Machine Learning Techniques HW 6

R09946006 | 何青儒 | HO Ching-Ru | Jan 14th, 2021

---

1. [**b**] 36
    - By the definition of $\delta_j^{(l)} = \sum_{k=1}^{d^{(l+1)}} \left(w_{jk}^{(l+1)} \delta_k^{(l+1)}\right) \cdot \left(\tanh'(s_j^{(l)})\right)$:
    - For Layer 1: $\delta_1^{(l)} = \sum_{k=1}^{d^{(1+1)}=6} \left(\delta_k^{(2)} w_{jk}^{(2)}\right) \cdot \left(\tanh'(s_j^{(1)})\right)$, $j \in \{1, d^{(1)}\} = \{1, 5\}$, and each has $(d^{(2)} = 6)$ operators. The number of operators is $(5 \times 6 = 30)$.
    - For Layer 2: $\delta_1^{(2)} = \sum_{k=1}^{d^{(2+1)}=1} \left(\delta_k^{(3)} w_{jk}^{(3)}\right) \cdot \left(\tanh'(s_j^{(2)})\right)$, $j \in \{1, d^{(2)}\} = \{1, 6\}$, and each has $(d^{(3)} = 1)$ operators. The number of operators is $(6 \times 1 = 6)$.
    - Summing up above, the total number of operators is 36.
2. [**d**] 1219
    - By the definition, the number of layer k's weights can be represented as $\left(d^{(k)} \cdot d^{(k+1)}\right)$. And the number of **total** weights equals to summing up each layer's (form layer 1 to layer $L-1$) number of weights, where is $\sum_{k=1}^{L-1} \left(d^{(k)} + 1\right) \cdot d^{(k+1)}$.
    - $$\begin{aligned} \blacktriangle(L) &= \sum_{k=1}^{L-1} \left(d^{(k)} + 1\right) \cdot d^{(k+1)} \\ &= \left(d^{(0)} + 1\right) \cdot d^{(1)} + \left(d^{(1)} + 1\right) \cdot d^{(2)} + \cdots + \left(d^{(L-1)} + 1\right) \cdot d^{(L)} \cdot \\ &= (20) \cdot d^{(1)} + \left(d^{(1)} + 1\right) \cdot d^{(2)} + \cdots + \left(d^{(L-1)} + 1\right) \cdot 3 \end{aligned}$$
    - On the other hand, we know there has 50 hidden layer, where $\sum_{k=1}^{L-1} \left(d^{(k)} + 1\right) = 50$, so $\blacksquare(L) = \sum_{k=1}^{L-1} \left(d^{(k)}\right) = 50 - (L-1)$.
    - If $L = 2$:
        - $\blacksquare(2) = d^{(1)} = 49$.
        - $\blacktriangle(2) = 20 \cdot d^{(1)} + \left(d^{(1)} + 1\right) \cdot 3 = 1130$, it is one of possible answer.
    - If $L = 3$:
        - $\blacksquare(3) = d^{(1)} + d^{(2)} = 48$.
        - $\blacktriangle(3) = 20 \cdot d^{(1)} + \left(d^{(1)} + 1\right) \cdot d^{(2)} + \left(d^{(2)} + 1\right) \cdot 3 = -(d^{(1)})^2 + 64d^{(1)} + 195$.
        - Try to find the value by first order differential, $-2d^{(1)} + 64 = 0$, $d^{(1)} = 32$.
        - Use $d^{(1)} = 32$, we know $\blacktriangle(3) = 1219$, it is one of possible answer.
    - It's hard to find the global maximum when $L \geq 4$.
3. [**d**] $q_k - v_k$

- $$\begin{aligned}\delta_k^{(L)} &= \frac{\partial \text{err}(x,y)}{\partial s_k^{(L)}} \\ &= \frac{\partial}{\partial s_k^{(L)}}(-v_k \ln q_k) \\ &= -[\![y=k]\!] \cdot \frac{\partial}{\partial s_k^{(L)}}\left(\ln \frac{\exp(s_k^{(L)})}{\sum_{k=1}^{K}\exp(s_k^{(L)})}\right) \\ &= -[\![y=k]\!] \cdot \left(\frac{\sum_{k=1}^{K}\exp(s_k^{(L)})}{\exp(s_k^{(L)})} \cdot \frac{\exp(s_k^{(L)})\sum_{k=1}^{K}\exp(s_k^{(L)}) - \exp(s_k^{(L)})\exp(s_k^{(L)})}{\left(\sum_{k=1}^{K}\exp(s_k^{(L)})\right)^2}\right) \\ &= -[\![y=k]\!] + \frac{\exp(s_k^{(L)})}{\sum_{k=1}^{K}\exp(s_k^{(L)})} \\ &= -v_k + q_k\end{aligned}.$$

