學號：B10705025　系級：資管三 姓名：彭鈞道

模型連結: public 最佳的

https://drive.google.com/file/d/1mwe0PmRFDft3bKcQu4_tL22wU4QwhWIo/view?usp=sharing

裡面包含 4 個檔案，model.pth 是 lstm 和 classifier 的模型，model, model.syn1neg.npy, model.wv.vectors.npy 則是 word2vec model.

1. (1%) 請以 block diagram 或是文字或者繪圖的方式說明這次表現最好的 model 使用哪些 layer module(如 Conv/RNN/Linear 和各類 normalization layer) 及連接方式(如一般 forward 或是使用 skip/residual connection)，並概念性逐項說明選用該 layer module 的理由以及設計想法。

   採用最簡單的雙層 Linear(兩層因為有較好的表現)，用 forward 連接，使的最後的輸出只有一個，並使用 sigmod()讓其映射到 0~1 之間，當進入 nn.Linear()曾之前會先做 batch normalization 以避免 overfitting，輸出後則會經過 activation function。最一開始有 dropout layer 同樣來避免 overfitting。

```
self.classifier = torch.nn.Sequential(nn.Dropout(0.2),
                                      nn.BatchNorm1d(hidden_dim),
                                      nn.Linear(hidden_dim, 48),
                                      nn.LeakyReLU(0.2),
                                      # nn.Dropout(0.2),
                                      nn.BatchNorm1d(48),
                                      nn.Linear(48, 1),
                                      torch.nn.Sigmoid())
```

2. (1%) 請比較 word2vec embedding layer 初始設為 non-trainable/trainable 的差別，列上兩者在 validation/public private testing 的結果，並嘗試在訓練過程中設置一策略改變 non-trainable/trainable 設定，描述自己判斷改變設定的機制以及該結果。

   發現在 embedding 時讓 requires_grad = True，ie, trainable 時有較好的表現，原本有試過從 epoch 數來決定 trainable 還是 non-trainable，沒有比一開始就設為 true 好，所以就只有直接設為 true 後來用。

```
    # if ( epoch > 10):
    #     backbone.embedding.weight.requires_grad = False
```

pred_with_batchsize 128 inner layer 48 has normalize using no label as word2vec traina... Complete · 2d ago	0.8289	0.8254	☐
pred_with_batchsize 128 inner layer 48 has normalize using no label as word2vec with ... Complete · 3d ago	0.8261	0.8254	☐
pred_with_batchsize 128 inner layer 48 has normalize using no label as word2vec.csv Complete · 3d ago	0.8266	0.8228	☐

由上到下分別是設為 trainable 不改變，策略改變 non-trainable/trainable，初始設為 non-trainable

3. (1%) 請敘述你如何對文字資料進行前處理，並概念性的描述你在資料中觀察到什麼因此你決定採用這些處理，並描述使用這些處理時作細節，以及比較其實際結果，該結果可以不用具備真正改進。如果你沒有作任何處理，請給出一段具體描述來說服我們為什麼不做處理可以得到好的結果，這個理由不能只是單純因為表現比較好。

    資料前處理時首先把所有的換行符號都去掉，所有字都轉成小寫，避免出現在不同位置時被判斷成不同的字。再把標點符號去掉(因為字的意思跟相對位置應該就能代表標點符號)。原本還想做 stemming 跟 stopword removal，但是觀察訓練資料後發現蠻多句子不太符合文法，做這些法而會留下奇怪的字，所以就沒做了。

pred (45).csv Complete · 7d ago	0.7968	0.7953	
pred (44).csv Complete · 10d ago	0.6991	0.702	

    可以發現做前處理後表現有顯著的提升。

4. (1%) 請「自行設計」兩句具有相同單字但擺放位置不同的語句，使得你表現最好的模型產生出不同的預測結果，例如 "Today is hot, but I am happy" 與 "I am happy, but today is hot"，並討論造成差異的原因，但請不要用範例句子。

