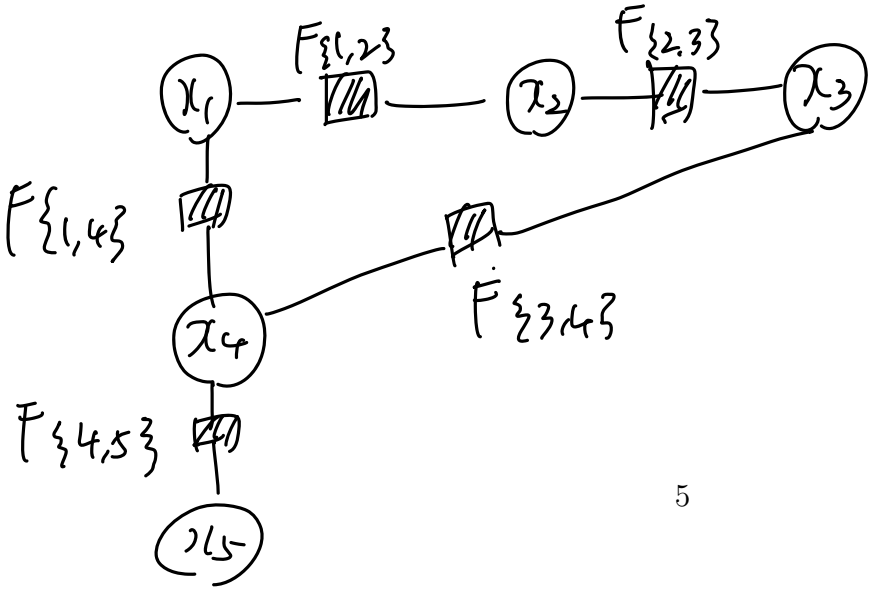
where Z is the normalization constant; for $i \in \{1, 2, 3, 4, 5\}$, $f_i(x_i) = \exp(x_i)$; for $I = \{i, j\}$, $x_I = (x_i, x_j)$ and

$$f_I(x_I) = \begin{cases} 1 & \text{if } x_i + x_j \leq 1 \\ 0 & \text{otherwise} \end{cases} \,.$$

Note that only configuration of $x_1, ..., x_5$ verifies all the constraints in the IP has non-zero probability, which is proportional to the exponential of the objective value of the IP. Hence, the MAP configuration is a solution to the integer programming.

(a) [2pt] Draw the factor graph corresponding to the joint probability $p(x_1, ..., x_5)$.

$\frac{1}{Z} \prod_{i=1} f_i(x_i) = F_{\{1,2\}}(\{1,2\}) F_{\{1,4\}}(\{1,4\}) F_{\{2,3\}}(\{2,3\})$
$F_{\{3,4\}}(\{3,4\}) \quad F_{\{4,5\}}(\{4,5\})$



(b) [5pt] Solve the IP using the max-product belief propagation algorithm. What is the optimal configuration $x^* = (x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)$?

4. [K-means] Given a dataset $\mathcal{D} = \{x^{(i)}\}_{i\in[N]}$ of N data points in $\mathbb{R}^2$, we want to partitioning them into K clusters using K-means algorithm. Let $\mu_k \in \mathbb{R}^2$ denote the center of cluster $k \in [K]$. Then, the hard K-means algorithm aims at optimizing:

$$\min_{\{r_{ik}\geq 0\},\{\mu_k\}} \quad \sum_{i\in[N]}\sum_{k\in[K]} \frac{1}{2} r_{ik}\|x^{(i)} - \mu_k\|_2^2 \tag{1a}$$

$$\text{s.t.} \quad r_{ik} \in \{0,1\} \quad \forall i \in [N], \forall k \in [K] \quad \text{and;} \tag{1b}$$

$$\sum_{k\in[K]} r_{ik} = 1 \quad \forall i \in [N]\ . \tag{1c}$$

For $\beta > 0$, the soft K-means algorithm aims at optimizing:

$$\min_{\{r_{ik}\geq 0\},\{\mu_k\}} \quad \sum_{i\in[N]}\sum_{k\in[K]} \frac{1}{2} r_{ik}\|x^{(i)} - \mu_k\|_2^2 - \beta \sum_{i\in[N]}\sum_{k\in[K]} r_{ik}\log r_{ik} \tag{2a}$$

$$\text{s.t.} \sum_{k\in[K]} r_{ik} = 1 \quad \forall i \in [N]\ . \tag{2b}$$

(a) [3pt] Explain the meaning of $-\sum_{k\in[K]} r_{ik}\log r_{ik}$ in probability theory, and describe when it is minimized.

r_{ik}는 n번째의 datapoint x_n과 cluster k에 속할 확률을 나타내는 indicator이다.
$-\sum_{k\in[K]} r_{ik}\log r_{ik}$는 Entropy로 randomness의 정도를 정량화하게 된다.
그리고 이는 one-hot-vector와 같이 표현이 되는 경우에 그 값을 최소로 가지게 된다.
(r_{ik}가 하나에 대해서 1일 때)

(b) [2pt] Describe the choice of β such that the hard and soft K-means algorithms correspond to each other.

β는 일종의 control knob으로 clustering의 simplicity와 reference point의 representability 사이의 trade-off이다. 여기서 β가 커지게 되면 hard한 K-means algorithm과 같아지게 되고, soft K-means algorithm을 하고자 한다면 β가 작아지기를 원한다.