4. [**a**] 0
   - By the gradient descent updating function: $w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta x_i^{(l-1)}\delta_j^{(l)}$, let $\eta = 1, i = 0, j = 1$, $l = 1$, $x_1^{(0)} = 1$, the update function becomes: $w_{01}^{(1)} \leftarrow w_{01}^{(1)} - \eta x_1^{(0)}\delta_1^{(1)} = w_{01}^{(1)} - \delta_1^{(1)}$.
   - Due to $\mathbf{w}^{(l)} = 0$ (initialization), implies that $w_{i1}^{(1)} = 0$, $\forall i$. Let $l = 1, j = 1$. Next, computes $\delta_1^{(1)}$:
   - $$\begin{aligned}\delta_1^{(1)} &= \sum_{k=1}^{d(1+1)} \left(\delta_k^{(1+1)}\right) \cdot \left(w_{ik}^{(1+1)}\right)\left(\cdot \tanh'(s_1^{(1)})\right) \\ &= \sum_{k=1}^{d(2)} \left(\delta_k^{(2)}\right) \cdot \left(w_{ik}^{(2)}\right) \cdot \underbrace{\left(\tanh'\left(\sum_{i=1}^{d(0)=4} w_{i1}^{(1)} x_i^{(0)}\right)\right)}_{=0} \\ &= 0\end{aligned}.$$
   - During the first update, $w_{01}^{(1)} \leftarrow w_{01}^{(1)} - \delta_1^{(1)} = 0 - \delta_1^{(1)} = -\delta_1^{(1)} = 0$.
   - During the second update, $w_{01}^{(1)} \leftarrow w_{01}^{(1)} - \delta_1^{(1)} = -\delta_1^{(1)} - \delta_1^{(1)} = -2\delta_1^{(1)} = 0$.
   - During the third update, $w_{01}^{(1)} \leftarrow w_{01}^{(1)} - \delta_1^{(1)} = -2\delta_1^{(1)} - \delta_1^{(1)} = -3\delta_1^{(1)} = 0$.
5. [**e**] half the average rating of the m-th movie
   - Follow the steps on *page 10 of Lecture 215*, we can optimize the $\mathbf{w}$ by minimizing the E_{in}.
   - $$\begin{aligned}\frac{\partial}{\partial w_m}\left(\min_{\mathbf{W},\mathbf{V}} E_{\text{in}}(\{\mathbf{w}_m\},\{\mathbf{v}_n\})\right) &\propto \frac{\partial}{\partial w_m}\left(\sum_{m=1}^{M}\left(\sum_{(\mathbf{x}_n, r_{nm})\in\mathcal{D}_m}\left(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n\right)^2\right)\right) \\ &= \frac{\partial}{\partial w_m}\left(\sum_{(\mathbf{x}_n, r_{nm})\in\mathcal{D}_m}\left(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n\right)^2\right) \\ &= -2\cdot\mathbf{v}_n\cdot\sum_{(\mathbf{x}_n, r_{nm})\in\mathcal{D}_m}\left(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n\right) = 0\end{aligned}.$$
   - The result implies that only when $\mathbf{v}_n \cdot \sum_{(\mathbf{x}_n, r_{nm})\in\mathcal{D}_m}\left(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n\right) = 0$, the E_{in} can be minimized.
   - Next, we have known that $\mathbf{v} = [\mathbf{v}_1\,\mathbf{v}_2\cdots\mathbf{v}_N]_{\tilde{d}\times N} = [2\,2\cdots 2]_{1\times N}$, which means $\mathbf{v}_n = 2$, $\forall n$.
   - $$\begin{aligned}&\mathbf{v}_n \sum_{(\mathbf{x}_n, r_{nm})\in\mathcal{D}_m}\left(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n\right) = 2\sum_{(\mathbf{x}_n, r_{nm})\in\mathcal{D}_m}\left(r_{nm} - 2\mathbf{w}_m^T\right) = 0 \\ &\longrightarrow \sum_{n=1}^{N}\left(r_{nm} - 2\mathbf{w}_m^T\right) = \sum_{n=1}^{N}(r_{nm}) - 2N\mathbf{w}_m^T = 0 \\ &\longrightarrow \mathbf{w}_m = \frac{1}{2}\left(\frac{1}{N}\sum_{n=1}^{N}(r_{nm})\right)\end{aligned}.$$
   - In fact, in this problem, $\mathbf{w}_m$ is a $\tilde{d} \times 1 = 1 \times 1$ dimension vector, which can be represented as w_m, as a scalar. Therefore, for the m-th movie, w_m is a half ($\frac{1}{2}$) the average ($\frac{1}{N}$) rating of the m-th movie ($\sum_{n=1}^{N}(r_{nm})$).
6. [**b**] $a_m \leftarrow (1-\eta)a_m + \eta\cdot(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n - b_n)$