    兩個句字分別是"I am right,but you are wrong", "I am wrong, but you are right"，不同的原因應是 model 已經知道要以 but 後面的句子判斷，但是 "you are wrong" 跟 "you are right" 是剛好相反的句子，所以預測結果不同。

```
   id                       text
0   1  I am right, but you are wrong
1   2  I am wrong, but you are right
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:557: UserWarning: This Da
  warnings.warn(_create_warning_msg(
```

```python
saved_header.eval()
saved_backbone.eval()
with torch.no_grad():
    for i, (idx_list, lengths, texts) in enumerate(check_
        lengths, inputs = lengths.to(device), texts.to(device)
        if not saved_backbone is None:
            inputs = saved_backbone(inputs)
        soft_predicted = saved_header(inputs, lengths)
        hard_predicted = (soft_predicted >= 0.5).int()
        for i, p in zip(idx_list, hard_predicted):
            print(str(i.item()), str(p.item()))
```

```
device: device
torch.device instance
```

```
1 0
2 1
```

Math1.1

hw4 1、

by what taught in class, E-step

$$Q(\theta|\theta^{(t)}) = \sum_{Z} p(Z|x;\theta^{(t)}) \log p(X,Z,\theta)$$

$$= \sum_{i=1}^{N} \mathbb{E}_{z_i|x_i;\theta^{(t)}}\left[\log p(x_i,z_i;\theta)\right]$$

$$P[z_i=k \mid x_i;\theta^{(t)}] = \frac{\pi_k^{(t)} \prod_{j=1}^{D} \mu_{kj}^{(t)\,x_i^{(j)}} (1-\mu_{kj}^{(t)})^{(1-x_i^{(j)})}}{\sum_{m=1}^{K} \pi_m^{(t)} \prod_{j=1}^{D} \mu_{kj}^{(t)\,x_i^{(j)}} (1-\mu_{kj}^{(t)})^{(1-x_i^{(j)})}}$$

$$= \delta_{ik}^{(t)}$$

$$\log p(x_i, z_i=k;\theta) = \ln\pi_k + \sum_{j=1}^{D}\left(x_i^{(j)} \ln\mu_{kj} + (1-x_i^{(j)})\ln(1-\mu_{kj})\right)$$

$$Q(\theta|\theta^{(t)}) = \sum_{i=1}^{N}\sum_{k=1}^{K} \delta_{ik}^{(t)}\left[\ln\pi_k + \sum_{j=1}^{D}\left(x_i^{(j)}\ln\mu_{kj} + (1-x_i^{(j)})\ln(1-\mu_{kj})\right)\right]$$

M-step $\theta^{(t+1)} = \underset{\theta}{\arg\max}\, Q(\theta|\theta^{(t)})$

$$\nabla_{\mu_{kj}} Q(\theta|\theta^{(t)}) = \sum_{i=1}^{N}\delta_{ik}^{(t)}\left(\frac{x_i^{(j)}}{\mu_{kj}} + \frac{x_i^{(j)}-1}{(1-\mu_{kj})}\right)$$

$$= \sum_{i=1}^{N}\delta_{ik}^{(t)}\left(\frac{x_i^{(j)} - x_i^{(j)}\mu_{kj} + x_i^{(j)}\mu_{kj} - \mu_{kj}}{\mu_{kj}(1-\mu_{kj})}\right)$$

$$= \sum_{i=1}^{N}\delta_{ik}^{(t)}\left(\frac{x_i^{(j)} - \mu_{kj}}{\mu_{kj}(1-\mu_{kj})}\right) = 0$$

$$\Rightarrow \mu_{kj} = \frac{\sum_{i=1}^{N}\delta_{ik}^{(t)} x_i^{(j)}}{\sum_{i=1}^{N}\delta_{ik}^{(t)}}$$

w4.

1. $\nabla_{\pi} Q(\theta|\theta^{(t)})$

Math 1.1, 2.a

$$\Rightarrow \mu_{kj} = \frac{\sum_{i=1}^{N} \delta_{ik}^{(t)} x_i^{(j)}}{\sum_{i=1}^{N} \delta_{ik}^{(t)}}$$

hw4.