5. [Expectation-Maximization] I have a pair of coins with unknown probabilities of heads, denoted by $\mu_1 \in (0,1)$ and $\mu_2 \in (0,1)$, respectively. In advance of each coin flip, I randomly choose the first coin with probability $\pi \in (0,1)$, and the second one with probability $1-\pi$. The chosen coin is flipped once and the result is recorded. The outcome of n-th toss is $x_n \in \{0,1\}$ (head = 1, tail = 0). Suppose I tell you the outcomes of the N tosses, denoted by $\boldsymbol{x} = \{x_1, ..., x_N\}$, but I never tell you which coins I used on each toss. Let $\boldsymbol{z} = \{z_1, ..., z_N\}$ be latent variables where $z_n \in \{1,2\}$ denotes which coin was used on n-th toss. Your task is to estimate parameter $\theta = (\mu_1, \mu_2, \pi)$.

(a) [2pt] Write the complete-data log-likelihood $\log p(\boldsymbol{x}, \boldsymbol{z} \mid \theta)$.

$$\mathcal{L}_c(\theta; \{x_n, z_n\}_{n\in[N]}) = \sum_{n\in[N]} \log p(x_n, z_n \mid \theta)$$
$$= \sum_{n\in[N]} \log\{p(x_n \mid z_n, \theta)\, p(z_n, \theta)\}$$
$$= \sum_{n\in[N]} \log\left\{ \prod_{k\in[K]} \left(\pi_k \mu_k^{x_n} (1-\mu_k)^{1-x_n}\right)^{z_{kn}} \right\} \quad \forall k = 1,2$$

continue

(b) [3pt] Defining $\gamma_n = p(z_n = 1 \mid x_n, \theta)$, writhe $Q(\theta'; \theta) = \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x},\theta}[\log p(\boldsymbol{x}, \boldsymbol{z} \mid \theta')]$ in terms of γ_n, $\theta = (\mu_1, \mu_2, \pi)$ and $\theta' = (\mu_1', \mu_2', \pi')$.

$$= \sum_{n\in[N]} \sum_{k\in[K]} z_{kn} \log\left\{\pi_k \mu_k^{x_n} (1-\mu_k)^{1-x_n}\right\}$$

$\gamma_n = p(z_n = k \mid x_n, \theta') = \mathbb{E}_{z_n \mid x_n, \theta'}[z_{kn}]$

$\gamma_n = p(z_n = 1 \mid x_n, \theta)$

$Q(\theta'; \theta) = \mathbb{E}_{z|x,\theta}[\log p(x, z \mid \theta')]$ $\quad \theta = (\mu_1, \mu_2, \pi)$, $\theta' = (\mu_1', \mu_2', \pi')$

$$= \mathbb{E}_{z|x,\theta}\left[\sum_{n=1}^{N} \log p(x_n, z_n \mid \theta')\right]$$
$$= \sum_{n\in[N]} \sum_{k\in[K]} p(z_n = k \mid x_n, \theta) \log p(x_n, z_n \mid \theta')$$

$$= \sum_{n \in [N]} \{ \underline{p(z_n = 1 \mid x_n, \theta)} \log p(x_n, z_n \mid \theta') + p(z_n = 2 \mid x_n, \theta) \log p(x_n, z_n \mid \theta') \}$$

$$= \sum_{n \in [N]} \{ \gamma_n \log p(x_n, z_n \mid \theta') + (1 - \gamma_n) \log p(x_n, z_n \mid \theta') \}$$

$$= \sum_{n \in [N]} \Big[\gamma_n \log \{ \pi' \mu_1'^{x_n} (1 - \mu_1')^{1 - x_n} \} + (1 - \gamma_n) \log \{ (1 - \pi') \mu_2'^{x_n} (1 - \mu_2')^{1 - x_n} \} \Big]$$

$$= \sum_{n \in [N]} \Big[\gamma_n \{ \log \pi' + x_n \log \mu_1' + (1 - x_n) \log (1 - \mu_1') \} + (1 - \gamma_n) \{ \log (1 - \pi') + x_n \log \mu_2' + (1 - x_n) \log (1 - \mu_2') \} \Big]$$

(c) [3pt] Derive the E-step of EM algorithm for the estimation of θ, i.e., compute γ_n.

$$\gamma_n = p(z_n=1 \mid x_n, \theta)$$

$$= \frac{p(z_n=1, x_n \mid \theta)}{p(x_n \mid \theta)}$$

$$= \frac{p(x_n, z_n=1 \mid \theta)}{\sum_{k=1}^{2} p(x_n, z_n=k \mid \theta)}$$

$$= \frac{p(x_n, z_n=1 \mid \theta)}{p(x_n, z_n=1 \mid \theta) + p(x_n, z_n=2 \mid \theta)}$$

$$= \frac{\pi \mu_1^{x_n}(1-\mu_1)^{1-x_n}}{\pi \mu_1^{x_n}(1-\mu_1)^{1-x_n} + (1-\pi)\mu_2^{x_n}(1-\mu_2)^{1-x_n}}$$

(d) [3pt] Derive the M-step of EM algorithm for the estimation of θ.

$$Q(\theta'; \theta) = \sum_{n \in [N]} \gamma_n \left\{ \log \pi' + x_n \log \mu_1' + (1-x_n) \log(1-\mu_1') \right\}$$

$$+ (1-\gamma_n) \left\{ \log(1-\pi') + x_n \log \mu_2' + (1-x_n) \log(1-\mu_2') \right\}$$

θ'에 대해서, 즉 μ_1', μ_2', π'에 대해서 미분을 하고 0으로 두고 parameter를 업데이트 하고자 한다.