- $$\begin{aligned}\frac{\partial \mathrm{err}}{\partial a_m} &= \frac{\partial}{\partial a_m}\left(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n - a_m - b_n\right)^2 \\ &= -2\left(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n - a_m - b_n\right)\end{aligned}.$$
- Then follow the update function of SGD:
- $$\begin{aligned} a_m \leftarrow & a_m - (\text{larning rate}) \cdot \mathrm{err} \\ = & a_m + \left(\frac{\eta}{2}\right) \cdot 2\left(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n - a_m - b_n\right) \\ = & a_m + \eta\left(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n - a_m - b_n\right) \\ = & (1-\eta)\cdot a_m + \eta \cdot \left(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n - b_n\right)\end{aligned}.$$

7. [**d**] $[0.16, 0.08, 0.24]$
   - Follow the definition of G on *page 7 of Lecture 207*,
     $G(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^{T=3} g_t(\mathbf{x})\right) = \operatorname{sign}\Big(g_1(\mathbf{x}) + g_2(\mathbf{x}) + g_3(\mathbf{x})\Big)$.
   - To make a "incorrect classification" in a 3-binary-classifier model, at least 2 classifiers need to make an error, i.e., while the real label is $+1$, but $G(\mathbf{x})$ is -1, meaning at least two of (g_1, g_2, g_3) needs to output -1. Which means the $E_{\text{out}}(G) \leq \frac{2}{3+1}\sum_{i=1}^{3} E_{\text{out}}(g_i)$.
   - [**a**] $E_{\text{out}}(G) \leq \frac{2}{3+1}\sum_{i=1}^{3} E_{\text{out}}(g_i) = \frac{1}{2}(0.04 + 0.16 + 0.16) = 0.18$, impossible.
   - [**b**] $E_{\text{out}}(G) \leq \frac{2}{3+1}\sum_{i=1}^{3} E_{\text{out}}(g_i) = \frac{1}{2}(0.04 + 0.08 + 0.24) = 0.18$, impossible.
   - [**c**] $E_{\text{out}}(G) \leq \frac{2}{3+1}\sum_{i=1}^{3} E_{\text{out}}(g_i) = \frac{1}{2}(0.06 + 0.04 + 0.16) = 0.18$, impossible.
   - [**d**] $E_{\text{out}}(G) \leq \frac{2}{3+1}\sum_{i=1}^{3} E_{\text{out}}(g_i) = \frac{1}{2}(0.16 + 0.08 + 0.24) = 0.24$, possible.
   - [**a**] $E_{\text{out}}(G) \leq \frac{2}{3+1}\sum_{i=1}^{3} E_{\text{out}}(g_i) = \frac{1}{2}(0.04 + 0.06 + 0.24) = 0.17$, impossible.
8. [**c**] 0.32
   - To make a "incorrect classification" in a 5-binary-classifier model, at least 3 classifiers need to make an error, i.e., while the real label is $+1$, but $G(\mathbf{x})$ is -1, meaning at least 3 of $(g_1, g_2, g_3, g_4, g_5)$ needs to output -1.
   - $$E_{\text{out}} \leq \binom{5}{3}(0.4)^3(0.6)^2 + \binom{5}{4}(0.4)^4(0.6)^1 + \binom{5}{5}(0.4)^5 = 0.234 + 0.0768 + 0.01024 = 0.32103$$
     , select [**c**] 0.32.
9. [**b**] 60.7%
   - When **an example is not sampled**, the probability is $\frac{N-1}{N}$.
   - When **two example is not sampled**, the probability is
     $\left(\frac{N-1}{N}\right)\cdot\left(\frac{N-1}{N}\right) = \left(\frac{N-1}{N}\right)^2$.
   - When **0.5N example is not sampled**, the probability is
     $$\underbrace{\left(\frac{N-1}{N}\right)\cdots\left(\frac{N-1}{N}\right)}_{0.5N} = \left(\frac{N-1}{N}\right)^{0.5N}.$$
   - When N is very large, $\left(\frac{N-1}{N}\right)^{0.5N} = \left(1 - \frac{1}{N}\right)^{0.5N} = e^{-0.5} \sim 0.60653066 \sim 60.7\%$, as $N \to \infty$.
10. [**e**] none of the other choices
    - If $\mathbf{x} = \mathbf{x}'$, any θ will make both classify into the same direction (classification), where $(\phi(\mathbf{x})^T)(\phi(\mathbf{x}')) = 1 \cdot 1 = (-1)\cdot(-1) = 1$. For any L and R ($L < R$), there have $(R - L + 1)$ even integers between (including) $2L$ and $2R$, and $(R - L)$ odd integers, which means there has $(R - L)$ candidates θ. Each candidate θ can create 2 decision stump: for the negative direction and the positive direction. Thus, $K_{ds}(\mathbf{x}, \mathbf{x}') = 2d \times (R - L)$ when $\mathbf{x} \neq \mathbf{x}'$.

