1. (a) The data point $x_d^{(n)}$ takes values either 0 or 1, which means the data are binary and follow Bernoulli distribution. We only fit a multivariate Gaussian model when the data are continuous and follow Gaussian distribution.

   Usually the image data have large number of pixels (D is large), which means the number of dimension is large. This requires extremely large number of parameters for the covariance matrix of multivariate Gaussian.

   Thus rather than a multivariate Gaussian model, a binary model is more suitable in this case.

   (b) To compute the maximum likelihood estimate for $\mathbf{p}$, we first compute likelihood and loglikelihood functions:

$$P(\mathbf{x}|\mathbf{p}) = \prod_{n=1}^{N}\prod_{d=1}^{D} p_d^{x_d^{(n)}}(1-p_d)^{1-x_d^{(n)}} = \mathcal{L}(\mathbf{p}|\mathbf{x}) \tag{1}$$

   Consider single element p_d, where $d \in \{1, \cdots, D\}$:

$$\ell(p_d|\mathbf{x}) = \log \mathcal{L}(p_d|\mathbf{x}) = \sum_{n=1}^{N} x_d^{(n)} \log p_d + (1 - x_d^{(n)}) \log(1-p_d)$$

$$\begin{aligned} \frac{\partial \ell(p_d|\mathbf{x})}{\partial p_d} &= \sum_{n=1}^{N} \frac{x_d^{(n)}}{p_d} - \frac{1-x_d^{(n)}}{1-p_d} \\ &= \frac{\sum_{n=1}^{N} x_d^{(n)}}{p_d} - \frac{N - \sum_{n=1}^{N} x_d^{(n)}}{1-p_d} = 0 \end{aligned} \tag{2}$$

$$\hat{p_d} = \frac{\sum_{n=1}^{N} x_d^{(n)}}{N}$$

   (c) The MAP estimate for $\mathbf{p}$:

$$\begin{aligned} \pi(\mathbf{p}|\mathbf{x}) &\propto \mathcal{L}(\mathbf{p}|\mathbf{x}) \times P(\mathbf{p}) \quad \text{(by Bayes theorem)} \\ &= \prod_{n=1}^{N}\prod_{d=1}^{D} p_d^{x_d^{(n)}}(1-p_d)^{1-x_d^{(n)}} \times \prod_{d=1}^{D} \frac{1}{B(\alpha,\beta)} p_d^{\alpha-1}(1-p_d)^{\beta-1} \end{aligned} \tag{3}$$

   Consider single element p_d, where $d \in \{1, \cdots, D\}$:

$$\begin{aligned} \ell(p_d|\mathbf{x}) &= \log \pi(p_d|\mathbf{x}) \\ &\propto \log\left(\prod_{n=1}^{N} p_d^{x_d^{(n)}}(1-p_d)^{1-x_d^{(n)}} \times \frac{1}{B(\alpha,\beta)} p_d^{\alpha-1}(1-p_d)^{\beta-1}\right) \\ &\propto (\sum_{n=1}^{N} x_d^{(n)} + \alpha - 1) \log p_d + (N - \sum_{n=1}^{N} x_d^{(n)} + \beta - 1)\log(1-p_d) \end{aligned} \tag{4}$$

$$\frac{\partial \ell(p_d|\mathbf{x})}{\partial p_d} = \frac{\sum_{n=1}^{N} x_d^{(n)} + \alpha - 1}{p_d} - \frac{N - \sum_{n=1}^{N} x_d^{(n)} + \beta - 1}{1-p_d} = 0$$

$$\hat{p_d} = \frac{\sum_{n=1}^{N} x_d^{(n)} + \alpha - 1}{\alpha + \beta + N - 2}$$

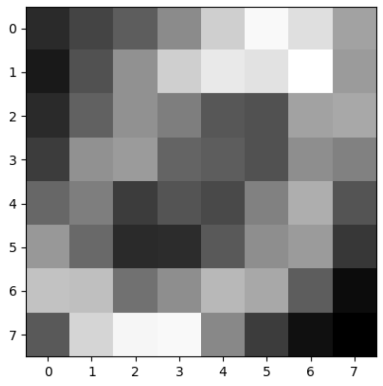

Figure 1: p_{ml} parameter image

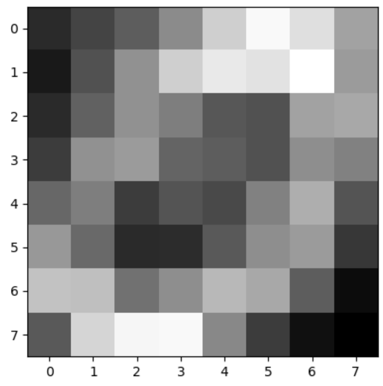

Figure 2: p_{map} parameter image

(d) The python code of learning p_{ml} and generating the figure 1:

```
Y = np.loadtxt('binarydigits.txt')
N,D = Y.shape
p_ml = sum(Y)/N
plt.figure()
plt.imshow(np.reshape(p_ml, (8,8)),
            interpolation="None",
            cmap='gray')
plt.show()
```

(e) The python code of learning p_{map} and generating the figure 2:

```
a = 3
b = 3
p_map = sum(Y)/(a+b-2+N) + np.repeat(1,D)*(a-1)/(a+b-2+N)
plt.figure()
plt.imshow(np.reshape(p_map, (8,8)),
            interpolation="None",
            cmap='gray')
plt.show()
```

Compared with ML, MAP adds the effect of prior to the parameters, for instance, MAP eliminates the zero-probability pixels. If the prior resembles well with the true distribution of the model parameters, MAP will give a better point estimate; however, if the prior cannot represent the true distribution well, MAP may have larger bias than ML.

2. (a) The probability of model1 if $p_d = 0.5$:

$$P(\mathbf{x}|\mathbf{p}) = \prod_{n=1}^{N}\prod_{d=1}^{D} p_d^{x_d^{(n)}}(1-p_d)^{1-x_d^{(n)}} = 0.5^{N*D} \tag{5}$$

The posterior of model1:

$$P(\mathbf{p}|\mathbf{x}) \propto P(\mathbf{x}|\mathbf{p}) \times P(\mathbf{p}) = 0.5^{6400} \tag{6}$$

(b) If p_d is unknown but identical, all the data point can be considered drawn from the same Bernoulli distribution, thus the probability of model2:

$$P(\mathbf{x}|\mathbf{p}) = \prod_{n=1}^{N}\prod_{d=1}^{D} p_d^{x_d^{(n)}}(1-p_d)^{1-x_d^{(n)}} = p_d^{2495}(1-p_d)^{3905} \tag{7}$$

where 3905 and 2495 are the numbers of 0s and 1s respectively in the data. The posterior of model2:

$$P(\mathbf{p}|\mathbf{x}) \propto P(\mathbf{x}|\mathbf{p}) \times P(\mathbf{p}) = p_d^{2495}(1-p_d)^{3905} \tag{8}$$

(c) If each component has unknown and separate p_d, the probability of model3:

$$P(\mathbf{x}|\mathbf{p}) = \prod_{n=1}^{N}\prod_{d=1}^{D} p_d^{x_d^{(n)}}(1-p_d)^{1-x_d^{(n)}} \tag{9}$$

The posterior of model3:

$$P(\mathbf{p}|\mathbf{x}) \propto P(\mathbf{x}|\mathbf{p}) \times P(\mathbf{p}) = \prod_{n=1}^{N}\prod_{d=1}^{D} p_d^{x_d^{(n)}}(1-p_d)^{1-x_d^{(n)}} \tag{10}$$

3. (a) Given that images are iid and pixels are independent of each other:

$$P(\mathbf{x}|\boldsymbol{\pi}, \mathrm{P}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} p_{kd}^{x_d^{(n)}} (1-p_{kd})^{(1-x_d^{(n)})} \tag{11}$$

(b) The responsibility r_{nk} can be expressed as:

$$\begin{aligned} r_{nk} &= P(s^{(n)} = k|\mathbf{x}^{(n)}, \boldsymbol{\pi}, \mathrm{P}) \\ &= \frac{P(\mathbf{x}^{(n)}|s^{(n)} = k, \mathbf{p})P(s^{(n)} = k|\boldsymbol{\pi})}{\sum_k P(\mathbf{x}^{(n)}|s^{(n)} = k, \mathbf{p})P(s^{(n)} = k|\boldsymbol{\pi})} \\ &= \frac{\pi_k \prod_{d=1}^{D} p_{kd}^{x_d^{(n)}} (1-p_{kd})^{(1-x_d^{(n)})}}{\sum_k \pi_k \prod_{d=1}^{D} p_{kd}^{x_d^{(n)}} (1-p_{kd})^{(1-x_d^{(n)})}} \end{aligned} \tag{12}$$

(c) In order to derive

$$\operatorname*{arg\,max}_{\boldsymbol{\pi}, P} \left\langle \sum_n \log P(\mathbf{x}^{(n)}, s^{(n)}|\boldsymbol{\pi}, \mathrm{P}) \right\rangle_{q(\{s^{(n)}\})}, \tag{13}$$

we firstly compute $\sum_n \log P(\mathbf{x}^{(n)}, s^{(n)}|\boldsymbol{\pi}, \mathbf{p})$.

$$\begin{aligned} P(\mathbf{x}^{(n)}, s^{(n)} = k|\boldsymbol{\pi}, \mathrm{P}) =& P(\mathbf{x}^{(n)}|\boldsymbol{\pi}, \mathrm{P})P(s^{(n)} = k|\mathbf{x}^{(n)}, \boldsymbol{\pi}, \mathrm{P}) \\ =& \left(\sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} p_{kd}^{x_d^{(n)}} (1-p_{kd})^{(1-x_d^{(n)})}\right) r_{nk} \\ =& \pi_k \prod_{d=1}^{D} p_{kd}^{x_d^{(n)}} (1-p_{kd})^{(1-x_d^{(n)})} \\ P(\mathbf{x}^{(n)}, s^{(n)}|\boldsymbol{\pi}, \mathrm{P}) =& \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} p_{kd}^{x_d^{(n)}} (1-p_{kd})^{(1-x_d^{(n)})} \\ \sum_n \log P(\mathbf{x}^{(n)}, s^{(n)}|\boldsymbol{\pi}, \mathrm{P}) =& \sum_n \log \left(\sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} p_{kd}^{x_d^{(n)}} (1-p_{kd})^{(1-x_d^{(n)})}\right) \end{aligned} \tag{14}$$

By taking the partial derivative wrt π_k:

$$\begin{aligned} \frac{\partial}{\partial \pi_k} \sum_n \log P(\mathbf{x}^{(n)}, s^{(n)}|\boldsymbol{\pi}, \mathrm{P}) &= \sum_n \frac{\prod_{d=1}^{D} p_{kd}^{x_d^{(n)}} (1-p_{kd})^{(1-x_d^{(n)})}}{\sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} p_{kd}^{x_d^{(n)}} (1-p_{kd})^{(1-x_d^{(n)})}} \\ &= \sum_n \frac{r_{nk}}{\pi_k} \end{aligned} \tag{15}$$

By taking the partial derivative wrt p_{kd}:

$$\begin{aligned} \frac{\partial}{\partial p_{kd}} \sum_n \log P(\mathbf{x}^{(n)}, s^{(n)}|\boldsymbol{\pi}, \mathrm{P}) &= \sum_n \frac{\pi_k \prod_{d=1}^{D} p_{kd}^{x_d^{(n)}-1} (1-p_{kd})^{-x_d^{(n)}} (x_d^{(n)} - p_{kd})}{\sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} p_{kd}^{x_d^{(n)}} (1-p_{kd})^{(1-x_d^{(n)})}} \\ &= \sum_n r_{nk} \frac{x_d^{(n)} - p_{kd}}{p_{kd}(1-p_{kd})} \end{aligned} \tag{16}$$

The maximum of p_{kd} is found by setting the partial derivatives to zero:

$$\begin{aligned}\frac{\partial}{\partial p_{kd}}\sum_n \log P(\mathbf{x}^{(n)}, s^{(n)}|\boldsymbol{\pi}, \mathrm{P}) = \sum_n r_{nk}\frac{x_d^{(n)} - p_{kd}}{p_{kd}(1-p_{kd})} = 0 \\ p_{kd} = \frac{\sum_n r_{nk}x_d^{(n)}}{\sum_n r_{nk}}\end{aligned} \tag{17}$$

We use the Lagrange multiplier λ to find the maximum for π_k, given the constraint that $\sum_k \pi_k = 1$:

$$\begin{aligned}L &= \sum_n \log P(\mathbf{x}^{(n)}, s^{(n)}|\boldsymbol{\pi}, \mathrm{P}) - \lambda(\sum_k \pi_k - 1) \\ \frac{\partial L}{\partial \pi_k} &= \frac{\partial}{\partial \pi_k}\sum_n \log P(\mathbf{x}^{(n)}, s^{(n)}|\boldsymbol{\pi}, \mathrm{P}) - \lambda \\ &= \sum_n \frac{r_{nk}}{\pi_k} - \lambda = 0 \\ \pi_k &= \frac{\sum_n r_{nk}}{\lambda} \\ \sum_k \pi_k &= \frac{\sum_k\sum_n r_{nk}}{\lambda} = 1 \\ \lambda &= \sum_k\sum_n r_{nk}\end{aligned} \tag{18}$$

By plugging $\lambda = \sum_k\sum_n r_{nk}$ back into the equation $\sum_n \frac{r_{nk}}{\pi_k} - \lambda = 0$ that we derived above, we get:

$$\begin{aligned}\sum_n \frac{r_{nk}}{\pi_k} &= \sum_k\sum_n r_{nk} \\ \pi_k &= \frac{\sum_n r_{nk}}{\sum_k\sum_n r_{nk}} \\ &= \frac{\sum_n r_{nk}}{N},\end{aligned} \tag{19}$$

since $\sum_k r_{nk} = 1$ (r_{nk} is already normalised).