1.

$$\nabla_{\pi_k}\left(Q(\theta|\theta^{(t)}) - \lambda \sum_{k=1}^{K} \pi_k\right)$$

$$= \sum_{i=1}^{N} \frac{\delta_{ik}^{(t)}}{\pi_k} - \lambda = 0 \Rightarrow \pi_k^{(t+1)} = \frac{\sum_{i=1}^{N} \delta_{ik}^{(t)}}{\lambda}$$

and $\sum_{k=1}^{K} \pi_k = 1 \Rightarrow \lambda = \sum_{i=1}^{N} \sum_{k=1}^{K} \delta_{ik}^{(t)} = N \Rightarrow \pi_k^{(t+1)} = \frac{\sum_{i=1}^{N} \delta_{ik}^{(t)}}{N}$

hw4.

2. by E-step

(a)

$$Q(\theta|\theta^{(t)}) = \sum_{i=1}^{N} \mathbb{E}_{z_i|x_i;\theta^{(t)}} [\log p(x_i, z_i; \theta)]$$

$$\log p(x_i, z_i = k; \theta) = \ln \pi_k + \ln \frac{1}{T_k} + \frac{-x_i}{T_k}$$

$$Q(\theta|\theta t) = \sum_{i=1}^{N} \sum_{k=1}^{K} \delta_{ik}^{(t)} \left(\ln \pi_k + \ln \frac{1}{T_k} + \frac{-x_i}{T_k}\right)$$

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta|\theta^{(t)})$$

$$\nabla_{T_k} Q(\theta|\theta t) = \sum_{i=1}^{N} \delta_{ik}^{(t)} \left(-\frac{1}{T_k} + \frac{x_i}{T_k^2}\right)$$

$$= \sum_{i=1}^{N} \delta_{ik}^{(t)} \left(\frac{x_i - T_k}{T_k^2}\right) = 0$$

$$\sum_{i=1}^{N} \delta_{ik}^{(t)} (x_i - T_k) = 0$$

$$T_k^{(t+1)} = \frac{\sum_{i=1}^{N} \delta_{ik}^{(t)} x_i}{\sum_{i=1}^{N} \delta_{ik}^{(t)}}$$

$$\nabla_{\pi_k}\left(Q(\theta|\theta^{(t)}) - \lambda \sum_{k=1}^{K} \pi_k\right) = \sum_{i=1}^{N} \frac{\delta_{ik}^{(t)}}{\pi_k} - \lambda = 0$$

$$\Rightarrow \pi_k^{(t+1)} = \frac{\sum_{i=1}^{N} \delta_{ik}^{(t)}}{\lambda}, \quad \lambda = \sum_{i=1}^{N} \sum_{k=1}^{K} \delta_{ik}^{(t)} = N \Rightarrow \pi_k^{(t+1)} = \frac{\sum_{i=1}^{N} \delta_{ik}^{(t)}}{N}$$

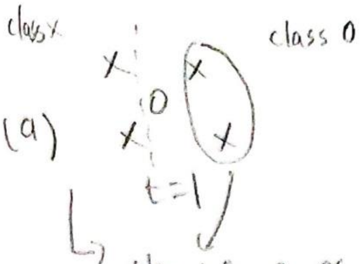
Math 2.b

hw4. 2-(b) $\delta_{ik}^{(t)} = \dfrac{\pi_k\left(\frac{1}{T_k} e^{\frac{-x_i}{T_k}}\right)}{\sum_{j=1}^{K} \pi_j \left(\frac{1}{T_k} e^{\frac{-x_i}{T_k}}\right)}$

Math 3

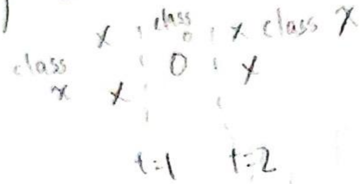
hw4 3.1

(a)



the left 2 x are classified wrong, so their weight increased

(b) 2 iterations



3.1 recall training

by what we taught in class

training data error rate $\leq \prod_{t=1}^{T} 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$

$\varepsilon_t = 1 - \gamma_t$

$\prod_{t=1}^{T} 2\sqrt{\varepsilon_t(1-\varepsilon_t)} = \prod_{t=1}^{T} \sqrt{1-4\gamma_t^2}$

