CSC412: Assignment 2

Due on Friday, Mar 15, 2019

Zhongtian Ouyang
1002341012

(collaborate with Yihao Ni)

The code for this assignment is written in Python 3

Problem 1

(a)

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{10000} P(\mathbf{x}^{(i)}|c,\theta_c) = \prod_{i=1}^{10000}\prod_{d=1}^{784} \theta_{cd}^{x_d^{(i)}}(1-\theta_{cd})^{(1-x_d^{(i)})} \\
\hat{\theta}_{cd} &= \frac{num\ of\ data\ that\ are\ digit\ c\ and\ d^{th}\ pixel\ is\ present}{num\ of\ data\ that\ are\ digit\ c} \\
&= \frac{\sum_{i=1}^{10000} x_d^{(i)} \wedge (label^{(i)} == c)}{\sum_{i=1}^{10000}(label^{(i)} == c)}
\end{aligned} \tag{1}
$$

(b)

$$
argmax_{\hat{\theta}} \frac{P(data|c,\theta_c=\hat{\theta}_c)P(\theta_c=\hat{\theta}_c)}{P(data|c)} = argmax_{\hat{\theta}_c} P(data|c,\theta_c=\hat{\theta}_c)P(\theta_c=\hat{\theta}_c)
$$

Since the prior for each θ is Beta(2,2),

$$
f(\theta;2,2) = \frac{\theta^{2-1}(1-\theta)^{2-1}}{B(2,2)} = \frac{\theta(1-\theta)}{B(2,2)} \propto \theta(1-\theta)
$$

$$
P(data|c,\theta_c=\hat{\theta}_c) = \prod_{i=1}^{10000} P(\mathbf{x}^{(i)}|c,\theta_c=\hat{\theta}_c) = \prod_{i=1}^{10000}\prod_{d=1}^{784} \hat{\theta}_{c^{(i)}d}^{x_d^{(i)}}(1-\hat{\theta}_{c^{(i)}d})^{(1-x_d^{(i)})}
$$

We can find seperately the θ_{cd} value for each c, d that would maxmize the terms including that θ_{cd}.

For an arbitray c,d value, let N_c be the number of samples that are digit c. Let N_{cd} be the number of samples that are digit c and the dth pixel is white.

Collecting the terms from $P(data|c,\theta_c=\hat{\theta}_c)$, we get:

$$
\theta_{cd}^{N_{cd}}(1-\theta_{cd})^{N_c-N_{cd}}
$$

Collecting the terms from $P(\theta_c=\hat{\theta}_c)$, we get:

$$
\theta_{cd}(1-\theta_{cd})
$$

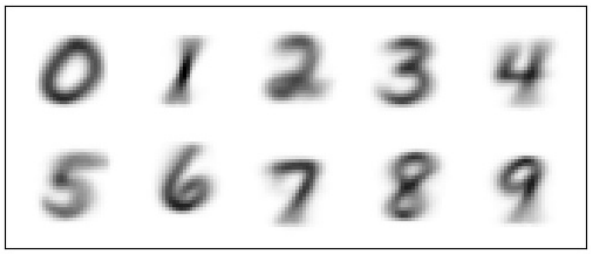
Let:

$$
\begin{aligned}
F &= \theta_{cd}^{N_{cd}}(1-\theta_{cd})^{N_c-N_{cd}}\theta_{cd}(1-\theta_{cd}) = \theta_{cd}^{N_{cd}+1}(1-\theta_{cd})^{N_c-N_{cd}+1} \\
log(F) &= (N_{cd}+1)log(\theta_{cd}) + (N_c-N_{cd}+1)log(1-\theta_{cd}) \\
\frac{d}{d\theta_{cd}}log(F) &= \frac{N_{cd}+1}{\theta_{cd}} - \frac{N_c-N_{cd}+1}{1-\theta_{cd}}
\end{aligned}
$$

Find $\hat{\theta}_{cd}$ by solving the following:

$$
\begin{aligned}
\frac{N_{cd}+1}{\hat{\theta}_{cd}} - \frac{N_c-N_{cd}+1}{1-\hat{\theta}_{cd}} &= 0 \\
\hat{\theta}_{cd} &= \frac{N_{cd}+1}{N_c+2}
\end{aligned}
$$

(c)



(d)

$$
\begin{aligned}
& log(p(c|\mathbf{x}, \boldsymbol{\theta}, \pi)) \\
& = log(\frac{p(\mathbf{x}, c|\boldsymbol{\theta}, \pi)}{p(\mathbf{x}|\boldsymbol{\theta})}) \\
& = log(\frac{p(c|\pi)p(\mathbf{x}|c, \theta_c)}{\sum_{c1=0}^{9} p(\mathbf{x}|c1, \theta_{c1})}) \\
& = log(p(c|\pi)) + log(p(\mathbf{x}|c, \theta_c)) - log(\sum_{c1=0}^{9} p(\mathbf{x}|c1, \theta_{c1})) \\
& = log(\pi_c) + log(\prod_{d=1}^{784} \theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}) - log(\sum_{c1=0}^{9}\prod_{d=1}^{784} \theta_{(c1)d}^{x_d}(1-\theta_{(c1)d})^{(1-x_d)}) \\
& = log(\pi_c) + \sum_{d=1}^{784}(x_d log(\theta_{cd}) + (1-x_d)log(1-\theta_{cd})) - log(\sum_{c1=0}^{9}\prod_{d=1}^{784} \theta_{(c1)d}^{x_d}(1-\theta_{(c1)d})^{(1-x_d)}) \\
& \propto log(\pi_c) + \sum_{d=1}^{784}(x_d log(\theta_{cd}) + (1-x_d)log(1-\theta_{cd}))
\end{aligned}
\tag{2}
$$

Notice that the last part $log(\sum_{c1=0}^{9} p(\mathbf{x}|c1, \theta_{c1}))$ is the same for any input argument c to the $log(p(c|\mathbf{x}, \boldsymbol{\theta}, \pi))$. We can therefore ignore this part when we making predictions to avoid the underflow problem.