(d) In this question, we fix our parameter initialisation for $\boldsymbol{\pi}$ and P, while testing different K values in $\{2, 3, 4, 7, 10\}$. The corresponding loglikelihood plots are as follows:

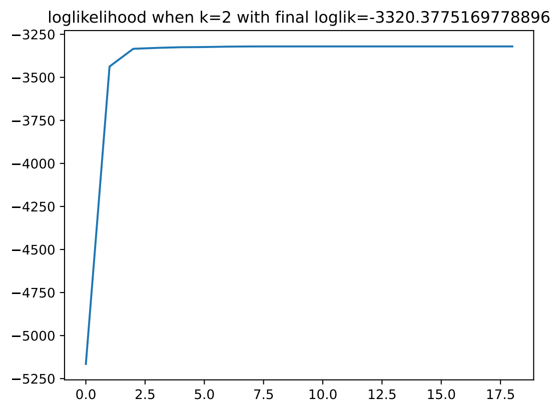

Figure 3: K=2

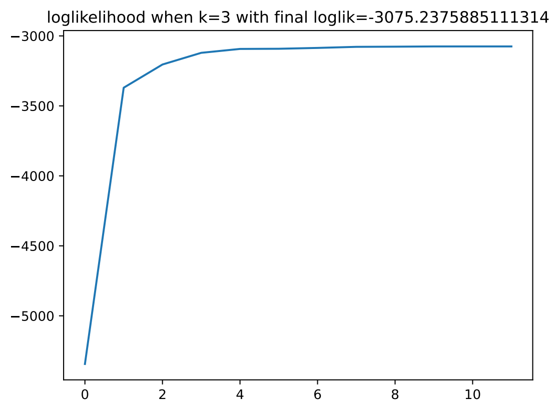

Figure 4: K=3

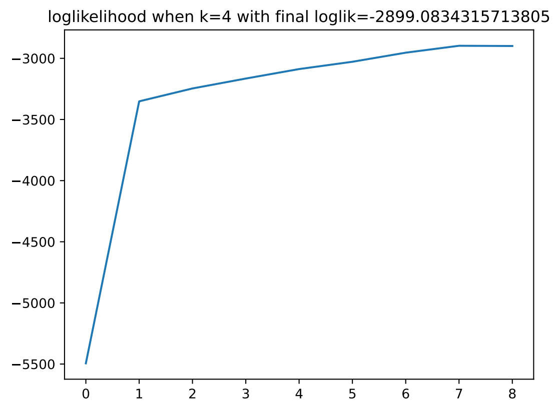

Figure 5: K=4

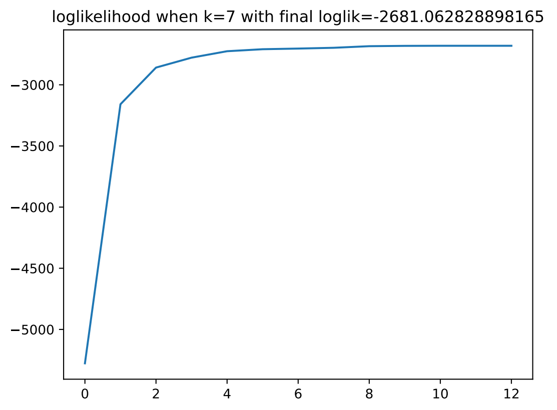

Figure 6: K=7

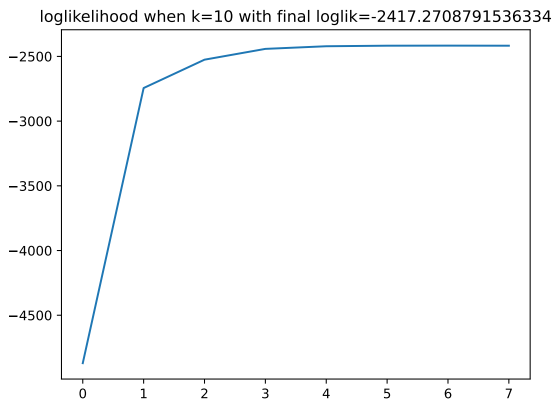

Figure 7: K=10

The values of parameters $\boldsymbol{\pi}$ (indicated as "weight" at the top of each subplot) and 8×8 images containing values of P are displayed as follows:

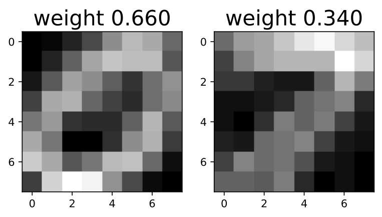

Figure 8: K=2

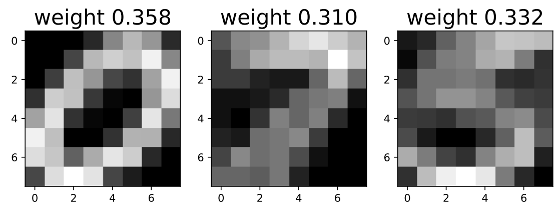

Figure 9: K=3

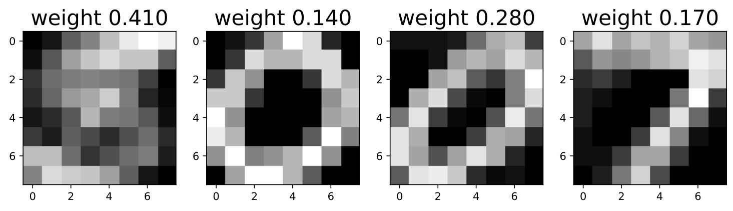

Figure 10: K=4

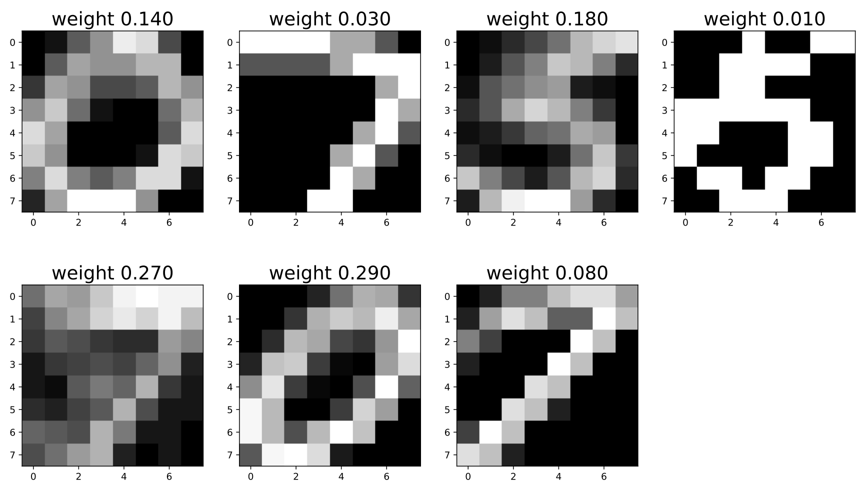

Figure 11: K=7

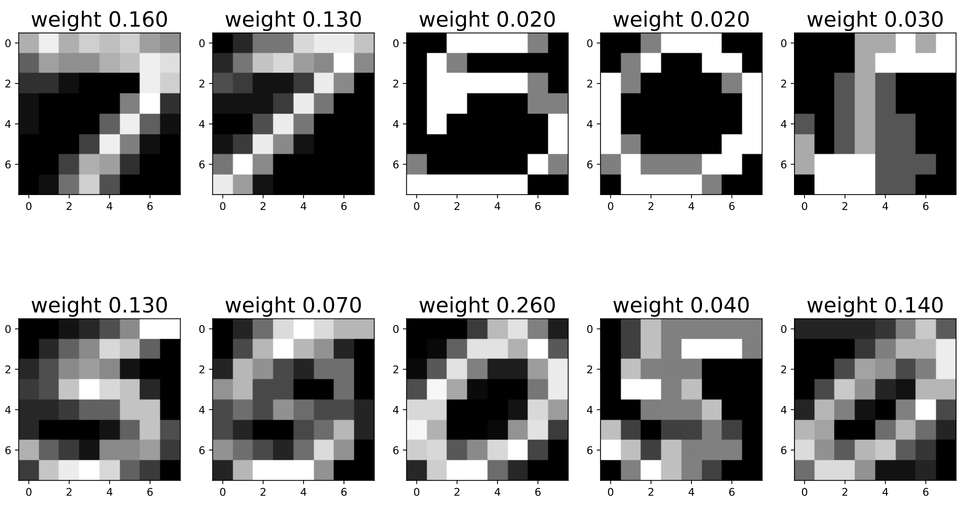

Figure 12: K=10

(e) In this question, we fix the value of K (take $K = 2$ as an example) and do several different parameter initialisations. The corresponding parameter plots (including $\boldsymbol{\pi}$, being indicated as "weight", and P) are as follows:

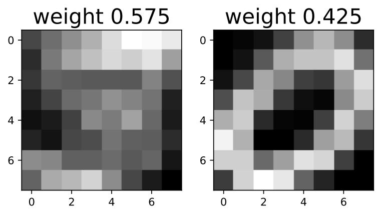

Figure 13: K=2,seed=0

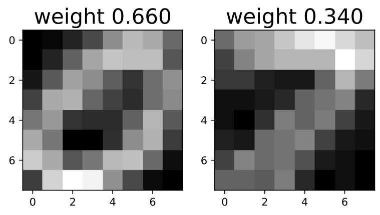

Figure 14: K=2,seed=1

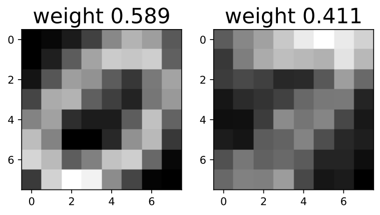

Figure 15: K=2,seed=2

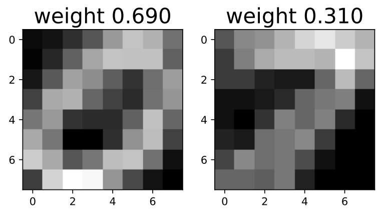

Figure 16: K=2,seed=3

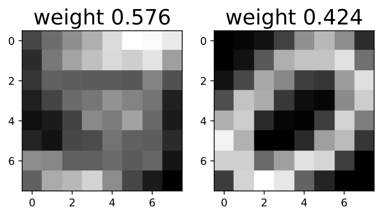

Figure 17: K=2,seed=4

From the images above, we can tell: different initial values will lead to different optimal of parameters $\boldsymbol{\pi}$ and P. This principle also applies on different Ks.

In general, the parameter images for K values {2,3,4,7,10} reveal 3 types of hand-written digits, namely 0,5,7. For K=7 or K=10, the clusters perform not very well, since some clusters have relatively too low weights (0.01, 0.02 etc.) and this may be caused by overfitting. For K=2, it seems that the number of clusters are too limited so that the algorithm tries to mix the digits/clusters. Comparing K=3 and K=4, both of which provide reasonable digits and weights, K=3 seems better, for the weights are quite even and 3 clusters exactly reveal 3 types of hand-written digits (0,5,7). Thus, K=3 seems to provide the best cluster.