① $$\frac{\partial Q}{\partial \pi'} = \sum_{n \in [N]} \left(\gamma_n \frac{1}{\pi'} + (1-\gamma_n) \frac{-1}{1-\pi'} \right)$$

$$= \sum_{n \in [N]} \frac{\gamma_n(1-\pi') - \pi'(1-\gamma_n)}{\pi'(1-\pi')}$$

$$= \sum_{n \in [N]} \frac{\gamma_n - \cancel{\gamma_n \pi'} - \pi' + \cancel{\gamma_n \pi'}}{\pi'(1-\pi')}$$

$$= \sum_{n \in [N]} \frac{\gamma_n - \pi'}{\pi'(1-\pi')} = 0$$

$$\frac{1}{\pi'(1-\pi')} \sum_{n\in[N]} (\gamma_n - \pi') = 0$$

$$\sum_{n\in[N]} \gamma_n - N\cdot\pi' = 0$$

$$\therefore \pi' = \frac{1}{N}\sum_{n\in[N]} \gamma_n$$

② $$\frac{\partial Q}{\partial \mu_1'} = \sum_{n\in[N]} \left\{ \gamma_n \left(\frac{x_n}{\mu_1'} + \frac{x_n - 1}{1-\mu_1'} \right) \right.$$

$$= \sum_{n\in[N]} \frac{\gamma_n \left(x_n(1-\mu_1') + \mu_1'(x_n - 1) \right)}{\mu_1'(1-\mu_1')}$$

$$= \sum_{n\in[N]} \frac{\gamma_n (x_n - \cancel{\mu_1' x_n} \cancel{+ x_n \mu_1'} - \mu_1')}{\mu_1'(1-\mu_1')}$$

$$= \sum_{n\in[N]} \frac{\gamma_n (x_n - \mu_1')}{\mu_1'(1-\mu_1')} = 0$$

$$\frac{1}{\mu_1'(1-\mu_1')} \sum_{n\in[N]} (\gamma_n x_n - \gamma_n \mu_1') = 0$$

$$\sum_{n\in[N]} \gamma_n x_n = \mu_1' \sum_{n\in[N]} \gamma_n$$

$$\therefore \mu_1' = \frac{\sum_{n\in[N]} \gamma_n x_n}{\sum_{n\in[N]} \gamma_n}$$

③ $$\frac{\partial Q}{\partial \mu_2'} = \sum_{n\in[N]} \left\{ (1-\gamma_n) \left(\frac{x_n}{\mu_2'} + \frac{x_n - 1}{1-\mu_2'} \right) \right\}$$

$$= \sum_{n\in[N]} \frac{(1-\gamma_n)\left(x_n(1-\mu_2') + \mu_2'(x_n-1) \right)}{\mu_2'(1-\mu_2')} = 0$$

$$\sum_{n\in[N]} (1-\gamma_n)\left(x_n - \cancel{\mu_2' x_n} + \cancel{\mu_2' x_n} - \mu_2 \right) = 0$$

$$\sum_{n\in[N]} (1-\gamma_n) x_n = \mu_2' \sum_{n\in[N]} (1-\gamma_n)$$

$$\therefore \mu_2' = \frac{\sum_{n\in[N]} (1-\gamma_n) x_n}{\sum_{n\in[N]} (1-\gamma_n)}$$

(e) [3pt] Let $Q(\theta';\theta) = \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x},\theta}[\log p(\boldsymbol{x},\boldsymbol{z} \mid \theta')]$, where the expectation is taken w.r.t. the conditional distribution of $\boldsymbol{z}$ given $\boldsymbol{x}$ and θ. Starting from writing $Q(\theta';\theta) - Q(\theta;\theta)$, show that maximizing $Q(\theta';\theta)$ w.r.t. $\theta' = (\mu_1', \mu_2', \pi')$ improves $p(\boldsymbol{x} \mid \theta')$ over $p(\boldsymbol{x} \mid \theta)$. (Hint: in the proof, you may need to use the non-negativity of Kullback-Leibler divergence, i.e., $\mathrm{KL}(p(\boldsymbol{x})\|q(\boldsymbol{x})) = \sum_{\boldsymbol{x}} \log p(\boldsymbol{x}) \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \geq 0$ for any distributions p and q.)

$$Q(\theta';\theta) = \mathbb{E}_{z|x,\theta}\left[\log p(x,z|\theta')\right]$$

$$Q(\theta';\theta) - Q(\theta;\theta)$$

$$= \mathbb{E}_{z|x,\theta}\left[\log p(x,z|\theta')\right] - \mathbb{E}_{z|x,\theta}\left[\log p(x,z|\theta)\right]$$

$$= \sum_{n\in[N]}\Big[\gamma_n\left\{\log\pi' + x_n\log\mu_1' + (1-x_n)\log(1-\mu_1')\right\} + (1-\gamma_n)\left\{\log(1-\pi') + x_n\log\mu_2' + (1-x_n)\log(1-\mu_2')\right\}\Big]$$

$$- \sum_{n\in[N]}\Big[\gamma_n\left\{\log\pi + x_n\log\mu_1 + (1-x_n)\log(1-\mu_1)\right\} + (1-\gamma_n)\left\{\log(1-\pi) + x_n\log\mu_2 + (1-x_n)\log(1-\mu_2)\right\}\Big]$$