$\leq \prod_{t=1}^{T} e^{-2\gamma_t^2} \leq \prod_{t=1}^{T} e^{-2\gamma^2} = \exp\left(-2\sum_{t=1}^{T}\gamma^2\right)$

to achive zero $\Rightarrow \exp\left(-2\sum_{t=1}^{T}\gamma^2\right) \leq \frac{1}{N}$ (No 1 dataset misclassified)

$\exp(-2T\gamma^2) \leq \frac{1}{N}$

$-2T\gamma^2 \leq -\ln N$

$T \geq \frac{\ln N}{2\gamma^2}$

Math 4.1

hw4 4.1, compute $\mu_k^{(t+1)}$

M-step $Q(\theta|\theta^t) = \sum_{i=1}^{N}\sum_{k=1}^{K} \delta_{ik}^{(t)} \{ \log(\frac{\pi_k}{\sqrt{(2\pi)^m |\Sigma_k|}}) - \frac{1}{2}(x_i-\mu_k)^T \Sigma_k^{-1}(x_i-\mu_k) \}$

$\delta_{ik}^{(t)} = 1,\ \delta_{ij}^{(t)} = 0\ \forall\ y_i = k \neq j$

so $Q(\theta|\theta t) = \sum_{i=1, y_i=k}^{l} \log(\frac{1}{\sqrt{(2\pi)^m |\Sigma_k|}}) - \frac{1}{2}(x_i-\mu_k)^T \Sigma_k^{-1}(x_i-\mu_k)$

l meas labelled($y_i=k$)

u means unlabeled ($y_i=0$)

$l+u=N$

$+ \sum_{i=l+1}^{l+u}\sum_{k=1}^{K} \delta_{ik}^{(t)} \{ \log(\frac{\pi_k}{\sqrt{(2\pi)^m |\Sigma_k|}}) - \frac{1}{2}(x_i-\mu_k)^T \Sigma_k^{-1}(x_i-\mu_k) \}$

$$\nabla_{\mu_k} Q(\theta|\theta^{(t)}) = \sum_{i=l+1}^{N} \Sigma_k^{-1}(x_i-\mu_k)\delta_{ik}^{(t)} + \sum_{i=1, y_i=k}^{l} \Sigma_k^{-1}(x_i-\mu_k)$$

$$= \Sigma_k^{-1}\Big(\sum_{i, y_i=0} \delta_{ik}^{(t)}(x_i-\mu_k) + \sum_{i, y_i=k}(x_i-\mu_k)\Big) = 0$$

erase Σ_k^{-1}

$$\sum_{i,y_i=0} \delta_{ik}^{(t)} x_i - \mu_k \sum_{i,y_i=0} \delta_{ik}^{(t)} + \sum_{i,y_i=k} x_i - \mu_k \cdot \underbrace{\sum_{i,y_i=k} 1}_{N_k} = 0$$

$$\mu_k\Big(N_k + \sum_{i,y_i=0}\delta_{ik}^{(t)}\Big) = \sum_{i,y_i=k} x_i + \sum_{i,y_i=0}\delta_{ik}^{(t)} x_i$$

$$\Rightarrow \mu_k^{(t+1)} = \frac{\sum_{i,y_i=k} x_i + \sum_{i,y_i=0}\delta_{ik}^{(t)} x_i}{N_k + \sum_{i,y_i=0}\delta_{ik}^{(t)}}$$

hw4 4.1 compute $\pi_k^{(t+1)}$

already know

$$Q(\theta|\theta^{(t)}) = \sum_{i:y_i=k, k\in[1,\dots K]} \log\left(\frac{1}{\sqrt{(2\pi)^m|\Sigma_k|}}\right) - \frac{1}{2}(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)$$

$$+ \sum_{i:y_i=0}\sum_{k=1}^{K} \delta_{ik}^{(t)}\left\{\log\left(\frac{\pi_k}{\sqrt{(2\pi)^m|\Sigma_k|}}\right) - \frac{1}{2}(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right\}$$