- If $\mathbf{x} \neq \mathbf{x}^{'}$, assume that $L = 0,\ R = 4$, and $\mathbf{x} = 2,\ \mathbf{x}^{'} = 4$ (in this dimension), $\|\mathbf{x} - \mathbf{x}^{'}\|_1 = 2$, there has 2 decision stump ($\theta = 3$, for positive and negative direction); if $\mathbf{x} = 2,\ \mathbf{x}^{'} = 6$, $\|\mathbf{x} - \mathbf{x}^{'}\|_1 = 4$, there has 4 decision stump ($\theta = 3 \vee 5$, and for each has positive and negative direction). From the example, we can know when the $l - 1$ norm distance increase 2, the number of decision stump will also increase 2, where the average of number of decision stump is $\frac{\|\mathbf{x} - \mathbf{x}^{'}\|_1}{2} \times 2 \cdot$
- When $\mathbf{x} \neq \mathbf{x}^{'}$, they are in the different classification. where $(\phi(\mathbf{x})^T)(\phi(\mathbf{x}^{'})) = 1 \cdot (-1) = (-1) \cdot 1 = (-1)$. The value of $(\phi(\mathbf{x})^T)(\phi(\mathbf{x}^{'}))$ from 1 becomes -1, need to minus 2. Thus, $K_{ds}(\mathbf{x}, \mathbf{x}^{'})$ should be $2d(R - L) - 2\|\mathbf{x} - \mathbf{x}^{'}\|_1$. None of the choices is correct.