$$= \sum_{n\in[N]}\Big[\gamma_n\log\left\{\pi'\mu_1'^{x_n}(1-\mu_1')^{1-x_n}\right\} + (1-\gamma_n)\log\left\{(1-\pi')\mu_2'^{x_n}(1-\mu_2')^{1-x_n}\right\}\Big]$$

$$- \sum_{n\in[N]}\Big[\gamma_n\log\left\{\pi\mu_1^{x_n}(1-\mu_1)^{1-x_n}\right\} + (1-\gamma_n)\log\left\{(1-\pi)\mu_2^{x_n}(1-\mu_2)^{1-x_n}\right\}\Big]$$

$$= \sum_{n \in [N]} \left[\gamma_n \log \left(\frac{\pi' \mu_1'^{x_n} (1-\mu_1')^{1-x_n}}{\pi \mu_1^{x_n} (1-\mu_1)^{1-x_n}} \right) \right.$$

$$\left. + (1-\gamma_n) \log \left(\frac{(1-\pi') \mu_2'^{x_n} (1-\mu_2')^{1-x_n}}{(1-\pi) \mu_2^{x_n} (1-\mu_2)^{1-x_n}} \right) \right]$$

$$\geq 0$$

$$KL(p(x) \| q(x)) = \sum_x \log p(x) \frac{p(x)}{q(x)} \geq 0$$

6. [PCA] Consider the conventional principle component analysis (PCA) for a high-dimensional dataset $\{x_n\}_{n=1,\dots,N}$, where each data point $x_n \in \mathbb{R}^D$. Let $\bar{x} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^{N} x_n$. Define X to be the $N \times D$ dimensional centered data matrix, of which the n-th row is given by $(x_n - \bar{x})^\top$ so that the data covariance matrix $S \in \mathbb{R}^{D\times D}$ is defined as:

$$S \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^\top = \frac{1}{N}X^\top X \ .$$

The conventional PCA, providing dimensionality reduction from D to $M < D$, extracts the orthonormal eigenvectors $\{u_i\}_{i=1,\dots,M}$ corresponding to the M largest eigenvalues $\{\lambda_i\}_{i=1,\dots,M}$ of the data covariance matrix S.

(a) [4pt] Write an optimization problem which yields the conventional PCA as solution, and discuss implications of the optimization.

Convential PCA는 2가지로 해석이 가능하다. 첫번째는 variance를 최대로하는것이고, 두번째는 error를 최소화 하는 것이다. 첫번째 관점으로 variance를 최대로 하면 projected dataset의 정보를 최대로하는것과 같다. Covariance matrix S에 대해서 optimization problem을 나타내기위해서 projected matrix $U=[u_1, \cdots u_M]$이라고 할때, 먼저 Largragian method를 이용하면 $\mathcal{L}(u_1,\lambda_1) = u_1^T S u_1 + \lambda_1(1-u_1^T u_1)$ 으로 하나에 대해 나타낼수 있고 KKT stationary condition을 통해서 $\frac{d\mathcal{L}}{du_1}=0 \Rightarrow Su_1=\lambda_1 u_1$ 임이 자명하다. 이를 M차원에서는 같이 orthonormal column vector condition을 $u_i^T u_j = \delta_{ij} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{if } i\neq j \end{cases}$ 가 되어 optimization problem을 나타내면

$$\text{maximize} \quad \sum_{i\in[M]} u_i^T S u_i$$

$$\text{subject to} \quad u_i^T u_j = \delta_{ij} \quad \forall i,j \in [M]$$

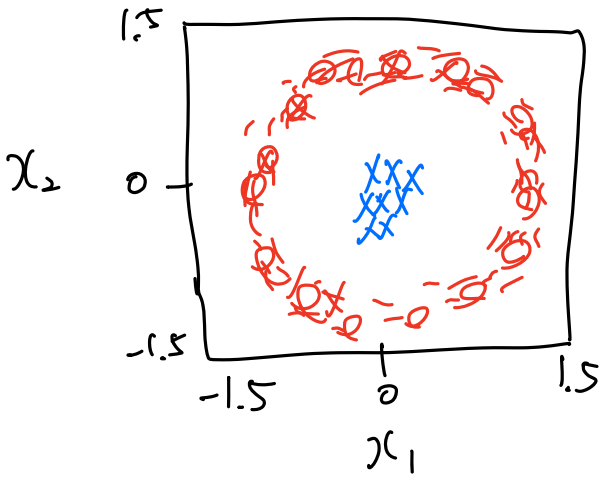
이 된다. 각 u_i는 data covariance matrix의 eigenvector에 해당하고 objective는 eigen value의 해당하는 summation이다. 결론 이는 M차원에서의 PCA를 M개의 가장 큰 eigenvalue에 대응하는 data covariance matrix의 eigen vector를 계산하는것과 같다.

(b) [5pt] Suppose $N \ll D$. A direct implementation of the conventional PCA requires the eigendecomposition of a $D \times D$ matrix, of which computational cost is $O(D^3)$. Propose and justify a way to reduce the computational cost for the eigendecomposition from $O(D^3)$ to $O(N^3)$.