To improve the algorithm, we may consider 2 ways:

1. Penalising the long distance between different clusters by adding penalty terms, so that the overfitting may be mitigated.
2. Introducing weak priors on $\boldsymbol{\pi}$ and P to find MAP estimates instead of the ML estimates. Potential priors: beta distribution.

(f) The bits can be calculated by $-\frac{\text{loglikelihood}}{\log(2)}$, thus:

K values	loglikelihood	bits
2	-3320.3775	4790
3	-3075.2376	4437
4	-2899.0834	4182
7	-2681.0628	3868
10	-2417.2709	3487

Table 1: loglikelihood to bits, noted that the loglikelihood values are taken from question(d), which sets seed at 1.

Using two-symbol Huffman coding, encoding 6400 pixels needs approximately 6885 bits. From the table above, if we choose bigger K, then the likelihood is larger, our length of encoding will be smaller.

(g) Take $K = 2$ as an example. We only have 2 parameters, namely $\boldsymbol{\pi}$ (with size $1 \times K = 2$) and P (with size $K \times D = 2 \times 64 = 128$). Thus our parameters use $128 + 2 = 130$ double numbers, which costs 8320 bits. The encoding of our model costs 4790 bits, calculated in the above table. Thus the total cost is $8320 + 4790 = 13110$ bits.

The encoding cost using two-symbol Huffman is 6885 bits, plus the cost storing the code table ($\{00 \to 11, 01 \to 01, 10 \to 10, 11 \to 001\}$), which is 17 bits, we get 6902 bits of total cost.

We discover that the naive encoding costs more than the two-symbol Huffman encoding. By increasing the value of K, the cost encoding the data will be smaller, but we will cost significantly more in terms of the parameters (with size $1 \times K$ and $K \times D$). There exists trade-off between maximising the likelihood and minimising the encoding cost.

4. (a) The Y plots under “filt” and “smooth” modes are displayed below:

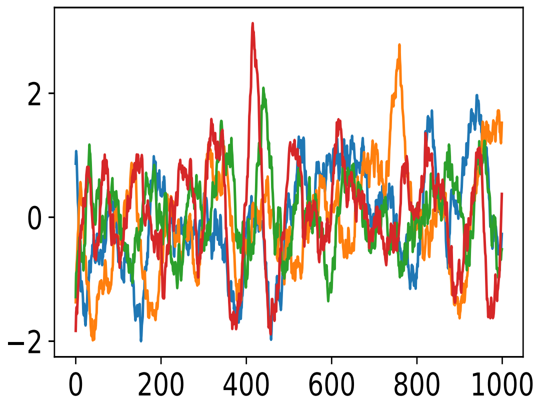
Figure 18: Y plot with mode “filt”

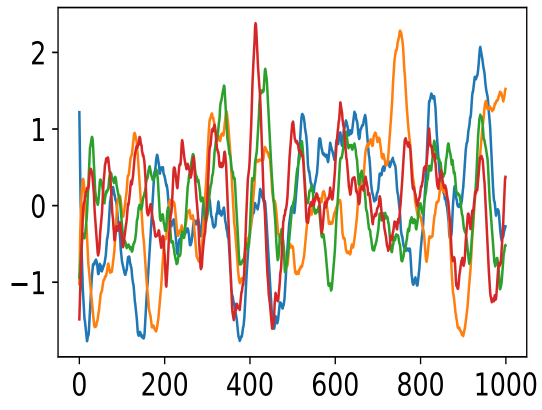
Figure 19: Y plot with mode “smooth”

As shown in the above plots, in both cases: the general trends and periodicity for the posterior means are roughly the same, both of which fluctuate roughly in the interval -2 to 2. The fluctuation is caused by the additive Gaussian noise. In terms of the differences between the 2 cases: the filtered means have obviously more fluctuations (like serrated waves) compared with the smoothed means, since the filtered means do not include the future observations. This also means the filtered means are more sensitive to the observation values.

The V plots under “filt” and “smooth” modes are displayed below:

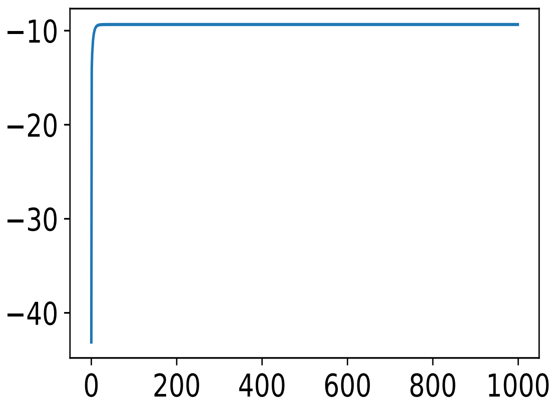
Figure 20: V plot with mode “filt”

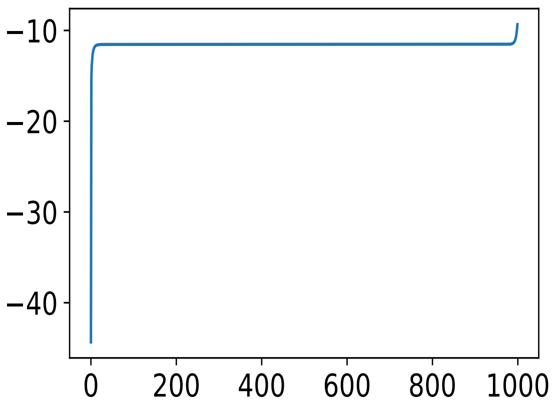
Figure 21: V plot with mode “smooth”

In both cases, the determinants of the covariance matrices increase sharply at the very beginning and become quite flat later on. The beginning point is given by the initialised prior Q. Noticeably, however:

- The determinants of the covariance matrix in the “smooth” case increases sharply again at the very end of time t. The sharp increase is caused by the lack of future observations.

- At the flat stage, the uncertainty level of the "smooth" case is lower than the one in the "filt" case, since the smoothed means involve both the past and future observations, making better prediction.

(b) The M-step for R:

$$\mathbb{P}(\mathbf{x}_t|\mathbf{y}_t) \propto \frac{1}{2\pi|R|}\exp\left\{-\frac{1}{2}(\mathbf{x}_t - C\mathbf{y}_t)^T R^{-1}(\mathbf{x}_t - C\mathbf{y}_t)\right\}, \tag{20}$$

$$\begin{aligned}
R_{\text{new}} &= \arg\max_R \left\langle \sum_{t=1}^{T} \log \mathbb{P}(\mathbf{x}_t|\mathbf{y}_t) \right\rangle_q \\
&= \arg\max_R \left\langle -\frac{T}{2}\log|R| - \frac{1}{2}\sum_{t=1}^{T}(\mathbf{x}_t - C\mathbf{y}_t)^T R^{-1}(\mathbf{x}_t - C\mathbf{y}_t) \right\rangle_q \\
&= \arg\min_R \left\langle \frac{T}{2}\log|R| + \frac{1}{2}\sum_{t=1}^{T}(\mathbf{x}_t - C\mathbf{y}_t)^T R^{-1}(\mathbf{x}_t - C\mathbf{y}_t) \right\rangle_q \\
\frac{\partial \ell}{\partial R} &= \frac{T|R|R^{-1}}{2|R|} - \frac{1}{2}R^{-1}\left\langle \sum_{t=1}^{T}(\mathbf{x}_t - C\mathbf{y}_t)(\mathbf{x}_t - C\mathbf{y}_t)^T \right\rangle_q R^{-1} \\
&= \frac{T}{2}R^{-1} - \frac{1}{2}R^{-1}\left\langle \sum_{t=1}^{T}(\mathbf{x}_t - C\mathbf{y}_t)(\mathbf{x}_t - C\mathbf{y}_t)^T \right\rangle_q R^{-1} = 0 \\
R_{\text{new}} &= \frac{1}{T}\left\langle \sum_{t=1}^{T}(\mathbf{x}_t - C\mathbf{y}_t)(\mathbf{x}_t - C\mathbf{y}_t)^T \right\rangle_q \\
&= \frac{1}{T}\left[\sum_{t=1}^{T}\left\langle \mathbf{x}_t\mathbf{x}_t^T \right\rangle - C\left(\sum_{t=1}^{T}\left\langle \mathbf{x}_t\mathbf{y}_t^T \right\rangle\right) - \left(\sum_{t=1}^{T}\left\langle \mathbf{x}_t\mathbf{y}_t^T \right\rangle\right)C^T\right] \\
&+ \frac{1}{T}\left[C\left(\sum_{t=1}^{T}\left\langle \mathbf{y}_t\mathbf{y}_t^T \right\rangle\right)C^T\right]
\end{aligned} \tag{21}$$

Plug $C = C_{\text{new}}$ into the last part (with coefficient excluded) of the above equation, where:

$$C_{\text{new}} = \left(\sum_{t=1}^{T}\langle \mathbf{x}_t\mathbf{y}_t^T\rangle\right)\left(\sum_{t=1}^{T}\langle \mathbf{y}_t\mathbf{y}_t^T\rangle\right)^{-1} \tag{22}$$

Thus:

$$\begin{aligned}\left[C\left(\sum_{t=1}^{T}\langle\mathbf{y}_t\mathbf{y}_t^T\rangle\right)C^T\right] &= \left(\sum_{t=1}^{T}\langle\mathbf{x}_t\mathbf{y}_t^T\rangle\right)\left(\sum_{t=1}^{T}\langle\mathbf{y}_t\mathbf{y}_t^T\rangle\right)^{-1}\left(\sum_{t=1}^{T}\langle\mathbf{y}_t\mathbf{y}_t^T\rangle\right)C_{\text{new}}^T \\ &= \left(\sum_{t=1}^{T}\langle\mathbf{x}_t\mathbf{y}_t^T\rangle\right)C_{\text{new}}^T \\ &= \left(\sum_{t=1}^{T}\langle\mathbf{x}_t\mathbf{y}_t^T\rangle\right)\left(\sum_{t=1}^{T}\langle\mathbf{y}_t\mathbf{y}_t^T\rangle\right)^{-1}\left(\sum_{t=1}^{T}\langle\mathbf{x}_t\mathbf{y}_t^T\rangle\right)^T \\ &= C_{\text{new}}\left(\sum_{t=1}^{T}\langle\mathbf{x}_t\mathbf{y}_t^T\rangle\right)^T,\end{aligned} \tag{23}$$

which cancels out with the second term in the above equation. Noticeably, the second term, third term and the fourth term are all equal. Therefore:

$$R_{\text{new}} = \frac{1}{T}\left[\sum_{t=1}^{T}\langle\mathbf{x}_t\mathbf{x}_t^T\rangle - \left(\sum_{t=1}^{T}\langle\mathbf{x}_t\mathbf{y}_t^T\rangle\right)C_{\text{new}}^T\right] \tag{24}$$

The M-step for Q:

$$\mathbb{P}(\mathbf{y}_{t+1}|\mathbf{y}_t) \propto \frac{1}{2\pi|Q|}\exp\left\{-\frac{1}{2}(\mathbf{y}_{t+1}-A\mathbf{y}_t)^TQ^{-1}(\mathbf{y}_{t+1}-A\mathbf{y}_t)\right\} \tag{25}$$

$$\begin{aligned}Q_{\text{new}} &= \arg\max_Q\left\langle\sum_{t=1}^{T-1}\log\mathbb{P}(\mathbf{y}_{t+1}|\mathbf{y}_t)\right\rangle_q \\ &= \arg\max_Q\left\langle-\frac{T-1}{2}\log|Q| - \frac{1}{2}\sum_{t=1}^{T-1}(\mathbf{y}_{t+1}-A\mathbf{y}_t)^TQ^{-1}(\mathbf{y}_{t+1}-A\mathbf{y}_t)\right\rangle_q \\ &= \arg\min_Q\left\langle\frac{T-1}{2}\log|Q| + \frac{1}{2}\sum_{t=1}^{T-1}(\mathbf{y}_{t+1}-A\mathbf{y}_t)^TQ^{-1}(\mathbf{y}_{t+1}-A\mathbf{y}_t)\right\rangle_q \\ \frac{\partial\ell}{\partial Q} &= \frac{(T-1)|Q|Q^{-1}}{2|Q|} - \frac{1}{2}Q^{-1}\left\langle\sum_{t=1}^{T-1}(\mathbf{y}_{t+1}-A\mathbf{y}_t)(\mathbf{y}_{t+1}-A\mathbf{y}_t)^T\right\rangle_q Q^{-1} \\ &= \frac{T-1}{2}Q^{-1} - \frac{1}{2}Q^{-1}\left\langle\sum_{t=1}^{T-1}(\mathbf{y}_{t+1}-A\mathbf{y}_t)(\mathbf{y}_{t+1}-A\mathbf{y}_t)^T\right\rangle_q Q^{-1} = 0 \\ Q_{\text{new}} &= \frac{1}{T-1}\left\langle\sum_{t=1}^{T-1}(\mathbf{y}_{t+1}-A\mathbf{y}_t)(\mathbf{y}_{t+1}-A\mathbf{y}_t)^T\right\rangle_q \\ &= \frac{1}{T-1}\left[\sum_{t=1}^{T-1}\langle\mathbf{y}_{t+1}\mathbf{y}_{t+1}^T\rangle - A\left(\sum_{t=1}^{T-1}\langle\mathbf{y}_{t+1}\mathbf{y}_t^T\rangle\right) - \left(\sum_{t=1}^{T-1}\langle\mathbf{y}_{t+1}\mathbf{y}_t^T\rangle\right)A^T\right] \\ &+ \frac{1}{T-1}\left[A\left(\sum_{t=1}^{T-1}\langle\mathbf{y}_t\mathbf{y}_t^T\rangle\right)A^T\right]\end{aligned} \tag{26}$$