$\sum_{k=1}^{K}\pi_k = 1$, $\nabla_{\pi_k}\left(Q(\theta|\theta^{(t)}) - \lambda\sum_{k=1}^{K}\pi_k\right)$

$$= \sum_{i:y_i=0}\frac{\delta_{ik}^{(t)}}{\pi_k} - \lambda$$

$$\Rightarrow \pi_k^{(t+1)} = \frac{\sum_{y_i=0}\delta_{ik}^{(t)}}{\lambda}, \quad \lambda = \sum_{i:y_i=0}\sum_{k=1}^{K}\delta_{ik}^{(t)} = \sum_{i:y_i=0} 1$$

$$\pi_k^{(t+1)} = \frac{\sum_{y_i=0}\delta_{ik}^{(t)}}{\sum_{i:y_i=0} 1}$$

compute $\Sigma_k^{(t+1)}$

rewrite $Q(\theta|\theta^{(t)})$ as

$$\sum_{i:y_i=k}\log\left(\frac{1}{\sqrt{(2\pi)^m}}\right) + \frac{1}{2}\left(\log|\Sigma_k^{-1}| - (x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right)$$

$$+ \sum_{i:y_i=0}\sum_{k=1}^{K}\delta_{ik}^{(t)}\left\{\log\left(\frac{\pi_k}{\sqrt{(2\pi)^m}}\right) + \frac{1}{2}\left(\log|\Sigma_k^{-1}| - (x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right)\right\}$$

we already know $\nabla_A \log(\det A) = (A^{-1})^T$, $\nabla_A tr(BA) = B^T$, $tr(BA) = tr(AB)$

$$\nabla_{\Sigma_k^{-1}}\left(Q(\theta|\theta^{(t)})\right) = \sum_{i:y_i=k}\left(\frac{1}{2}\nabla_{\Sigma_k^{-1}}\left(\log|\Sigma_k^{-1}|\right) - \frac{1}{2}\nabla_{\Sigma_k^{-1}}\operatorname{trace}\left((x_i-\mu_k)(x_i-\mu_k)^T\Sigma_k^{-1}\right)\right)$$

$$+ \sum_{i:y_i=0}\delta_{ik}^{(t)}\left(\frac{1}{2}\nabla_{\Sigma_k^{-1}}\left(\log|\Sigma_k^{-1}|\right) - \frac{1}{2}\nabla_{\Sigma_k^{-1}}\operatorname{trace}\left((x_i-\mu_k)(x_i-\mu_k)^T\Sigma_k^{-1}\right)\right)$$

$$= \sum_{i:y_i=k}\frac{1}{2}(\Sigma_k)^T - \frac{1}{2}\left((x_i-\mu_k)(x_i-\mu_k)^T\right)^T$$

$$+ \sum_{i:y_i=0}\delta_{ik}^{(t)}\,\frac{1}{2}(\Sigma_k)^T - \frac{1}{2}\left((x_i-\mu_k)(x_i-\mu_k)^T\right)^T = 0$$

$$\Rightarrow \Sigma_k^{(t+1)}\cdot\left(\sum_{i:y_i=k}1 + \sum_{i:y_i=0}\delta_{ik}^{(t)}\right) = \sum_{i:y_i=k}(x_i-\mu_k)(x_i-\mu_k)^T + \sum_{i:y_i=0}\delta_{ik}^{(t)}(x_i-\mu_k)(x_i-\mu_k)^T$$

$$\Rightarrow \Sigma_k^{(t+1)} = \frac{\sum_{i:y_i=k}(x_i-\mu_k^{(t+1)})(x_i-\mu_k^{(t+1)})^T + \sum_{i:y_i=0}\delta_{ik}^{(t)}(x_i-\mu_k^{(t+1)})(x_i-\mu_k^{(t+1)})^T}{N_k + \sum_{i:y_i=0}\delta_{ik}^{(t)}}$$

hw4 4.2

Math 4.2

hw4 4.2

$$N_k + \sum_{i:y_i=0} \delta_{ik}^{(t)}$$

$$\forall y_i = k \neq 0,\ \delta_{ij}^{(t)} = 0,\ j \neq k,\ \delta_{ik}^{(t)} = 1$$

$$\forall y_i = 0,\ \delta_{ik}^{(t)} = \frac{\pi_k^{(t)} \frac{1}{\sqrt{(2\pi)^m|\Sigma_k|}} \exp\left(-\frac{1}{2}(x_i-\mu_k)^T \Sigma_k^{-1}(x_i-\mu_k)\right)}{\sum_{j=1}^{K} \pi_j^{(t)} \cdot \frac{\exp\left(-\frac{1}{2}(x_i-\mu_k)^T \Sigma_k^{-1}(x_i-\mu_k)\right)}{\sqrt{(2\pi)^m|\Sigma_k|}}}$$