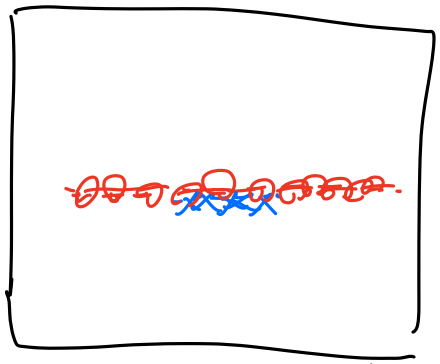
우리는 PCA를 통해서 X^TX의 eigen vector를 찾아야 한다. 차원을 줄이는 trick을 쓸것이고, data covariance matrix에 해당하는 eigenvector 식은 $\frac{1}{N} X^T X u_i = \lambda_i u_i$ 가되어 (여기서 $X \in R^{N \times D}$ matrix, n-th row는 $(x_n - \bar{x})^T$) 여기에 X를 곱하면 $\frac{1}{N} X X^T X u_i = \lambda X u_i$ 가된다. 여기서 우리는 Xu_i를 따로 불러지킬것이고 $\frac{1}{N}$과 λ는 scalar이다. 이렇게 식을 trick을써서 바꾸면 Xu_i를 쉽게 찾을수있다. 예제 $XX^T \in R^{N \times N}$ 이 되어 computational tractable 하게 된다. 원래 처음 식은 DXD 행렬이었는데 식을 바꿔서 u_i 대신 Xu_i를 찾음으로 computational cost가 줄어들게 된다. 그래서 $Xu_i = v_i$로 치환해서 $\frac{1}{N} XX^T(v_i) = \lambda_i(v_i)$ 를 구할것이고 이로부터 원래의 u_i를 구할수있다. 기존의 $O(D^3)$에서 $O(N^3)$로 줄어들게 되면서 여기 여기에 X^T를 곱해서 $\frac{1}{N} X^T X X^T v_i = \lambda_i X^T v_i$ 가되어 $X^T v_i$ 구해서 u_i를 구하게된다. 즉, XX^T의 eigenvector v_i를 얻고 normalized eigenvector u_i를 v_i에 대해서 구하면 우리는 $u_i = \frac{1}{\sqrt{N\lambda_i}} X^T v_i$ 로부터 u_i를 computational cost를 줄여서 구할수 있게 된다.

(c) [3pt] Provide your own example which motivates us to use the kernel PCA instead of the conventional PCA, and explain your choice of kernel.

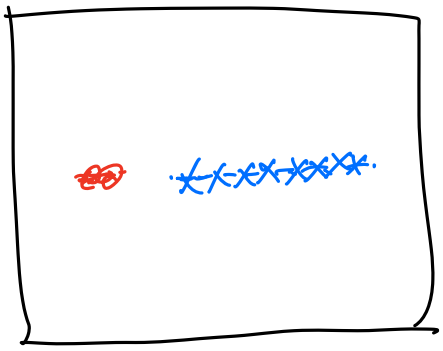
Data가 3차원과 같이 분포되어 있을 때 2차원이나 1차원으로 projection 시키게 되는데 conventional PCA를 사용하면 특정 방향으로 data를 projection 시킬때에 구별하기 어려워지는 현상이 나타난다. 이를 해결하고자 kernel PCA (KPCA)를 통해서 다른 plane 이나 space에 projection 시키고자 한다.



이와 같이 2종류의 data distribution이 존재한다고 가정하자. KPCA를 이용해 2차원의 data에 대한 mapping function을 $\phi(x_1, x_2) = (x_1, x_2, \sqrt{x_1^2 + x_2^2})$ 과 같이 norm을 추가해서 정의하게 되면 feature extractor를 사용해서 차원을 도입하고 여기에 통상한 conventional PCA를 적용하게 되면 아주 쉽게 분리될 것이다. 이는 추가적인 feature space에서 효과적인 정보 압축이 가능해진 것이다.



conventional PCA



kernel PCA

7. [Generative Adversarial Networks] Consider the following max-min problem for a dataset $\mathcal{D}$ consisting of x's:

$$\max_{\phi} \min_{w} -\sum_{x\in\mathcal{D}} \log p_w(y=1 \mid x) - \sum_{z\in\mathcal{Z}} \log(1-p_w(y=1 \mid G_\phi(z))) + \frac{C}{2}\|w\|_2^2 \,. \qquad (3)$$

Here the generator $G_\phi(z)$ parameterized by ϕ transforms noise $z \in \mathcal{Z}$ into artificial data. The discriminator $p_w(y \mid x)$ parameterized by w checks if x is artificial or not, where $y=1$ indicates that x is real, and $y=0$ indicates that x is artificial. The hyper-parameter $C \geq 0$ controls impact of regularization. Note that solving (3) is challenging mainly due to the objective is neither convex in w nor concave in ϕ in general. We will check if the cost function is is convex in w for specific choice of the discriminator model. To do so, we use several facts:

Fact1. A function $f(w)$ is convex in w if Hessian[1] of $f(w)$ is positive semi-definite[2].

Fact2. A sum of convex functions is also convex.

Fact3. You can use http://www.matrixcalculus.org/ for matrix calculus.