Plug $A = A_{\text{new}}$ into the last part (with coefficient excluded) of the above equation, where:

$$A_{\text{new}} = \left(\sum_{t=1}^{T-1} \langle \mathbf{y}_{t+1}\mathbf{y}_t^T \rangle\right) \left(\sum_{t=1}^{T-1} \langle \mathbf{y}_t\mathbf{y}_t^T \rangle\right)^{-1} \tag{27}$$

Thus:

$$\begin{aligned}
\left[A\left(\sum_{t=1}^{T-1} \langle \mathbf{y}_t\mathbf{y}_t^T \rangle\right) A^T\right] &= \left(\sum_{t=1}^{T-1} \langle \mathbf{y}_{t+1}\mathbf{y}_t^T \rangle\right) \left(\sum_{t=1}^{T-1} \langle \mathbf{y}_t\mathbf{y}_t^T \rangle\right)^{-1} \left(\sum_{t=1}^{T-1} \langle \mathbf{y}_t\mathbf{y}_t^T \rangle\right) A_{\text{new}}^T \\
&= \left(\sum_{t=1}^{T-1} \langle \mathbf{y}_{t+1}\mathbf{y}_t^T \rangle\right) A_{\text{new}}^T \\
&= \left(\sum_{t=1}^{T-1} \langle \mathbf{y}_{t+1}\mathbf{y}_t^T \rangle\right) \left(\sum_{t=1}^{T-1} \langle \mathbf{y}_t\mathbf{y}_t^T \rangle\right)^{-1} \left(\sum_{t=1}^{T-1} \langle \mathbf{y}_{t+1}\mathbf{y}_t^T \rangle\right)^T \\
&= A_{\text{new}} \left(\sum_{t=1}^{T-1} \langle \mathbf{y}_{t+1}\mathbf{y}_t^T \rangle\right)^T ,
\end{aligned} \tag{28}$$

which cancels out with the second term in the above equation. Noticeably, the second term, third term and the fourth term are all equal. Therefore:

$$\begin{aligned}
Q_{\text{new}} &= \frac{1}{T-1}\left[\sum_{t=1}^{T-1} \langle \mathbf{y}_{t+1}\mathbf{y}_{t+1}^T \rangle - \left(\sum_{t=1}^{T-1} \langle \mathbf{y}_{t+1}\mathbf{y}_t^T \rangle\right) A_{\text{new}}^T\right] \\
&= \frac{1}{T-1}\left[\sum_{t=2}^{T} \langle \mathbf{y}_t\mathbf{y}_{t-1}^T \rangle - \left(\sum_{t=2}^{T} \langle \mathbf{y}_t\mathbf{y}_{t-1}^T \rangle\right) A_{\text{new}}^T\right]
\end{aligned} \tag{29}$$

By setting the initial condition as "the generating parameters above" and "10 random choices", the loglikelihood vs iteration plots are obtained and displayed below. Noted that the red dotted lines in both cases are the initial loglikelihood provided by the initial values in question (a).

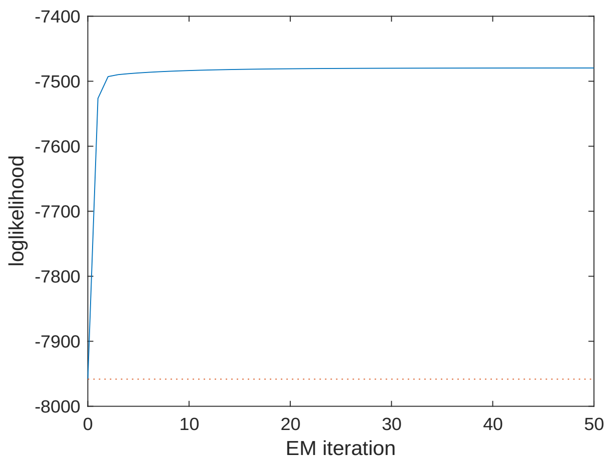

Figure 22: initial condition = (1)

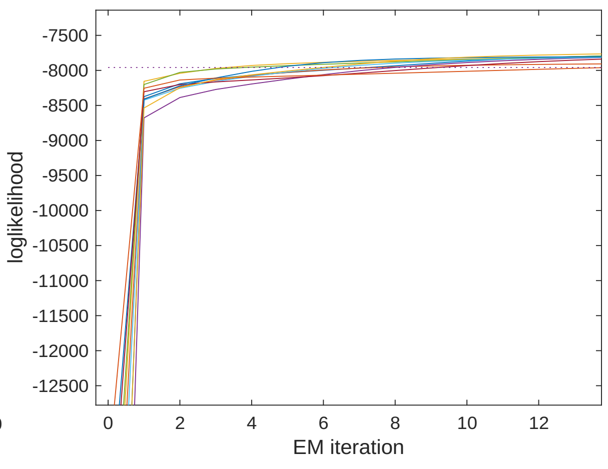

Figure 23: initial condition = (2), repeat 10 times

From Figure 22 and Figure 23:

- In both cases (manually set initialisation, random initialisation), the loglikelihood monotonically increases as the EM iteration goes. The loglikelihood increases sharply at the very beginning, then quickly becomes steady (but may still need some time to eventually converge).
- In the manual initialisation case, EM does not terminate immediately since the parameters with max likelihood may be different, there exists variability.
- In the random initialisation case, the loglikelihood starts from a much larger starting value compared with the manual initialisation case. Besides, different initial values have different converging speeds, and tend to converge to different levels.

We set the iteration time as 500 and obtain the plot below:

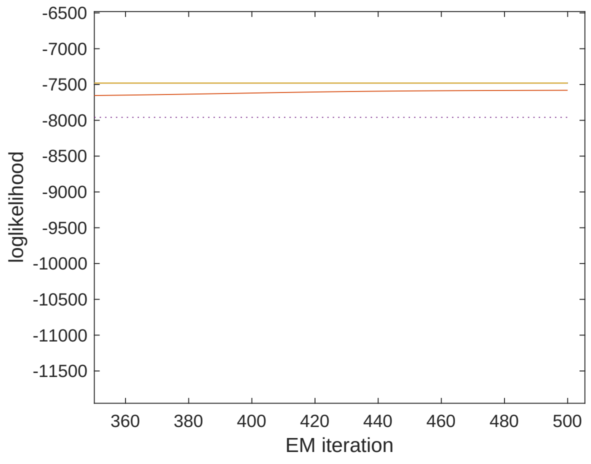

Figure 24: random initialisation, with iteration = 500

Slightly different from what we discovered in Figure 23, different random values eventually converge to roughly the same level, which is also the same level as the one in the manual initialisation case.

5. (a) Estimate for transition probabilities:

$$\psi(\alpha, \beta) = \frac{\#\text{ pairs of symbol }(\alpha, \beta)}{\#\text{ symbol }\beta} \tag{30}$$

The transition table is splited to 3 sub-tables and are displayed below as Table 2, Table 3 and Table 4.

Estimate for stationary distribution:

$$\psi(\gamma) = \frac{\#\text{ symbol }\gamma}{\#\text{ all symbols}} \tag{31}$$

The stationary table is displayed below as Table 5.

(b) The latent variables $\sigma(s)$ are not independent, since we assume that the mapping between symbols is one-to-one. Specifically, if $\sigma(s_i) = a$, then $\sigma(s_j) \neq a$, where $s_j \neq s_i$. Thus not independent.

$$\begin{aligned}\mathbb{P}(e_1, e_2, \cdots, e_n|\sigma) &= \mathbb{P}(e_n|e_{n-1}, \cdots, e_2, e_1, \sigma)\mathbb{P}(e_{n-1}, \cdots, e_2, e_1|\sigma) \\ &= \mathbb{P}(\sigma(s_n)|\sigma(s_{n-1}), \cdots, \sigma(s_2), \sigma(s_1), \sigma)\mathbb{P}(\sigma(s_{n-1}), \cdots, \sigma(s_2), \sigma(s_1)|\sigma)\end{aligned} \tag{32}$$

Given that s_n depends only on s_{n-1}, and σ is an one-to-one map, thus $\sigma(s)$ also depends only on $\sigma(s_{n-1})$. Thus by continuously applying Markov property:

$$\begin{aligned}\mathbb{P}(e_1, e_2, \cdots, e_n|\sigma) &= \mathbb{P}(\sigma(s_n)|\sigma(s_{n-1}), \sigma)\mathbb{P}(\sigma(s_{n-1}), \cdots, \sigma(s_2), \sigma(s_1)|\sigma) \\ &= \mathbb{P}(\sigma(s_1)|\sigma)\prod_{i=2}^{n}\mathbb{P}(\sigma(s_i)|\sigma(s_{i-1}), \sigma)\end{aligned} \tag{33}$$

(c) The proposal probability is:

$$S(\sigma \to \sigma') = \frac{1}{{}^nP_2} = \frac{1}{n(n-1)} = \frac{1}{53 \times 52}, \tag{34}$$

which does not depend on σ. Thus noticeably, $S(\sigma \to \sigma') = S(\sigma' \to \sigma)$.

Given that we have a uniform prior, our acceptance probability is:

$$\begin{aligned}\alpha &= \min\{1, \frac{\mathbb{P}(\sigma'|s_1, \cdots, s_n)\mathbb{P}(\mathbf{s})}{\mathbb{P}(\sigma|s_1, \cdots, s_n)\mathbb{P}(\mathbf{s})} \times \frac{S(\sigma' \to \sigma)}{S(\sigma \to \sigma')}\} \\ &= \min\{1, \frac{\mathbb{P}(\sigma'|s_1, \cdots, s_n)}{\mathbb{P}(\sigma|s_1, \cdots, s_n)}\},\end{aligned} \tag{35}$$

where:

$$\begin{aligned}\mathbb{P}(\sigma|s_1, \cdots, s_n) \propto \mathbb{P}(s_1, \cdots, s_n|\sigma) \times 1 = \mathbb{P}(s_1|\sigma)\prod_{i=2}^{n}\mathbb{P}(s_i|s_{i-1}, \sigma) \\ \mathbb{P}(\sigma'|s_1, \cdots, s_n) \propto \mathbb{P}(s_1, \cdots, s_n|\sigma') \times 1 = \mathbb{P}(s_1|\sigma')\prod_{i=2}^{n}\mathbb{P}(s_i|s_{i-1}, \sigma')\end{aligned} \tag{36}$$

(d) **Initialisation method:**

By analysing the stationary distribution calculated in question (a), we discover that "⟨space⟩" and "e" are the most frequently appearing symbols. Then by finding the 2 most frequently appearing symbols ("p" and "?") in the encrypted text as well, and let σ^{-1} map ("p" and "?") to (" " and "e") respectively, while other symbols are mapped to themselves, we get our initialisation.