Math 5.1

```
T = np.zeros((10,10))
for i in range(10):
    for j in range(10):
        denominator_sum = np.sum(w[:, j])
        T[i][j] = w[i][j] / (denominator_sum)

T
```

```
array([[0.00322581, 0.0047619 , 0.48095238, 0.48095238, 0.0047619 ,
        0.00909091, 0.32580645, 0.00322581, 0.0047619 , 0.0047619 ],
       [0.00322581, 0.0047619 , 0.0047619 , 0.0047619 , 0.48095238,
        0.00909091, 0.00322581, 0.00322581, 0.0047619 , 0.48095238],
       [0.32580645, 0.0047619 , 0.0047619 , 0.0047619 , 0.0047619 ,
        0.00909091, 0.00322581, 0.32580645, 0.0047619 , 0.0047619 ],
       [0.32580645, 0.0047619 , 0.0047619 , 0.0047619 , 0.0047619 ,
        0.00909091, 0.32580645, 0.00322581, 0.0047619 , 0.0047619 ],
       [0.00322581, 0.48095238, 0.0047619 , 0.0047619 , 0.0047619 ,
        0.91818182, 0.00322581, 0.00322581, 0.0047619 , 0.0047619 ],
       [0.00322581, 0.0047619 , 0.0047619 , 0.0047619 , 0.48095238,
        0.00909091, 0.00322581, 0.00322581, 0.0047619 , 0.0047619 ],
       [0.32580645, 0.0047619 , 0.0047619 , 0.48095238, 0.0047619 ,
        0.00909091, 0.00322581, 0.32580645, 0.0047619 , 0.0047619 ],
       [0.00322581, 0.0047619 , 0.48095238, 0.0047619 , 0.0047619 ,
        0.00909091, 0.32580645, 0.00322581, 0.48095238, 0.0047619 ],
       [0.00322581, 0.0047619 , 0.0047619 , 0.0047619 , 0.0047619 ,
        0.00909091, 0.00322581, 0.32580645, 0.0047619 , 0.48095238],
       [0.00322581, 0.48095238, 0.0047619 , 0.0047619 , 0.0047619 ,
        0.00909091, 0.00322581, 0.00322581, 0.48095238, 0.0047619 ]])
```

Math 5.2

The python code for the algorithm, the result shows x1,x2,x3,x4,x8 belong to same class, and x5, x6, x7, x9, x10 belong to another class. The result corresponds with the graph.

```
t = 0
while True:
    t = t+1
    Y = np.matmul(T,y_list[t-1])
    for i in range (10) :
        sum = Y[i][0]+Y[i][1]
        Y[i][0] = Y[i][0] / sum
        Y[i][1] = Y[i][1] / sum
    Y[0][0] = 1
    Y [0][1] = 0
    Y[6][0] =0
    Y[6][1] = 1
    y_list.append(Y)
    print(t)
    print(y_list[t])
    if np.linalg.norm(y_list[t] - y_list[t-1], 'fro') < 10 ** (-8):
        break
```

```
 [0.79383113 0.20616887]
 [0.8815315  0.1184685 ]
 [0.09161317 0.90838683]
 [0.12206457 0.87793543]
 [0.         1.        ]
 [0.64129413 0.35870587]
 [0.38238569 0.61761431]
 [0.20242413 0.79757587]]
71
[[1.         0.        ]
 [0.82956518 0.17043482]
 [0.79383113 0.20616887]
 [0.8815315  0.1184685 ]
 [0.09161317 0.90838683]
 [0.12206457 0.87793543]
 [0.         1.        ]
 [0.64129412 0.35870588]
 [0.38238568 0.61761432]
 [0.20242413 0.79757587]]
```