(a) [1pt] Suppose that we model the discriminator as follows:

$$p_w(y=1 \mid x) = \frac{1}{1+\exp(w^\top x)} \,.$$

Using this, write down the resulting cost function for (3).

$$-\sum_{x\in D} \log p_w(y=1|x) - \sum_{x\in Z} \log\left(1-p_w(y=1|G_\phi(z))\right) + \frac{C}{2}\|w\|_2^2$$

$$= -\sum_{x\in D} \log \frac{1}{1+\exp(w^T x)} - \sum_{x\in Z} \log\left(1-\frac{1}{1+\exp(w^T G_\phi(z))}\right) + \frac{C}{2}\|w\|_2^2$$

$$= \sum_{x\in D} \log\left(1+\exp(w^T x)\right) - \sum_{x\in Z} \log\left(\frac{\exp(w^T G_\phi(z))}{1+\exp(w^T G_\phi(z))}\right) + \frac{C}{2}\|w\|_2^2$$

$$= \sum_{x\in D} \log\left(1+\exp(w^T x)\right) - \sum_{x\in Z} \left\{\log\left(\exp(w^T G_\phi(z))\right) - \log\left(1+\exp(w^T G_\phi(z))\right)\right\} + \frac{C}{2}\|w\|_2^2$$

$$= \sum_{x\in D} \log\left(1+\exp(w^T x)\right) - \sum_{x\in Z} \left\{w^T G_\phi(z) - \log\left(1+\exp(w^T G_\phi(z))\right)\right\} + \frac{C}{2}\|w\|_2^2$$

$$= \sum_{x\in D} \log\left(1+\exp(w^T x)\right) + \sum_{x\in Z} \left\{\log\left(1+\exp(w^T G_\phi(z))\right) - w^T G_\phi(z)\right\} + \frac{C}{2}\|w\|_2^2$$

[1] https://en.wikipedia.org/wiki/Hessian_matrix
[2] https://en.wikipedia.org/wiki/Definite_symmetric_matrix

(b) [1pt] Obtain Hessian of (A) $= \frac{C}{2}\|w\|_2^2 - w^\top b$ in w. Check if (A) is convex, and justify your answer.

$(A) = \frac{C}{2}\|w\|_2^2 - w^T b$

$\frac{\partial (A)}{\partial w} = C\|w\|_2 - b$

$\frac{\partial^2 (A)}{\partial w \partial w^T} = C \cdot I$ = Hessian of (A)

$C \geq 0$으로부터 Hessian of (A)가 positive semi-definite하고 이계도함수가 양수로 갖는 $\bigcup$ 집합은 convex하다. (By Fact 1)

(c) [1pt] Obtain Hessian of (B) $= \log(1 + \exp(w^\top b))$ in w. Check if (B) is convex, and justify your answer.

$(B) = \log(1+\exp(w^T b))$

$\frac{\partial (B)}{\partial w} = \frac{\exp(w^T b)\cdot b^T}{1+\exp(w^T b)}$

$\frac{\partial^2 (B)}{\partial w \partial w^T} = \frac{b\exp(w^T b)\cdot b^T \cdot (1+\exp(w^T b)) - \exp(w^T b)\cdot b \exp(w^T b)\cdot b^T}{(1+\exp(w^T b))^2}$

$= \frac{\exp(w^T b)\, b\cdot b^T \cdot (1+\exp(w^T b) - \exp(w^T b))}{(1+\exp(w^T b))^2}$

$= \frac{\exp(w^T b)\cdot b b^T}{(1+\exp(w^T b))^2}$

$(1+\exp(w^T b))^2 \geq 0$

$\exp(w^T b) \geq 1$

$x^T b\, b^T x = (b^T x)^T (b^T x) = (b^T x)^2 \geq 0$

positive semi-definite → convex (By Fact 1)

(d) [1pt] Check if the cost function obtained in Problem 7a is convex, and justify your answer.

$\sum_{x\in D} \underbrace{\log(1+\exp(w^T x))}_{(b)} + \underbrace{\sum_{x\in Z}\{\log(1+\exp(w^T G\phi(z))) - w^T G\phi(z)\}}_{①} + \underbrace{\frac{C}{2}\|w\|_2^2}_{(c)}$

(b)와 (c)로부터 convex함을 확인했고, ①도 convex하므로 Fact 2에 의해 convex 함수들의 합은 다시 convex이므로 7a의 cost function 또한 convex하다.

(e) [1pt] Introducing auxiliary variables $\xi_x = w^\top x$ and $\xi_z = w^\top G_\phi(z)$, consider the following optimization (for the discriminator):

$$\min_w \quad \sum_{x\in\mathcal{D}} \log(1+\exp\xi_x) + \sum_{z\in\mathcal{Z}} \log(1+\exp(\xi_z)) - \sum_{z\in\mathcal{Z}} w^\top G_\phi(z) + \frac{C}{2}\|w\|_2^2 \quad (4a)$$

$$\text{s.t.} \quad \xi_x = w^\top x \quad \forall x \in \mathcal{D} \quad (4b)$$

$$\xi_z = w^\top G_\phi(z) \quad \forall z \in \mathcal{D} \quad (4c)$$

Write the Lagrangian for this optimization, where λ_x and λ_z are Lagrange multipliers corresponding to (4b) and (4c), resp.

$$\sum_{x\in\mathcal{D}} \log(1+\exp\xi_x) + \sum_{z\in\mathcal{Z}} \{\log(1+\exp\xi_x) - \sum_{z\in\mathcal{Z}} w^\top G_\phi(z) + \frac{C}{2}\|w\|^2$$

$$-\sum_{x\in\mathcal{D}} \lambda_x (w^\top x - \xi_x) - \sum_{z\in\mathcal{Z}} \lambda_z (w^\top G_\phi(z) - \xi_z)$$

(f) [1pt] Obtain the value of

$$\min_w \frac{C}{2}\|w\|_2^2 - w^\top b$$

in terms of b and $C \geq 0$.