Final result for decoding the first 60 symbols:

"in my younger and more vulnerable years my father gave me so"

Code output print:

```
100: wl (, ,pxlsey bln (pye mxuleyb?ue ,ebyt (, .brkey sbme (e tp
200: nd z, ,tudsey odw ztye iubdeyo:be ,eoyp z, .orkey soie ze pt
300: nd cm mtudsei odw ctie kuadeio:ae meoip cm .oryei soke ce pt
400: nd cm mtudsei od. ctie kurdeiowre meoip cm :oayei soke ce pt
500: nd cm mtudsei od. ctie kurdeiowre meoip cm voayei soke ce pt
600: nd cm moudsei ad. coie kurdeiawre meaip cm vatyei sake ce po
700: nd cm moudlei ad. coie kurdeiawre meais cm vatyei lake ce so
800: nd cm moudlei ad. coie kurdeiawre meais cm vatyei lake ce so
900: nd cm moudlei ad. coie kurdeiawre meais cm vatyei lake ce so
1000: nd cm moudyei ad. coie kurdeiawre meais cm vatlei yake ce so
1100: nd cm moudyei ad. coie vurdeiawre meais cm katlei yave ce so
1200: nd ch houdyei adg coie kurdeialre heais ch vatwei yake ce so
1300: nd cy youdhei adg coie kurdeialre yeais cy vatwei hake ce so
1400: nd ly youdhei adg loie kurdeiafre yeais ly vatwei hake le so
1500: nd ly youdhei adg loie kurdeiafre yeais ly vatwei hake le so
1600: nd hy youdlei adg hoie kurdeiafre yeais hy vatwei lake he so
1700: nd hy youdlem adg home kurdemafre yeams hy vatiem lake he so
1800: nd hy youdlem adg home kurdemavre yeams hy fatiem lake he so
1900: id hy youdker adg hore vulderanle years hy fatmer kave he so
2000: id hy youd,er adg hore vulderanle years hy fatmer ,ave he so
2100: id hy youd.er adg hore vulderanle years hy fatmer .ave he so
2200: id hy youd.er adg hore vulderanle years hy fatmer .ave he so
2300: in hy youn.er ang hore vulneraple yeart hy fasmer .ave he to
2400: in hy youn.er ang hore vulneraple yeart hy fasmer .ave he to
2500: in hy youn.er ang hore vulneraple yeart hy fasmer .ave he to
2600: in hy youn.er ang hore vulneraple yeart hy fasmer .ave he to
2700: in hy youn.er ang hore vulneramle yeart hy fasper .ave he to
2800: in hy youn.er ang hore vulneramle yeart hy fasper .ave he to
2900: in my youn.er ang more vulnerahle yeart my fasper .ave me to
3000: in my youn.er ang more vulnerahle yeart my fasper .ave me to
3100: in my youn.er and more vulnerahle yeart my fasper .ave me to
3200: in my youn.er and more vulnerahle yeart my fasper .ave me to
3300: in my youn.er and more vulnerahle yeart my fasper .ave me to
3400: in my youn.er and more vulnerahle yeart my fasper .ave me to
3500: in my youn.er and more vulnerahle yeart my fasper .ave me to
3600: in my youn.er and more vulnerahle yeart my fasper .ave me to
3700: in my youn.er and more vulnerahle yeart my fasper .ave me to
3800: in my youn.er and more vulnerahle yeart my fasper .ave me to
3900: in my youn.er and more vulneraple yeart my fasher .ave me to
4000: in my youn.er and more vulneraple yeart my fasher .ave me to
4100: in my youn.er and more vulneraple yeart my fasher .ave me to
4200: in my youn.er and more vulneraple yeart my fasher .ave me to
4300: in my youn.er and more vulneraple yeart my fasher .ave me to
4400: in my youn.er and more vulneraple yeart my fasher .ave me to
4500: in my youn.er and more vulneraple yeart my fasher .ave me to
```

```
4600: in my youn.er and more vulneraple yeart my fasher .ave me to
4700: in my youn.er and more vulneraple yeart my fasher .ave me to
4800: in my youn.er and more vulnerable yeart my fasher .ave me to
4900: in my youn.er and more vulnerable yeart my fasher .ave me to
5000: in my younker and more vulnerable yeart my fasher kave me to
5100: in my younker and more vulnerable yeart my fasher kave me to
5200: in my younker and more vulnerable yeart my fasher kave me to
5300: in my younker and more vulnerable yeart my fasher kave me to
5400: in my younker and more vulnerable yeart my fasher kave me to
5500: in my younker and more vulnerable yeart my fasher kave me to
5600: in my younker and more vulnerable yeart my fasher kave me to
5700: in my younker and more vulnerable yeart my fasher kave me to
5800: in my younker and more vulnerable yeart my fasher kave me to
5900: in my younker and more vulnerable yeart my fasher kave me to
6000: in my younker and more vulnerable yeart my fasher kave me to
6100: in my younker and more vulnerable yeart my fasher kave me to
6200: in my younker and more vulnerable yeart my fasher kave me to
6300: in my younker and more vulnerable yeart my fasher kave me to
6400: in my younker and more vulnerable yeart my fasher kave me to
6500: in my younker and more vulnerable yeart my fasher kave me to
6600: in my younker and more vulnerable yeart my fasher kave me to
6700: in my younger and more vulnerable yeart my fasher gave me to
6800: in my younger and more vulnerable yeart my fasher gave me to
6900: in my younger and more vulnerable years my father gave me so
7000: in my younger and more vulnerable years my father gave me so
7100: in my younger and more vulnerable years my father gave me so
7200: in my younger and more vulnerable years my father gave me so
7300: in my younger and more vulnerable years my father gave me so
7400: in my younger and more vulnerable years my father gave me so
7500: in my younger and more vulnerable years my father gave me so
7600: in my younger and more vulnerable years my father gave me so
7700: in my younger and more vulnerable years my father gave me so
7800: in my younger and more vulnerable years my father gave me so
7900: in my younger and more vulnerable years my father gave me so
8000: in my younger and more vulnerable years my father gave me so
8100: in my younger and more vulnerable years my father gave me so
8200: in my younger and more vulnerable years my father gave me so
8300: in my younger and more vulnerable years my father gave me so
8400: in my younger and more vulnerable years my father gave me so
8500: in my younger and more vulnerable years my father gave me so
8600: in my younger and more vulnerable years my father gave me so
8700: in my younger and more vulnerable years my father gave me so
8800: in my younger and more vulnerable years my father gave me so
8900: in my younger and more vulnerable years my father gave me so
9000: in my younger and more vulnerable years my father gave me so
9100: in my younger and more vulnerable years my father gave me so
9200: in my younger and more vulnerable years my father gave me so
9300: in my younger and more vulnerable years my father gave me so
9400: in my younger and more vulnerable years my father gave me so
9500: in my younger and more vulnerable years my father gave me so
9600: in my younger and more vulnerable years my father gave me so
9700: in my younger and more vulnerable years my father gave me so
9800: in my younger and more vulnerable years my father gave me so
9900: in my younger and more vulnerable years my father gave me so
10000: in my younger and more vulnerable years my father gave me so
```

(e) $\psi(\alpha, \beta) = 0$ will affect the ergodicity of the chain. $\psi(\alpha, \beta) = 0$ means that the transition from β to α is impossible. If a chain does the map from β to α, then the log joint distribution of this map is negative infinity, and thus this map is not accepted. To restore the ergodicity, we can consider adding a small bias.

(f)
1. **Would symbol probability alone be sufficient?**
   Symbol probability alone is not sufficient. For instance, if 2 symbols have similar probabilities, the algorithm may not distinguish the 2 symbols.
2. **If we used a second order Markov chain for English text, what problems might we encounter?**
   If a second-order Markov chain is applied, a 3-D matrix will be used to describe the transition probabilities. This may increase the computational time compared with a 2-D matrix. On the positive side, the decoding may become more accurate.
3. **Will it work if the encryption scheme allows two symbols to be mapped to the same encrypted value?**
   The algorithm will also work in the case of two-to-one mapping, as long as the mappings are probabilistic. However, it may create larger state space (there are more combinations), thus increasing the computational time and the built-in phase.
4. **Would it work for Chinese with $>$ 10000 symbols?**
   The algorithm may not work, since the extremely large state space requires much longer computational time and much longer burn-in phase.

The transition probability tables for question 5(a):

	=		-	,	;	:	!	?	/	.	’	”	(	)	[	]	*
=	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0.003	0	0	0	0	0	0	0	0	0	0	0.001	0	0	0	0.001
-	0	0.009	0.003	0	0	0	0	0	0	0	0	0	0	0	0	0	0
”,”	0	0.926	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
;	0	0.927	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
:	0	0.916	0	0	0	0	0	0	0	0.001	0	0	0.002	0	0	0	0.002
!	0	0.85	0	0	0	0	0	0	0	0.096	0	0	0	0.002	0	0	0.001
?	0	0.843	0	0	0	0	0	0	0	0.083	0	0	0.001	0.001	0	0	0
/	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0.659	0	0	0	0	0	0.001	0	0.096	0	0	0.001	0.003	0	0	0
’	0	0	0	0.143	0	0	0	0	0	0	0	0	0	0	0	0	0
””””	0	0.409	0	0	0	0	0	0	0	0	0	0	0	0.045	0	0	0.045
(	0	0	0	0	0	0	0	0	0	0	0	0.002	0	0	0	0	0
)	0	0.618	0	0.265	0.008	0.002	0	0	0	0.056	0	0	0.002	0	0	0	0
[	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0
*	0	0.89	0	0	0	0	0	0	0	0	0	0	0.057	0	0	0	0.014
0	0	0.258	0	0.069	0	0	0	0	0	0.006	0	0	0	0.006	0	0.006	0
1	0	0.044	0	0.01	0	0	0	0	0	0.121	0	0	0.003	0.046	0	0	0
2	0	0.234	0	0.092	0	0	0	0.007	0	0.092	0	0	0	0.277	0	0.007	0
3	0	0.207	0	0.103	0	0	0	0	0	0.121	0	0	0	0.276	0	0	0
4	0	0.174	0.043	0.087	0	0	0	0	0	0.261	0	0	0	0.13	0	0	0
5	0	0.16	0.02	0.32	0.02	0.02	0	0	0.02	0.16	0	0	0	0.02	0	0	0
6	0	0.217	0.043	0.174	0.022	0	0	0.022	0	0.109	0	0	0	0	0	0	0
7	0	0.342	0	0.184	0	0	0.026	0	0	0.105	0	0	0	0	0	0	0
8	0	0.036	0	0.01	0	0	0	0	0	0.021	0	0	0	0	0	0	0
9	0	0.419	0	0.161	0	0.032	0	0	0	0.161	0	0	0	0	0	0	0
a	0	0.061	0	0.005	0	0	0.001	0	0	0.003	0	0	0	0	0	0	0
b	0	0.003	0	0.001	0	0	0	0	0	0.001	0	0	0	0	0	0	0
c	0	0.008	0	0.001	0	0	0	0	0	0	0	0	0	0	0	0	0
d	0	0.537	0.002	0.036	0.001	0.002	0.003	0.002	0	0.024	0	0	0	0	0	0	0
e	0	0.292	0.001	0.02	0.001	0.001	0.002	0.002	0	0.014	0	0	0	0	0	0	0
f	0	0.311	0.003	0.009	0	0	0.001	0.001	0	0.008	0	0	0	0	0	0	0
g	0	0.348	0.002	0.029	0.001	0.001	0.003	0.003	0	0.021	0	0	0	0	0	0	0
h	0	0.074	0	0.008	0	0	0.001	0.001	0	0.003	0	0	0	0	0	0	0
i	0	0.026	0	0.001	0	0	0	0	0	0.001	0	0	0	0	0	0	0
j	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
k	0	0.187	0.002	0.028	0.001	0.001	0.003	0.003	0	0.02	0	0	0	0	0	0	0
l	0	0.102	0.001	0.015	0	0	0.001	0.001	0	0.006	0	0	0	0	0	0	0
m	0	0.138	0	0.024	0.001	0.001	0.002	0.002	0	0.027	0	0	0	0	0	0	0
n	0	0.179	0.001	0.017	0	0	0.001	0.001	0	0.011	0	0	0	0	0	0	0
o	0	0.124	0	0.004	0	0	0.001	0.001	0	0.002	0	0	0	0	0	0	0
p	0	0.05	0	0.008	0	0	0.001	0	0	0.006	0	0	0	0	0	0	0
q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	0.164	0.001	0.019	0.001	0	0.002	0.002	0	0.014	0	0	0	0	0	0	0
s	0	0.296	0	0.038	0.001	0.001	0.003	0.002	0	0.021	0	0	0	0	0	0	0
t	0	0.217	0	0.015	0	0	0.002	0.002	0	0.011	0	0	0	0	0	0	0
u	0	0.046	0	0.005	0	0	0.002	0.001	0	0.003	0	0	0	0	0	0	0
v	0	0.06	0	0.022	0	0	0.001	0.001	0	0.011	0	0	0	0	0	0	0
w	0	0.103	0	0.017	0	0	0.002	0.001	0	0.009	0	0	0	0	0	0	0
x	0	0.055	0.002	0.009	0	0.001	0	0	0	0.004	0	0	0	0	0	0	0
y	0	0.515	0.003	0.069	0.002	0.002	0.006	0.005	0	0.047	0	0	0	0.001	0	0	0
z	0	0.032	0.002	0.014	0	0	0.001	0.001	0	0.006	0	0	0	0	0	0	0