Math 5.3, 5.4

hw4. 5-3 $Y^t = T Y^{t-1}$

$Y^t_{i,j} = Y^t_{ij} / \sum_{k=1}^{c} Y^t_{i,k}$

$\Rightarrow Y^t_{ij} = \dfrac{T_{i\text{th row}} \cdot Y^{t-1}_{j\text{th column}}}{T_{i\text{th row}} \cdot \text{sum}(Y^{t-1}\text{'s column})}$

$Y^t = \bar{T} Y^{t-1}$

$\Rightarrow Y^t_{ij} = \bar{T}_{i\text{th row}} \cdot Y^{t-1}_{j\text{th colum}}$

$\bar{T}_{ij} = T_{ij} / \sum_k T_{ik} = \bar{T}_{i\text{th row}} = \dfrac{T_{i\text{th row}}}{\text{sum of } T_{i\text{th row}}}$

combine these, we now have to prove

$T_{i\text{th row}} \cdot \text{sum}(Y^{t-1}\text{'s column}) = \text{sum}(T_{i\text{th row}})$

Y^{t-1}'s colum $i \Rightarrow$ the probability of point $k \in$ class i

$\Rightarrow$ sum of Y^{t-1}'s column $= \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ (since all probability sum $=1$)

T ith row $\cdot \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} =$ add all elements of ith row

$= \text{sum}(T \text{ ith row})$

$\Rightarrow Y^t = \bar{T} Y^{t-1}$

hw4 5-4 $Y^t = \bar{T} Y^{t-1}$

$\Rightarrow \begin{bmatrix} Y^t_L \\ Y^t_U \end{bmatrix} = \begin{bmatrix} \bar{T}_{ll} & \bar{T}_{lu} \\ \bar{T}_{ul} & \bar{T}_{uu} \end{bmatrix} \cdot \begin{bmatrix} Y^{t-1}_l \\ Y^{t-1}_u \end{bmatrix}$

$= \begin{bmatrix} \bar{T}_{ll} Y^{t-1}_l + \bar{T}_{lu} Y^{t-1}_u \\ \bar{T}_{ul} Y^{t-1}_l + \bar{T}_{uu} Y^{t-1}_u \end{bmatrix}$

$\Rightarrow Y^t_u = \bar{T}_{ul} Y^{t-1}_l + \bar{T}_{uu} Y^{t-1}_u$

since we will clamp the labeled data

$Y^t_{ij} = 0$ if $y_i \neq j$, $Y^t_{ij} = 1$ if $y_i = j$

for $i = 1 \dots l$, $j = 1 \dots c$ (Y^t_L)

exactly like $Y^{t-1}_L, Y^{t-2}_L \dots Y^0_L$

$\Rightarrow Y^t_L = Y^{t-1}_L$

$\alpha_t = kb + c$

Math 5.5,5.6

hw4 5.5 $Y^t_u = \bar{T}_{ul} Y^{t-1}_L + \bar{T}_{uu} Y^{t-1}_u$

$= \bar{T}_{ul} Y_L + \bar{T}_{uu} (\bar{T}_{ul} Y_L + \bar{T}_{uu} Y^{t-2}_u)$

$= \bar{T}_{ul} (Y_L + \bar{T}_{uu} Y_L) + \bar{T}_{uu}^2 Y^{t-2}_u$

$Y^t_u \cdots\cdots$

$= \bar{T}_{uu}^t Y^0_u + \bar{T}_{ul} Y_L \cdot (1 + \bar{T}_{uu} + \cdots + \bar{T}_{uu}^{t-1})$

$= \bar{T}_{uu}^t Y^0_u + \sum_{i=1}^{t} \bar{T}_{uu}^{i-1} \bar{T}_{ul} Y_L$

hw4. 5.6 $\because \bar{T}$ is row normalized

$\Rightarrow \sum_j \bar{T}_{ij} = 1 \Rightarrow \sum_j \bar{T}_{uu_{ij}} + \sum_j \bar{T}_{ul_{ij}} = 1$

$\Rightarrow \bar{T}$ coms from $T \Rightarrow$ no 0 element, all > 0

$\Rightarrow \sum_j \bar{T}_{ul_{ij}} > 0 \Rightarrow \sum_j \bar{T}_{uu_{ij}} = 1 - \sum_j \bar{T}_{ul_{ij}} < 1$