$$\min_w \frac{C}{2}\|w\|_2^2 - w^\top b \quad \text{subject to} \quad b \geq 0, \; C \geq 0.$$

$$\mathcal{L} = \frac{C}{2}\|w\|_2^2 - w^\top b - \lambda b - \lambda' C \qquad \lambda \text{ and } \lambda' : \text{dual variable.}$$

$$\frac{\partial \mathcal{L}}{\partial w} = C w^\top - b^\top = 0 \quad \rightarrow \quad w = \frac{b}{C}$$

$$\therefore \quad \frac{C}{2}\cdot\frac{b^2}{C^2} - w^\top b = \frac{\|b\|_2^2}{2C} - \frac{\|b\|_2^2}{C}$$

$$= -\frac{\|b\|_2^2}{2C}$$

(g) [1pt] Obtain the value of

$$\min_{\xi} \quad \lambda\xi + \log(1 + \exp\xi)$$

in terms of λ assuming $-1 \le \lambda \le 0$.

$\min_{\xi} \lambda\xi + \log(1+\exp\xi)$ $\quad \mathcal{L} = \lambda\xi + \log(1+\exp\xi) + \alpha(-1-\lambda) + \beta\lambda$

subject to $\lambda \ge -1$, $\lambda \le 0$ $\quad \frac{\partial \mathcal{L}}{\partial \xi} = \lambda + \frac{\exp\xi}{1+\exp\xi} = 0$

$\frac{e^{\xi}}{1+e^{\xi}} = -\lambda$

continue

(h) [2pt] Combining Problems 7e, 7f, and 7g and using $H(a) = a\log(-a) - (1+a)\log(1+a)$, obtain dual function $g(\lambda)$ for (4). For training the discriminator, we can replace the original minimization over w described in (3) with the dual maximization over valid values of λ. Using this, write down an alternative of GAN training in (3), in which we have a max-max problem instead of the max-min problem. Note that such an alternative training in max-max form can help to bypass challenges from fining a saddle-point, i.e., solving the max-min problem.

$e^{\xi} = X$

$\frac{X}{1+X} = -\lambda$

$X = -\lambda - \lambda X$

$X(1+\lambda) = -\lambda$

$X = \frac{-\lambda}{1+\lambda} = e^{\xi}$

$\xi = \log\left(\frac{-\lambda}{1+\lambda}\right)$

$\therefore \lambda\log\left(\frac{-\lambda}{1+\lambda}\right) + \log\left(1+\frac{-\lambda}{1+\lambda}\right)$

$= \lambda\{\log(-\lambda) - \log(1+\lambda)\} - \log(1+\lambda)$

$= \lambda\log(-\lambda) - \lambda\log(1+\lambda) - \log(1+\lambda)$

(h) $H(a) = a\log(-a) - (1+a)\log(1+a)$

$H(\lambda) = \lambda\log(-\lambda) - (1+\lambda)\log(1+\lambda)$

$$\min_{w} \quad \sum_{x\in\mathcal{D}} \log(1+\exp\xi_x) + \sum_{z\in\mathcal{Z}} \log(1+\exp(\xi_z)) - \sum_{z\in\mathcal{Z}} w^\top G_\phi(z) + \frac{C}{2}\|w\|_2^2 \quad (4a)$$

$$\text{s.t.} \quad \xi_x = w^\top x \quad \forall x \in \mathcal{D} \quad (4b)$$

$$\xi_z = w^\top G_\phi(z) \quad \forall z \in \mathcal{D} \quad (4c)$$

from 9e)

$$\sum_{x\in D} \log(1+\exp\xi_x) + \sum_{z\in Z}\{\log(1+\exp\xi_x) - \sum_{z\in Z} w^\top G_\phi(z) + \frac{C}{2}\|w\|^2$$

$$-\sum_{x\in D} \lambda_x(w^\top x - \xi_x) - \sum_{z\in Z} \lambda_z(w^\top G_\phi(z) - \xi_z)$$

$$= \sum_{x\in D}\{\log(1+\exp\xi_x) + \lambda_x\xi_x\} + \sum_{z\in Z}\{\log(1+\exp\xi_x) + \lambda_z\xi_z\}$$

$$-\sum_{x\in D}\lambda_x w^\top x - \sum_{z\in Z}\lambda_z w^\top G_\phi(z) - \sum_{z\in Z} w^\top G_\phi(z) + \frac{C}{2}\|w\|^2$$

$$= \sum_{x\in D}\{\log(1+\exp\xi_x) + \lambda_x\xi_x\} + \sum_{z\in Z}\{\log(1+\exp\xi_x) + \lambda_z\xi_z\} \quad \text{9g}$$

$$-w^\top\left[\sum_{x\in D}\lambda_x x + \sum_{z\in Z}\{\lambda_z G_\phi(z) + G_\phi(z)\}\right] + \frac{C}{2}\|w\|_2^2 \quad \text{9f}$$

from $w_{min} = -\frac{\|b\|_2^2}{2C}$, $\xi_{min} = \lambda\log(-\lambda) - \lambda\log(1+\lambda) - \log(1+\lambda)$

$= H(\lambda)$

$$\therefore g(\lambda) = -\frac{\left\| \sum_{x \in D} \lambda_x x + \sum_{z \in Z} \{\lambda_z G_\phi(z) + G_\psi(z)\} \right\|_2^2}{2C} + \sum_{x \in D} H(\lambda_x) + \sum_{z \in Z} H(\lambda_z)$$