Table 2: The transition table 1

0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	g	h
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.123	0.044	0.037	0.03	0.02	0.038	0.017	0.089
0	0.001	0	0	0	0	0.001	0	0.001	0	0.027	0.063	0.096	0.083	0.043	0.071	0.019	0.039
0	0	0	0	0	0	0	0	0	0	0.016	0.005	0.003	0.002	0.002	0.002	0.002	0.004
0	0	0	0	0	0	0	0	0	0	0.01	0.007	0.001	0	0.002	0.001	0.001	0.009
0	0.002	0.001	0	0	0	0	0	0	0	0.009	0.001	0.001	0.004	0.002	0.001	0	0.01
0	0	0	0	0	0	0	0	0	0	0.007	0.002	0.001	0.001	0.002	0.001	0.001	0.003
0	0	0	0	0	0	0	0	0	0	0.004	0.004	0.001	0.002	0.001	0.001	0	0.009
0	0	0	0	0.111	0	0	0	0	0	0	0	0.222	0.222	0	0	0	0.111
0	0.001	0	0	0	0	0	0	0	0	0.028	0.014	0.016	0.005	0.004	0.006	0.001	0.017
0	0	0	0	0	0	0	0	0	0	0.143	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0.045	0	0	0	0
0	0.026	0.059	0.023	0.005	0.002	0	0	0.002	0	0.124	0.024	0.008	0.011	0.011	0.021	0.006	0.115
0	0	0	0	0	0	0	0	0	0	0.011	0.003	0	0	0	0.002	0	0.003
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.007	0	0.004	0.004	0	0	0	0
0.113	0.025	0.019	0	0	0.17	0.075	0.094	0.031	0.107	0	0	0.013	0	0	0	0	0
0.038	0.041	0.159	0.051	0.005	0.023	0.008	0.021	0.426	0	0	0	0.003	0	0	0	0	0
0.085	0.014	0.035	0.014	0.021	0.021	0.014	0.014	0	0	0.007	0	0.05	0	0	0	0	0
0.121	0.017	0	0	0	0	0	0	0	0	0	0	0.017	0	0	0	0	0
0.087	0.087	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.12	0	0	0	0.02	0	0	0	0	0.02	0	0	0.06	0	0	0	0	0
0.109	0	0.043	0	0.022	0	0.13	0.022	0	0	0	0	0.022	0	0	0	0	0
0.079	0.026	0	0	0	0	0	0	0.105	0.026	0	0	0.026	0	0	0	0	0
0.411	0.443	0.021	0	0.005	0	0.005	0.005	0.005	0.016	0	0	0	0	0	0	0	0
0.097	0	0	0	0	0	0.032	0.032	0	0	0	0.032	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.017	0.035	0.056	0	0.008	0.017	0.002
0	0	0	0	0	0	0	0	0	0	0.092	0.007	0	0.001	0.329	0	0	0
0	0	0	0	0	0	0	0	0	0	0.109	0	0.017	0	0.213	0	0	0.188
0	0	0	0	0	0	0	0	0	0	0.025	0.003	0.002	0.013	0.118	0.002	0.004	0.004
0	0	0	0	0	0	0	0	0	0	0.045	0.002	0.019	0.093	0.026	0.011	0.007	0.004
0	0	0	0	0	0	0	0	0	0	0.073	0.001	0.001	0	0.084	0.052	0.001	0.003
0	0	0	0	0	0	0	0	0	0	0.069	0.001	0.001	0.002	0.115	0.001	0.009	0.12
0	0	0	0	0	0	0	0	0	0	0.166	0.001	0.001	0.001	0.45	0.001	0	0.001
0	0	0	0	0	0	0	0	0	0	0.017	0.007	0.05	0.051	0.049	0.022	0.025	0
0	0	0	0	0	0	0	0	0	0	0.037	0	0	0	0.22	0	0	0
0	0	0	0	0	0	0	0	0	0	0.027	0.001	0.003	0	0.284	0.001	0.001	0.042
0	0	0	0	0	0	0	0	0	0	0.08	0.001	0.001	0.073	0.175	0.024	0.001	0.001
0	0	0	0	0	0	0	0	0	0	0.153	0.018	0.001	0	0.248	0.003	0	0.001
0	0	0	0	0	0	0	0	0	0	0.032	0.001	0.051	0.191	0.082	0.006	0.138	0.002
0	0	0	0	0	0	0	0	0	0	0.008	0.005	0.008	0.016	0.003	0.089	0.004	0.003
0	0	0	0	0	0	0	0	0	0	0.099	0	0.003	0	0.183	0.002	0	0.01
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.056	0.002	0.011	0.03	0.243	0.004	0.01	0.002
0	0	0	0	0	0	0	0	0	0	0.055	0.004	0.015	0.001	0.11	0.003	0.001	0.069
0	0	0	0	0	0	0	0	0	0	0.035	0.001	0.003	0.001	0.087	0.001	0	0.334
0	0	0	0	0	0	0	0	0	0	0.021	0.014	0.032	0.022	0.03	0.005	0.051	0.001
0	0	0	0	0	0	0	0	0	0	0.065	0	0	0	0.552	0	0	0
0	0	0	0	0	0	0	0	0	0	0.207	0	0.001	0.006	0.132	0.001	0	0.203
0	0	0	0	0	0	0	0	0	0	0.08	0	0.14	0	0.063	0.001	0	0.015
0	0	0	0	0	0	0	0	0	0	0.031	0.005	0.006	0.002	0.057	0.004	0.001	0.003
0	0	0	0	0	0	0	0	0	0	0.044	0	0	0.013	0.326	0	0	0.069

Table 3: The transition table 2

i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.053	0.003	0.007	0.023	0.034	0.026	0.061	0.032	0.002	0.027	0.072	0.156	0.01	0.007	0.071	0.001	0.011	0
0.049	0.001	0.02	0.065	0.037	0.037	0.051	0.028	0.001	0.029	0.09	0.072	0.016	0.003	0.024	0	0.018	0.007
0.003	0	0.001	0.002	0.002	0.002	0.002	0.003	0	0.002	0.005	0.007	0.001	0	0.007	0	0.001	0
0.007	0	0	0.001	0.001	0.001	0.004	0.002	0	0	0.006	0.018	0	0	0.003	0	0.002	0
0.007	0	0	0.001	0.004	0	0.002	0.005	0	0	0.009	0.015	0	0	0.004	0	0	0
0.004	0.001	0.002	0.002	0.003	0.002	0.001	0.001	0	0	0.002	0.007	0	0.001	0.007	0	0.003	0
0.013	0	0	0.001	0.001	0.005	0.002	0.001	0	0	0.001	0.011	0	0	0.011	0	0.003	0
0	0	0	0.111	0	0	0	0	0	0	0	0.111	0	0	0	0	0.111	0
0.015	0.002	0.002	0.002	0.003	0.012	0.01	0.017	0	0.004	0.013	0.051	0	0	0.013	0	0.001	0
0	0	0	0	0	0	0	0	0	0	0.714	0	0	0	0	0	0	0
0.045	0	0	0	0	0	0	0.318	0	0.045	0	0.045	0	0	0	0	0	0
0.065	0.002	0.008	0.009	0.018	0.026	0.05	0.035	0.002	0.006	0.067	0.152	0.003	0.005	0.117	0	0.002	0
0.005	0	0	0	0.002	0	0.005	0.003	0	0	0.002	0.011	0	0	0.005	0	0	0
0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.004	0	0.004	0	0	0	0	0.004	0	0	0	0	0.004	0	0.007	0	0.004	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.006	0	0	0
0	0	0	0	0	0	0	0	0	0	0.003	0	0	0	0	0	0	0
0	0	0.007	0	0	0.007	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0.086	0	0.052	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.087	0	0	0	0.043	0	0
0	0	0	0	0	0	0	0	0	0	0	0.02	0	0	0	0	0.02	0
0	0	0	0	0	0	0	0	0	0	0	0.065	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.079	0	0	0	0	0	0
0	0	0	0	0	0	0.005	0.005	0	0	0	0.01	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.032	0	0	0	0	0	0
0.043	0.001	0.013	0.07	0.023	0.225	0	0.023	0	0.088	0.097	0.139	0.012	0.021	0.011	0	0.026	0.002
0.029	0.005	0	0.105	0.002	0	0.118	0	0	0.057	0.014	0.007	0.153	0.001	0.001	0	0.074	0
0.042	0	0.043	0.033	0	0	0.208	0	0.002	0.034	0.002	0.067	0.027	0	0	0	0.005	0
0.068	0.002	0.001	0.012	0.004	0.004	0.044	0.002	0.001	0.032	0.022	0.007	0.01	0.003	0.003	0	0.01	0
0.012	0	0.001	0.033	0.024	0.084	0.006	0.012	0.001	0.142	0.065	0.026	0.001	0.017	0.012	0.01	0.012	0.001
0.086	0	0	0.023	0.001	0	0.148	0.001	0	0.104	0.004	0.045	0.034	0	0.002	0	0.002	0
0.05	0	0	0.029	0.001	0.019	0.062	0.001	0	0.055	0.018	0.009	0.027	0	0.002	0	0.002	0
0.155	0	0.001	0.001	0.002	0.001	0.082	0	0	0.008	0.001	0.026	0.01	0	0.001	0	0.005	0
0.002	0	0.007	0.046	0.057	0.281	0.043	0.005	0	0.035	0.123	0.124	0.001	0.02	0	0.002	0	0.003
0.004	0	0	0	0	0	0.278	0	0	0	0	0	0.461	0	0	0	0	0
0.17	0	0	0.017	0.002	0.104	0.014	0.001	0	0.005	0.038	0.002	0.034	0.001	0.005	0	0.005	0
0.099	0	0.011	0.138	0.004	0.001	0.087	0.004	0	0.005	0.017	0.018	0.015	0.005	0.005	0	0.105	0
0.08	0	0	0.003	0.023	0.004	0.109	0.057	0	0.002	0.032	0.003	0.027	0	0.001	0	0.04	0
0.032	0.001	0.008	0.012	0.001	0.011	0.07	0.001	0.001	0.001	0.035	0.091	0.005	0.004	0.001	0	0.012	0
0.012	0.001	0.018	0.039	0.059	0.141	0.038	0.017	0	0.104	0.036	0.054	0.124	0.03	0.053	0.001	0.004	0.001
0.094	0	0	0.094	0.001	0	0.106	0.059	0	0.175	0.024	0.049	0.026	0	0.001	0	0.008	0
0	0	0	0	0	0	0	0	0	0	0	0	0.999	0	0	0	0	0
0.089	0	0.006	0.009	0.021	0.018	0.101	0.004	0	0.034	0.056	0.033	0.02	0.005	0.003	0	0.039	0.001
0.058	0	0.014	0.008	0.011	0.004	0.055	0.021	0.001	0.001	0.062	0.112	0.025	0.001	0.006	0	0.002	0
0.064	0	0	0.014	0.002	0.001	0.108	0.001	0	0.026	0.019	0.022	0.014	0	0.006	0	0.014	0.001
0.027	0	0.002	0.098	0.021	0.137	0.002	0.043	0	0.125	0.137	0.171	0	0.001	0	0.001	0	0.001
0.18	0	0	0.008	0	0.021	0.056	0	0	0.004	0.01	0.001	0.001	0	0.001	0	0.004	0
0.17	0	0.001	0.004	0	0.038	0.078	0	0	0.009	0.013	0.001	0	0	0.001	0	0.001	0
0.12	0	0	0	0	0	0.003	0.317	0.001	0.001	0.001	0.113	0.004	0.037	0.001	0.03	0.001	0
0.026	0	0.001	0.005	0.005	0.002	0.134	0.003	0	0.003	0.032	0.022	0.002	0	0.005	0	0	0
0.108	0	0.002	0.02	0.028	0.008	0.283	0	0	0	0.001	0	0.012	0	0.001	0	0.006	0.021

Table 4: The transition table 3

The stationary probability table for question 5(a):

symbols	stationary_prob
=	0.0
	0.166
-	0.001
”,”	0.013
;	0.0
:	0.0
!	0.001
?	0.001
/	0.0
.	0.01
’	0.0
””””	0.0
(	0.0
)	0.0
[	0.0
]	0.0
*	0.0
0	0.0
1	0.0
2	0.0
3	0.0
4	0.0
5	0.0
6	0.0
7	0.0
8	0.0
9	0.0
a	0.065
b	0.011
c	0.02
d	0.038
e	0.1
f	0.018
g	0.016
h	0.054
i	0.055
j	0.001
k	0.007
l	0.031
m	0.02
n	0.059
o	0.061
p	0.015
q	0.001
r	0.048
s	0.052
t	0.073
u	0.021
v	0.009
w	0.019
x	0.001
y	0.015
z	0.001