$\sum_j \bar{T}_{uu_{ij}} > 0$ $\qquad = \gamma_i$

$\Rightarrow \alpha\, \gamma_i < 1$

Math 5.6, 5.7

⑥. hw4 5-6

since $\sum_{j=1}^{u} \overline{T}_{uu\,ij} = \gamma_i$, $0 < \gamma_i < 1$

$\Rightarrow \sum_{j=1}^{u} \overline{T}_{uu\,ij} \le \gamma \;\forall i = 1, \dots u$, $\gamma = \max(\gamma_i)$

$$\sum_{j=1}^{u} \overline{T}^{n}_{uu\,ij} = \sum_j \sum_k \overline{T}^{(n-1)}_{uu\,ik} \, \overline{T}_{uu\,kj}$$

$$= \sum_k \overline{T}^{n-1}_{uu\,ik} \sum_j \overline{T}_{uu\,ij}$$

$$\le \sum_k \overline{T}^{(n-1)}_{uu\,ik} \cdot \gamma$$

$$\le \gamma^n$$

when $n \to \infty$, $\gamma^n \to 0$

$\Rightarrow \lim_{n\to\infty} \sum_j \overline{T}^{n}_{uu\,ij} \to 0 \Rightarrow$ the row sums of $\overline{T}^{n}_{uu}$ converges to 0

$\Rightarrow$ all elements of $\overline{T}_{uu} \to 0$ ($\overline{T}_{uu}$'s elements are all non-negative)

$$\Rightarrow \lim_{n\to\infty} \overline{T}^{n}_{uu} = 0$$

hw4 5.7

$$S_n = I + \overline{T}_{uu} + \overline{T}^{2}_{uu} + \dots + \overline{T}^{n-1}_{uu} = \sum_{i=1}^{n} \overline{T}^{i-1}_{uu}$$

$$S_n (I - \overline{T}_{uu}) = I - \overline{T}^{n}_{uu}$$

$$\lim_{n\to\infty} S_n (I - \overline{T}_{uu}) = I - \lim_{n\to\infty} \overline{T}^{n}_{uu} = I$$

$$\Rightarrow \lim_{n\to\infty} S_n = I \cdot (I - \overline{T}_{uu})^{-1} = (I - \overline{T}_{uu})^{-1}$$

since $Y^t_u = \overline{T}^{t}_{uu} Y^0_u + \sum_{i=1}^{t} \overline{T}^{i-1}_{uu} \overline{T}_{ul} Y_L$

$$= \overline{T}^{t}_{uu} Y^0_u + S_t \overline{T}_{ul} Y_L$$

$$\lim_{t\to\infty} Y^t_u = \lim_{t\to\infty} \overline{T}^{t}_{uu} Y^0_u + \lim_{t\to\infty} S_t \overline{T}_{ul} Y_L = (I - \overline{T}_{uu})^{-1} \overline{T}_{ul} Y_L$$

Math 5.8

For simplicity, I swap x7 and x2. The answer is roughly the same as to the iteration solution.

```
T_new = np.zeros((10,10))
for i in range(10):
    for j in range(10):
        denominator_sum = np.sum(T[i])
        T_new[i][j] = T[i][j] / (denominator_sum)
```

```
Tuu = T_new[2:10, 2:10]
Tul = T_new[2:10, 0:2]
```

```
I = np.eye(8)
a = I - Tuu
multiple = np.matmul(np.linalg.inv(a),Tul)
```

```
y_l = np.eye(2)
y_u = np.matmul(multiple, y_l)
```

```
concatenated_matrix = np.concatenate((y_l, y_u), axis=0)
concatenated_matrix
```

```
array([[1.        , 0.        ],
       [0.        , 1.        ],
       [0.79383112, 0.20616888],
       [0.8815315 , 0.1184685 ],
       [0.09161316, 0.90838684],
       [0.12206455, 0.87793545],
       [0.82956518, 0.17043482],
       [0.64129412, 0.35870588],
       [0.38238567, 0.61761433],
       [0.20242412, 0.79757588]])
```