11. [**a**] 19
    - To find the optimal re-weighting, let:
    - $$\begin{aligned}\epsilon_u &= \frac{\sum_{n=1}^N u_n^{(1)} [\![y_n \neq g_1(\mathbf{x}_n)]\!]}{\underbrace{\sum_{n=1}^N u_n^{(1)}}_{=1}} \\ &= \sum_{n=1}^N u_n^{(1)} [\![y_n \neq -1]\!] \\ &= \frac{(1 + 1 + \cdots + 1)}{N} \\ &= \frac{0.05N}{N} \text{ (5\% of the samples are positive, making the incorrect example)} \\ &= 0.05\end{aligned}.$$
    - Thus, $\blacklozenge_u = \sqrt{(1 - \epsilon_u)/\epsilon_u} = \sqrt{(1 - 0.05)/0.05} = \sqrt{19}$.
    - For the incorrect examples ($[\![(y_t = 1) \neq (g_1(\mathbf{x}_n) = -1)]\!]$), $u_+^{(t+1)} = u_+^t \cdot \blacklozenge_u$; and for the correct example ($[\![(y_t = -1) = (g_t(\mathbf{x}_n) = 1)]\!]$), $u_-^{(t+1)} = u_-^t / \blacklozenge_u$.
    - When $t = 2$, $\frac{u_+^{(2)}}{u_-^{(2)}} = \frac{u_+^{(1)} \cdot \blacklozenge_u}{u_-^{(2)} / \blacklozenge_u} = \blacklozenge_u^2 = 19$.
12. [**d**] $E_{\text{in}}(G_T) \leq \exp(-2T(\frac{1}{2} - \epsilon)^2)$
    - Follow the definition on *page 14 of Lecture 208*, for the incorrect examples ($[\![y_n \neq g_t(\mathbf{x}_n)]\!]$), $u_n^{(t+1)} = u_n^t \cdot \blacklozenge_t$, and for the correct example ($[\![y_n = g_t(\mathbf{x}_n)]\!]$), $u_n^{(t+1)} = u_n^t / \blacklozenge_t$, where $\blacklozenge_t = \sqrt{(1 - \epsilon_t)/\epsilon_t}$.
    - Compute $\frac{U_{t+1}}{U_t}$ by the definition as problem mentioned:
    - $$\begin{aligned}\frac{U_{t+1}}{U_t} &= \frac{\sum_{n=1}^N u_n^{(t+1)}}{\sum_{n=1}^N u_n^{(t)}} \\ &= \frac{\sum_{n=1}^N u_n^{(t+1)} [\![y_n = g_t(\mathbf{x}_n)]\!] + \sum_{n=1}^N u_n^{(t+1)} [\![y_n \neq g_t(\mathbf{x}_n)]\!]}{\sum_{n=1}^N u_n^{(t)} [\![y_n = g_t(\mathbf{x}_n)]\!] + \sum_{n=1}^N u_n^{(t)} [\![y_n \neq g_t(\mathbf{x}_n)]\!]} \\ &= \frac{\sum_{n=1}^N (u_n^{(t)} / \blacklozenge_t) [\![y_n = g_t(\mathbf{x}_n)]\!] + \sum_{n=1}^N (u_n^{(t)} \cdot \blacklozenge_t) [\![y_n \neq g_t(\mathbf{x}_n)]\!]}{\sum_{n=1}^N u_n^{(t)} [\![y_n = g_t(\mathbf{x}_n)]\!] + \sum_{n=1}^N u_n^{(t)} [\![y_n \neq g_t(\mathbf{x}_n)]\!]} \\ &= \frac{\sum_{n=1}^N (u_n^{(t)} / \sqrt{(1 - \epsilon_t)/\epsilon_t}) [\![y_n = g_t(\mathbf{x}_n)]\!] + \sum_{n=1}^N (u_n^{(t)} \cdot \sqrt{(1 - \epsilon_t)/\epsilon_t}) [\![y_n \neq g_t(\mathbf{x}_n)]\!]}{\sum_{n=1}^N u_n^{(t)} [\![y_n = g_t(\mathbf{x}_n)]\!] + \sum_{n=1}^N u_n^{(t)} [\![y_n \neq g_t(\mathbf{x}_n)]\!]} \\ &= \sqrt{(1 - \epsilon_t)/\epsilon_t} \cdot \epsilon_t + \sqrt{\epsilon_t/(1 - \epsilon_t)} \cdot (1 - \epsilon_t) \\ &= 2\sqrt{\epsilon_t(1 - \epsilon_t)} \\ &\leq 2\sqrt{\epsilon(1 - \epsilon)} \\ &\leq 2 \cdot \frac{1}{2}\exp(-2(\frac{1}{2} - \epsilon)^2) = \exp(-2(\frac{1}{2} - \epsilon)^2 \text{ (from the Hint)}\end{aligned}.$$
    - We can rewrite the result into:

$$\frac{U_{t+1}}{U_t} \leq \exp(-2(\frac{1}{2}-\epsilon)^2)$$
$$\longrightarrow U_{t+1} \leq U_t \cdot (\exp(-2(\frac{1}{2}-\epsilon)^2)$$
$$\longrightarrow U_{T+1} \leq U_T \cdot (\exp(-2(\frac{1}{2}-\epsilon)^2)$$
$$\longrightarrow U_{T+1} \leq U_{T-1} \cdot (\exp(-2(\frac{1}{2}-\epsilon)^2) \cdot (\exp(-2(\frac{1}{2}-\epsilon)^2)$$
$$\vdots \text{...iteration...}$$