Table 5: The stationary distribution table

6. (a) The 4 sample plots are as follows:

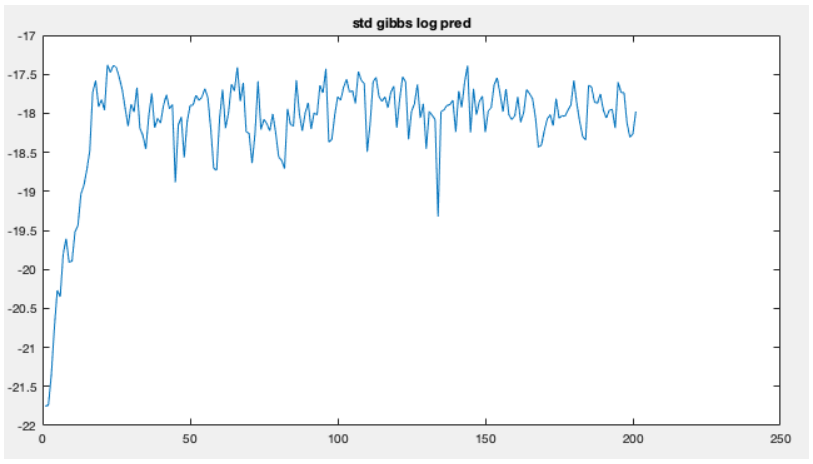

Figure 25: standard gibbs sampling log pred

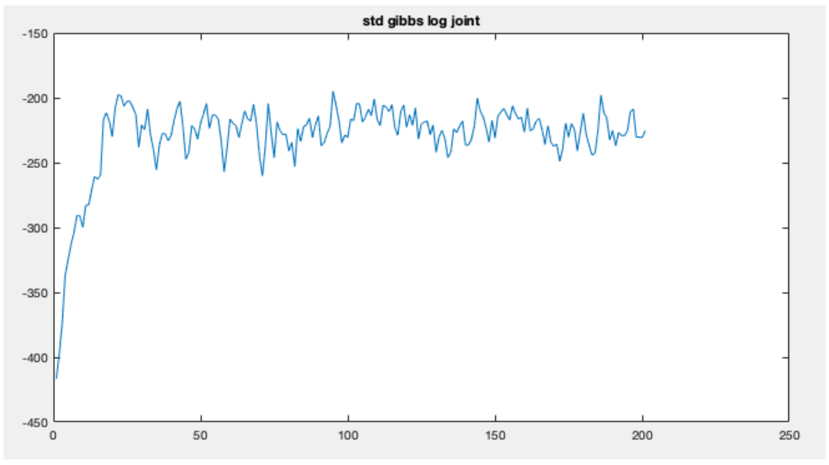

Figure 26: standard gibbs sampling log joint

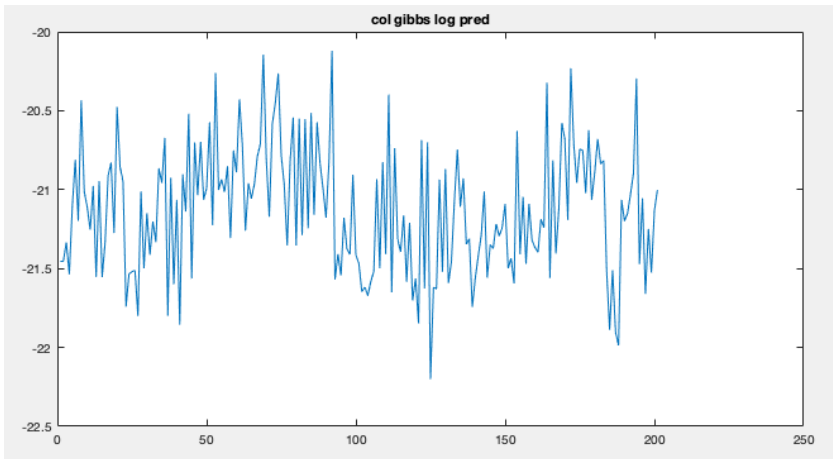

Figure 27: collapsed gibbs sampling log pred

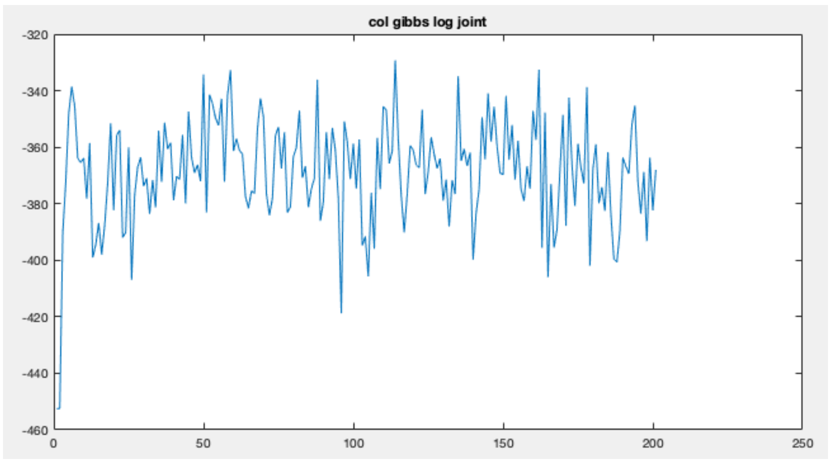

Figure 28: collapsed gibbs sampling log joint

The code is listed below:
standard Gibbs log-joint:

```
function L = stdgibbs_logjoint(theta,phi,Adk,Bkw,Mk,...
        I,D,K,W,di,wi,ci,citest,Id,Iw,Nd,alpha,beta);
% standard gibbs log joint probability over everything

prior_dk = D*(log(gamma(K*alpha)) - K*log(gamma(alpha)
   ));
prior_kw = K*(log(gamma(W*beta)) - W*log(gamma(beta)))
   ;
dk_ll = sum(sum((alpha - 1 + Adk) .* log(theta)));
kw_ll = sum(sum((beta - 1 + Bkw) .* log(phi)));
L = prior_dk + prior_kw + dk_ll + kw_ll;

end
```

collapsed Gibbs log-joint

```
function L = gibbs_loglik(Adk,Bkw,Mk,...
        I,D,K,W,di,wi,ci,citest,Id,Iw,Nd,alpha,beta);
% collapsed gibbs log joint probability over
   everything

prior_dk = D*(log(gamma(K*alpha)) - K*log(gamma(alpha)
   )) - sum(log(gamma(K*alpha+Nd)));
prior_kw = K*(log(gamma(W*beta)) - W*log(gamma(beta)))
    - sum(log(gamma(W*beta+Mk)));
dk_ll = sum(sum(log(gamma(alpha + Adk))));
kw_ll = sum(sum(log(gamma(beta + Bkw))));
L = prior_dk + prior_kw + dk_ll + kw_ll;

end
```

standard Gibbs update

```matlab
function [zi,theta,phi,Adk,Bkw,Mk] = stdgibbs_update(
   zi,theta,phi,Adk,Bkw,Mk,...
        I,D,K,W,di,wi,ci,citest,Id,Iw,Nd,alpha,beta);
% standard gibbs update

Adk = zeros(size(Adk));
Bkw = zeros(size(Bkw));

for ii = 1:I
    % conditional distribution
    pii = theta(di(ii),:)' .* phi(:,wi(ii));
    pii = pii / sum(pii);

    % resample zi{ii}
    s = mnrnd(1,pii,ci(ii));
    [~,index] = max(s,[],2);
    zi{ii} = index';

    % update Adk & Bkw
    for k=1:K
        Adk(di(ii),k)= Adk(di(ii),k) + sum(zi{ii}==k);
        Bkw(k,wi(ii))= Bkw(k,wi(ii)) + sum(zi{ii}==k);
    end
end

Nd = sum(Adk,2);
Mk = sum(Bkw,2);

% resample theta
for d=1:D
    theta(d,:) = dirichrnd(1,alpha + Adk(d,:));
end

% resample phi
for k=1:K
    phi(k,:) = dirichrnd(1,beta + Bkw(k,:));
end

end
```

collapsed Gibbs update

```matlab
function [zi,Adk,Bkw,Mk] = gibbs_update(zi,Adk,Bkw,Mk,
   ...
        I,D,K,W,di,wi,ci,citest,Id,Iw,Nd,alpha,beta);
% collapsed gibbs update

```

```
Adk = zeros(size(Adk));
Bkw = zeros(size(Bkw));

for ii = 1:I
    pii = zeros(1,K);
    for k=1:K
        % conditional prob
        pii(k) = (alpha+Adk(di(ii),k)-sum(zi{ii}==k))
            * (beta+Bkw(k,wi(ii))-sum(zi{ii}==k))...
            /(K*alpha+Nd(di(ii)) - sum(zi{ii}==k))/(W*
               beta+Mk(k) - sum(zi{ii}==k));
        % update Adk & Bkw
        Adk(di(ii),k)= Adk(di(ii),k)+sum(zi{ii}==k);
        Bkw(k,wi(ii))= Bkw(k,wi(ii))+sum(zi{ii}==k);
    end
    pii = pii/sum(pii);

    % resample zi{ii}
    s = mnrnd(1,pii,ci(ii));
    [~,index]=max(s,[],2);
    zi{ii} = index';
end

Nd = sum(Adk,2);
Mk = sum(Bkw,2);

end
```

(b) For standard Gibbs, the burn-in phase is roughly around 25 times, while for the collapsed Gibbs, the burn-in phase is slightly shorter, roughly around 15-20 times. By setting the iteration times to 1000, we obtain the following autocorrelation plots.