$g(\lambda)$ 가 max-max problem 에 대한 alternative traing GAN 은 다음과 같다.

$$\therefore \max_{\phi} \max_{\lambda} g(\lambda) \quad \text{s.t.} \quad -1 \le \lambda_i \le 0, \ \forall_i = \{x, z\}$$

(i) [1pt] Complete `GAN.py`, which is an implementation of the alternative training of GAN obtained in Problem 7h with the $\log D$ trick in the lecture. You may handwrite what you added in below. (Hint: use `target1` and `target2`)

Task 1 >

⋮

```
logit = disc(torch.cat((data, xhat.detach()), 0))
loss = criterion(logit, target1)
```

⋮

Task 2 >

⋮

```
xhat = gen(z)
logit = disc(xhat)
loss = criterion(logit, target2)
```

⋮

8. [Q-learning] Consider an MDP with state space $\mathcal{S} = \{1, 2, 3\}$ and action space $\mathcal{A} = \{-, +\}$, where state 3 is only terminal state. We evaluate policy π using discounted value $v_\pi(s) := \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \gamma^t R_{t+1} \mid S_0 = s \right]$ where $\gamma = 0.5$ is discount factor and T is terminating time, i.e., $S_T = 3$. For this MDP, we will perform Q-learning algorithm.

   (a) [1 pt] Algorithm 1 is a pseudo code of Q-learning algorithm, in which one typo exists. Find and fix the typo.

Algorithm 1 Q-learning

Initialize $Q(s, a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$
repeat (for each episode)
    Initialize $S = 1$
    repeat (for each step of episode)
        Choose A from S using some behavior policy
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + 0.5[R + \max_{a \in \mathcal{A}} Q(S', a) - Q(S, A)]$
        $S \leftarrow S'$
    until S is terminal, i.e., $S = 3$
until Q converges

$Q(S,A) \leftarrow Q(S,A) + 0.5[R + \gamma \max_{a \in A} Q(S',a) - Q(S,A)]$

$Q(S,A) \leftarrow Q(S,A) + 0.5[R + \gamma \max_{a \in A} Q(S',a) - Q(S,A)]$

(b) [2 pt] Suppose that in the first episode, we observe the sequence of state transitions and rewards in Table 1. With your correction in Problem 8a, compute $Q(\cdot,\cdot)$ after the first episode.

S_0	A_0	R_1	S_1	A_1	R_2	S_2
1	+	**1**	2	+	**-1**	3

Table 1: The first episode

	−	+
1	0	
2	0	
3	0	0

初始 $Q(\cdot,\cdot)=0$

Table 2: $Q(\cdot,\cdot)$ after the first episode in Table 1

$$Q(1,+) = 0 + 0.5[1 + 0.5 \max\{0,0\} - 0]$$

$$= 0.5(1 + 0 - 0)$$

$$= 0.5$$

$$Q(2,+) = 0 + 0.5[-1 + 0.5 \max\{0.5,0\} - 0]$$

$$= 0.5(-1 + 0 - 0)$$

$$= -0.5$$

(c) [3 pt] Suppose that in the second episode, we observe the sequence of state transitions and rewards in Table 3. With your correction in Problem 8a, compute $Q(\cdot, \cdot)$ after the second episode.

S_0	A_0	R_1	S_1	A_1	R_2	S_2	A_2	R_3	S_3	A_3	R_4	S_4	A_4	R_5	S_5
1	$-$	**0**	1	$+$	**1**	2	$-$	**0**	1	$+$	**1**	2	$+$	**-1**	3

Table 3: The second episode

first episode

	$-$	$+$
1	0	0.5
2	0	-0.5
3	0	0

0.125 → 0.75 → 0.921875 ; -0.75 ; 0.1875

Table 4: $Q(\cdot, \cdot)$ after the second episode in Table 3

$$Q(1,-) = 0 + 0.5\left[0 + 0.5 \max\{0, 0.5\} - 0\right] = 0.125$$

$$Q(1,+) = 0.5 + 0.5\left[1 + 0.5 \max\{0, -0.5\} - 0.5\right] = 0.75$$

$$Q(2,-) = 0 + 0.5\left[0 + 0.5 \max\{0.125, 0.75\} - 0\right] = 0.1875$$

$$Q(1,+) = 0.75 + 0.5\left[1 + 0.5 \max\{0.1875, -0.75\} - 0.75\right] = 0.921875$$

$$Q(2,+) = -0.5 + 0.5\left[-1 + 0.5 \max\{0, 0\} + 0.5\right] = -0.75$$

(d) [1 pt] Given Table 3, what are the greedy actions at state 1 and state 2?

	$-$	$+$
1	0.125	0.921875
2	0.1875	-0.75
3	0	0

State 1 → $+$

State 2 → $-$

(e) [2 pt] Considering the accumulated discounted rewards with discount factor $\gamma \in [0, 1]$ as value function, provide an example of MDP, of which optimal policy changes depending on the choice of γ. Note that you need to justify your answer. (I will choose one for the courses in next year.)

바둑게임

agent : 본인 컴퓨터

environment : 상대방

state : 바둑알 배치

action : 바둑알 다음위치

reward : 이기면 +1, 지면 -1

discount factor $\gamma \in [0,1]$

MDP가 가능한 이유는 state와 action이 주어졌을 때 확률적이지만 다음 미래는 현재 state와 action으로 설명할 수 있다.

This is the end of the exam. Happy holidays ☺