- $$\longrightarrow U_{T+1} \leq U_1 \cdot \underbrace{(\exp(-2(\frac{1}{2}-\epsilon)^2) \cdots (\exp(-2(\frac{1}{2}-\epsilon)^2)}_{T}$$.

$$\longrightarrow U_{T+1} \leq U_1 \cdot (\exp(-2(\frac{1}{2}-\epsilon)^2)^T$$
$$\longrightarrow U_{T+1} \leq \frac{1}{N}\sum_{n=1}^{N} \exp\underbrace{\left(-y_n \sum_{\tau=1}^{t=0} \alpha_\tau g_\tau(\mathbf{x}_n)\right)}_{=0} \cdot (\exp(-2(\frac{1}{2}-\epsilon)^2)^T$$
$$\longrightarrow E_{\text{in}}(G_T) \leq U_{T+1} \leq (\exp(-2T(\frac{1}{2}-\epsilon)^2)$$

13. [**d**] the closeness, which is $1-|\mu_+ - \mu_-|$.
    - Normalized classification error is $\dfrac{\min(\mu_+,\mu_-)}{1/2} = 2\min(\mu_+,\mu_-)$, the maximum value is $2 \times \dfrac{1}{2} = 1$, when $\mu_+ = \mu_- = \frac{1}{2}$. Substitute $\mu_- = 1-\mu_+$ into each choice:
    - [**a**] The Gini index: $1-\mu_+^2-(1-\mu_+)^2 = -2\mu_+^2+2\mu_+$. Solve $\frac{\partial(-2\mu_+^2+2\mu_+)}{\partial\mu_+} = 0$, $\mu_+^* = \frac{1}{2}$. The maximum value is $\frac{1}{2}$. Normalized form is $\dfrac{-2\mu_+^2+2\mu_+}{1/2} = -4\mu_+^2+4\mu_+$.
    - [**b**] The squared error: $\begin{aligned}&\mu_+(1-(\mu_+-(1-\mu_+)))^2+(1-\mu_+)(-1-(\mu_+-(1-\mu_+)))\\ =&\mu_+(-2\mu_++2)^2+(1-\mu_+)(-2\mu_+)^2\\ =&-4\mu_+^2+4\mu_+\end{aligned}$.
      Solve the F.O.D. equals to 0, $\mu_+^* = \frac{1}{2}$. The maximal value 1. Normalized form is $-4\mu_+^2+4\mu_+$.
    - [**c**] The entropy: $-\mu_+\ln\mu_+ - (1-\mu_+)\ln(1-\mu_+)$. Solve the F.O.D. equals to 0, $\mu_+^* = \frac{1}{2}$. The maximal value $= \ln 2$. Normalized form is $\dfrac{-\mu_+\ln\mu_+ - (\mu_-)\ln\mu_-}{\ln 2}$.
    - [**d**] The closeness: $1-|\mu_+-\mu_-| = 1-|\mu_+-(1-\mu_+)|$. The maximal value $=1$, when $\mu_+^* = \mu_-^* = \frac{1}{2}$. Normalized form is $1-|\mu_+-\mu_-|$.
        - If $\mu_+ < 0.5$, $1-|\mu_+-\mu_-| = 1-|2\mu_+-1| = 1-(1-2\mu_+) = 2\mu_+$.
        - If $\mu_- \geq 0.5$,
          $1-|\mu_+-\mu_-| = 1-|2\mu_+-1| = 1-(2\mu_+-1) = 2-2\mu_+ = 2(1-\mu_+) = 2\mu_-$.
        - Thus, it can imply $1-|\mu_+-\mu_-| = 2\min(\mu_+,\mu_-)$.
14. [**c**] 0.18
15. [**d**] 0.28
16. [**a**] 0.01
17. [**d**] 0.16
18. [**b**] 0.07
19. [**e**] neural networks and deep learning
    - First reason, they are similar to my research topic. Other reason, in my opinion, deep learning is like a magic, we have few idea about its "black box" mechanism, but sometimes it just work very well. This makes me feel that I'm a magician when I create a model to train my dataset, and really get some interesting output results.
20. [**b**] matrix factorization
    - I think matrix factorization is a boring process. Though I know that the subject of "matrix" has strong power and really be a nice tool for studying and research, no matter for mathematics or computer science, but I'm just lack of interesting on it :(.