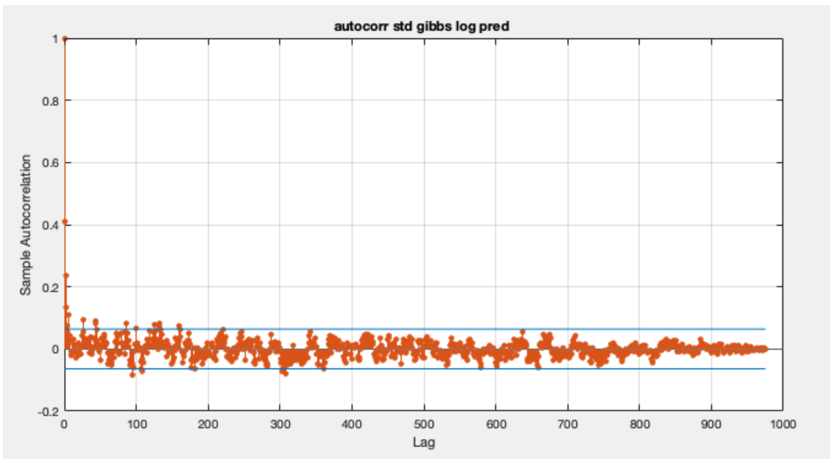

Figure 29: autocorr standard Gibbs sampling log pred

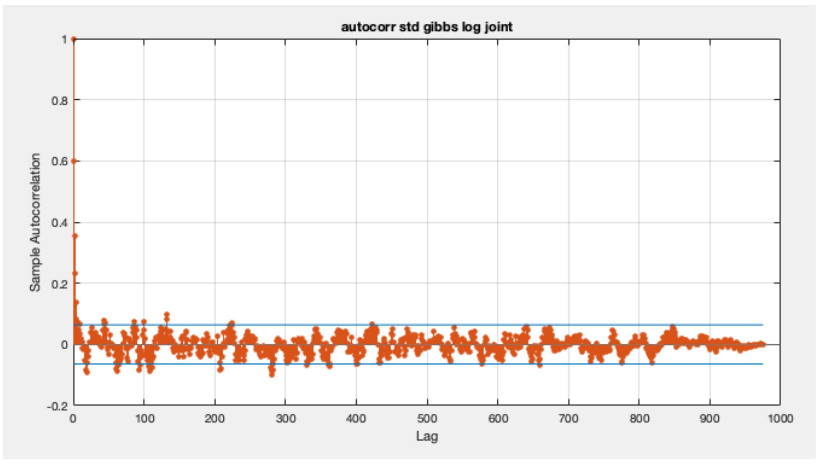

Figure 30: autocorr standard Gibbs sampling log joint

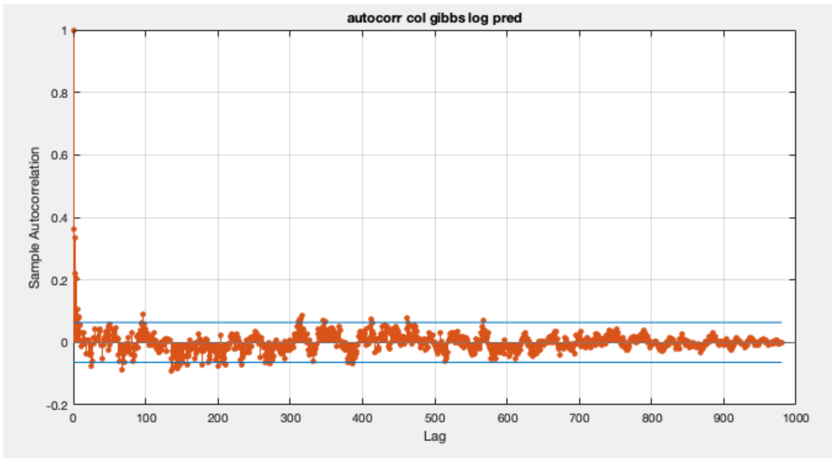

Figure 31: autocorr collapsed Gibbs sampling log pred

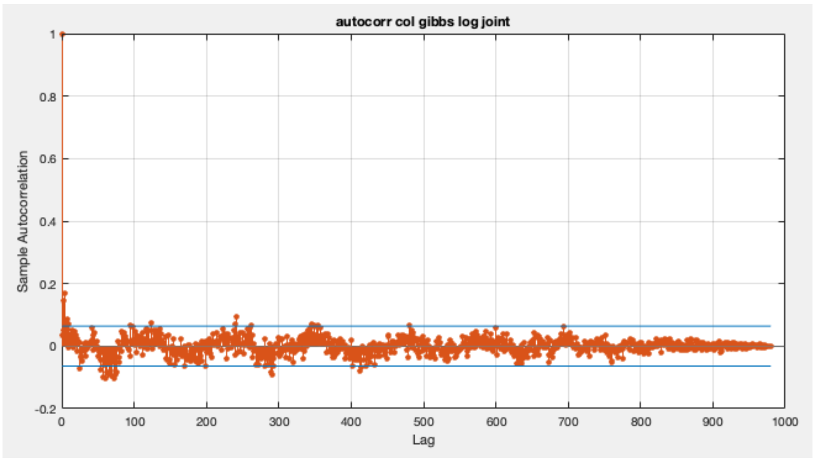

Figure 32: autocorr collapsed Gibbs sampling log joint

From the plots (especially Figure 32), we discover that there is a periodicity/seasonality roughly 100. We may infer at least 100 samples is needed.

(c) The plots generated in question (a) indicate that the collapsed Gibbs reach the flat stage faster than the standard Gibbs, while the plots from (b) reveal collapsed Gibbs also reach the steady stage faster with less fluctuation. Thus we may conclude that the collapsed Gibbs sampler is a better one, converging faster.

(d) We set α as $1, 4, 8$, fix β and K at the initial values, and generate the following plots:

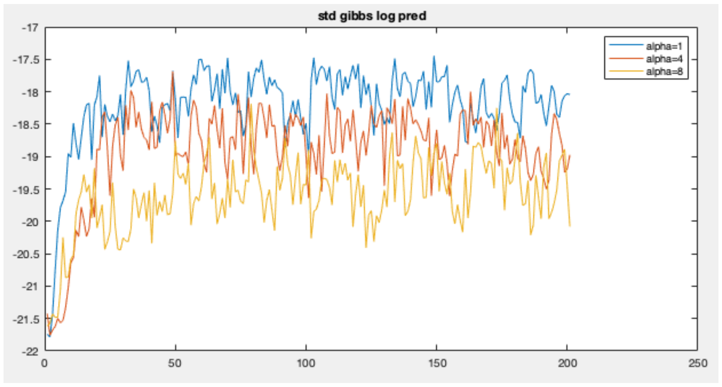

Figure 33: standard Gibbs sampling log pred

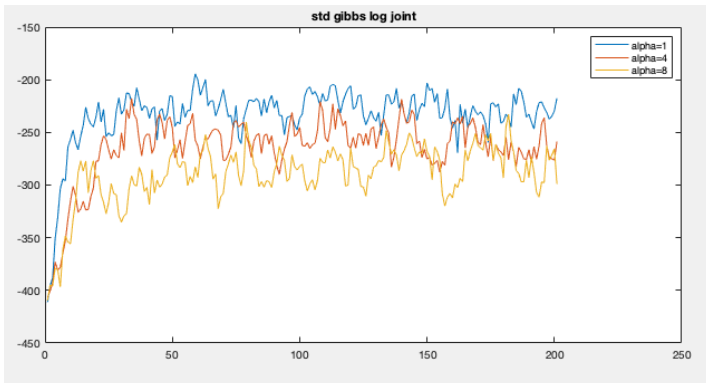

Figure 34: standard Gibbs sampling log joint

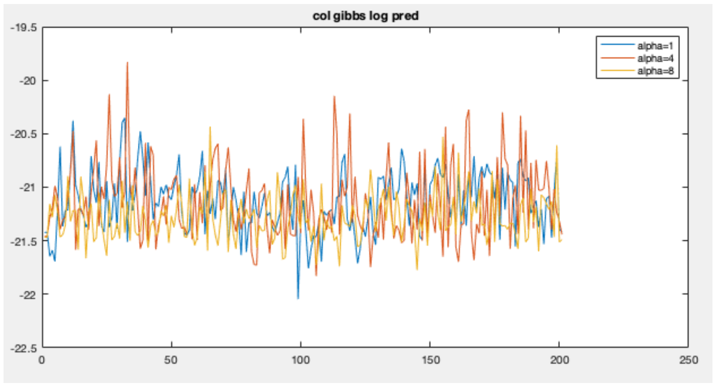

Figure 35: collapsed Gibbs sampling log pred

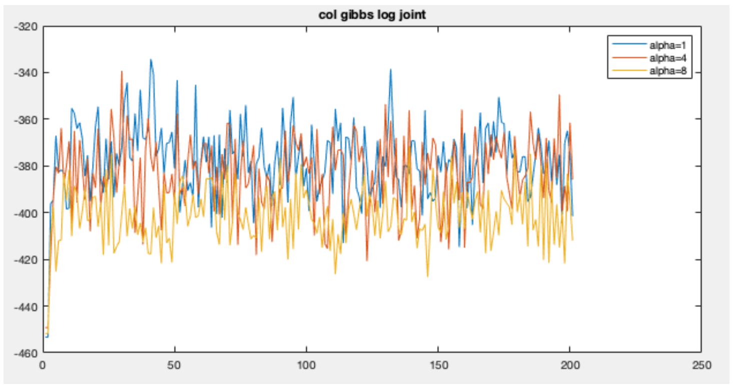

Figure 36: collapsed Gibbs sampling log joint

In predictive and log joint plots, we can tell smaller α tends to get larger likelihood. α provides priors for θ, where θ is the distribution over topics in a document. It makes sense that α takes smaller value, since a document may tend to focus on specifically limited topics, instead of trying to cover more.

We set $\beta = 1, 4, 8$ and fix α, K at the initial values, and generate the following plots:

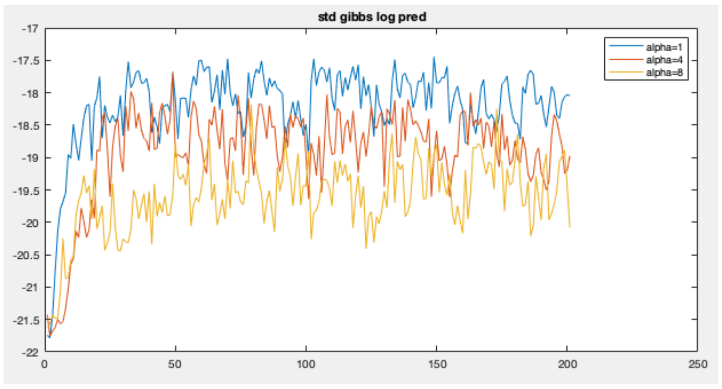

Figure 37: standard Gibbs sampling log pred

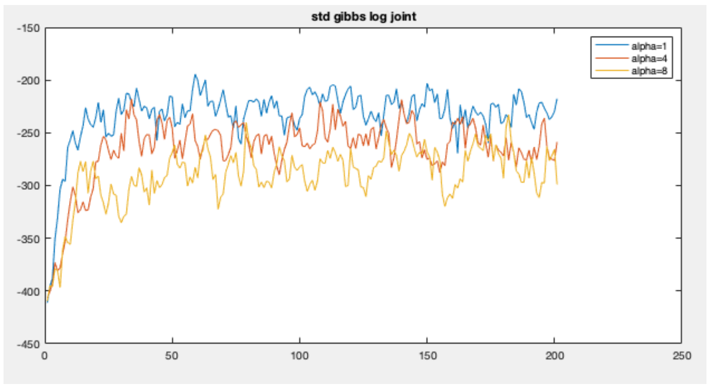

Figure 38: standard Gibbs sampling log joint

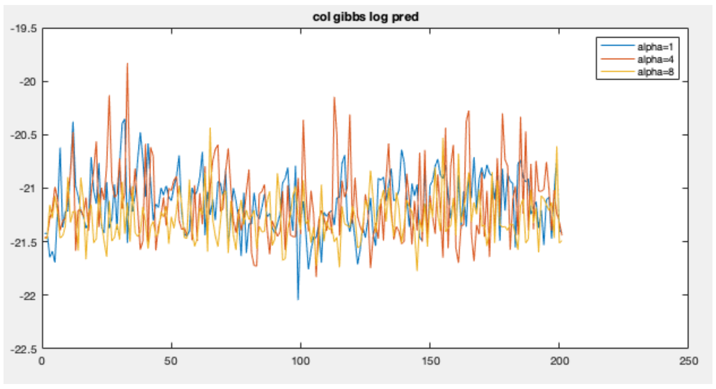

Figure 39: collapsed Gibbs sampling log pred

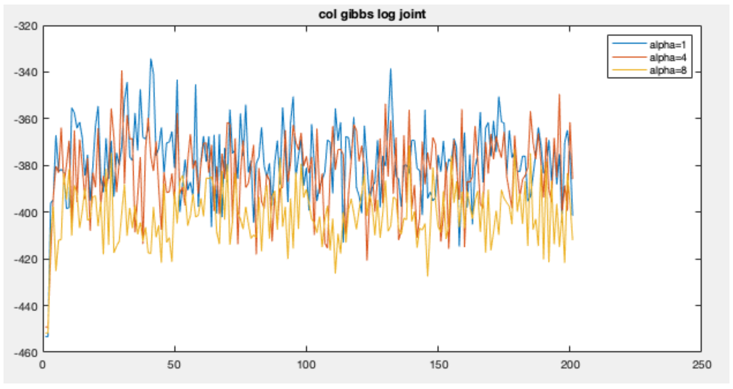

Figure 40: collapsed Gibbs sampling log joint

The case is quite similar as the one of α: smaller β tends to have larger likelihood for both predictive and joint cases.

7. (a) Define the Lagrangian as:

$$L(x, y, \lambda) = f(x, y) + \lambda g(x, y) = x + 2y + \lambda(y^2 + xy - 1)\,, \tag{37}$$

where λ is the Lagrange multiplier. To obtain the stationary points of the constrained system, we need:

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} = \frac{\partial L}{\partial \lambda} = 0\,. \tag{38}$$

Thus:

$$\begin{aligned} \frac{\partial L}{\partial x} &= 1 + \lambda y = 0 \\ \frac{\partial L}{\partial y} &= 2 + \lambda(2y + x) = 0 \\ \frac{\partial L}{\partial \lambda} &= y^2 + xy - 1 = 0 \end{aligned} \tag{39}$$

By solving the equations above, we obtain the local extrema:

1. $x = 0, y = 1, \lambda = -1$
2. $x = 0, y = -1, \lambda = 1$

(b) i. We need to find x such that $x = \ln(a)$, by re-organising the function, we get $e^x = a$. Define $f(x, a) = e^x - a$ and thus $f'(x) = e^x$.

ii. By applying Newton's method:

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \\ &= x_n - \frac{e^x - a}{e^x} \\ &= \frac{e^x(x_n - 1) + a}{e^x}\,, \end{aligned} \tag{40}$$

which is the update equation for computing $x = \ln(a)$.