(e)

Notice that the average of log likelihood here ignore the part mensioned above.
Training set Average log likelihood: -172.3538233466875

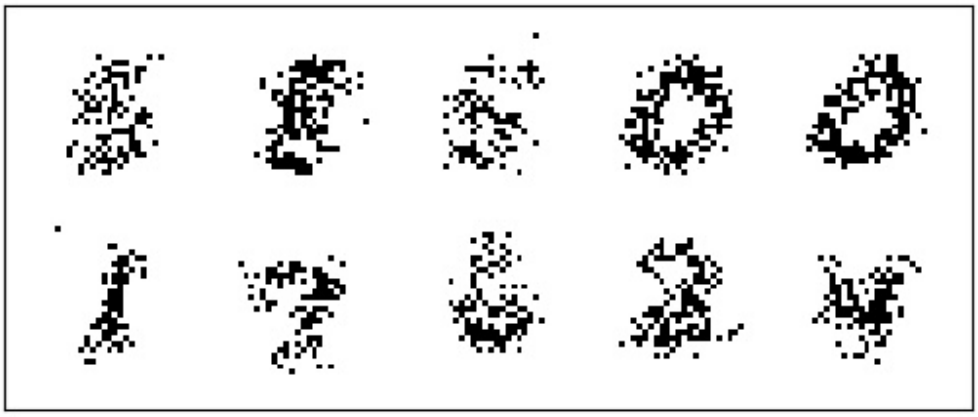
Test set Average log likelihood: -173.04360309120443
Training set accuracy: 0.8398
Test set accuracy: 0.8372

Problem 2

(a)
True

(b)
False

(c)
generated digits are:[5 8 5 0 0 1 7 6 2 4]. The digits are randomly generated with a fixed seed to make sure the result is reproducible.



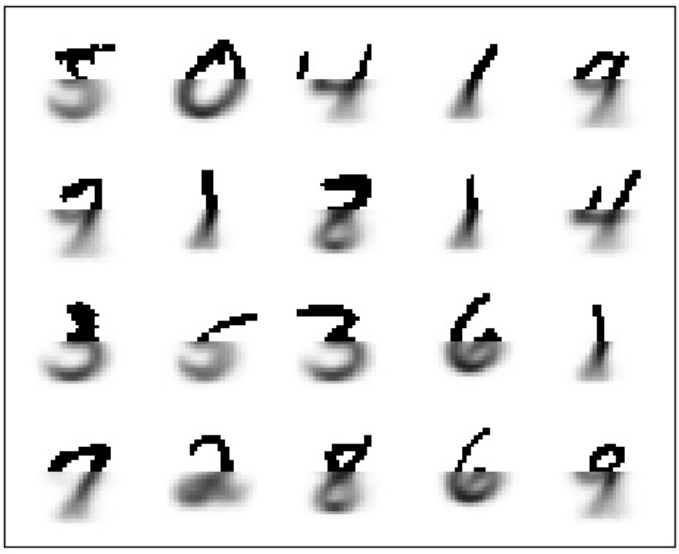
(d)

$$\begin{aligned} p(\mathbf{x}_{bottom}|\mathbf{x}_{top}, \boldsymbol{\theta}, \pi) &= \frac{p(\mathbf{x}_{bottom}, \mathbf{x}_{top}|\boldsymbol{\theta}, \pi)}{p(\mathbf{x}_{top}|\boldsymbol{\theta}, \pi)} \\ &= \frac{p(\mathbf{x}|\boldsymbol{\theta}, \pi)}{p(\mathbf{x}_{top}|\boldsymbol{\theta}, \pi)} \\ &= \frac{\sum_{c=0}^{9} p(\mathbf{x}|c, \boldsymbol{\theta}, \pi)}{\sum_{c=0}^{9} p(\mathbf{x}_{top}|c, \boldsymbol{\theta}, \pi)} \\ &= \frac{\sum_{c=0}^{9} \prod_{d=1}^{784} \theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}}{\sum_{c=0}^{9} \prod_{d=1}^{392} \theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}} \end{aligned} \tag{3}$$

(e)

$$\begin{aligned} p(\mathbf{x}_{i\in bottom}|\mathbf{x}_{top}, \boldsymbol{\theta}, \pi) &= \frac{p(\mathbf{x}_{i\in bottom}, \mathbf{x}_{top}|\boldsymbol{\theta}, \pi)}{p(\mathbf{x}_{top}|\boldsymbol{\theta}, \pi)} \\ &= \frac{\sum_{c=0}^{9} p(\mathbf{x}_{i\in bottom}, \mathbf{x}_{top}|c, \boldsymbol{\theta}, \pi)}{\sum_{c=0}^{9} p(\mathbf{x}_{top}|c, \boldsymbol{\theta}, \pi)} \\ &= \frac{\sum_{c=0}^{9} p(\mathbf{x}_{i\in bottom}|c, \boldsymbol{\theta}, \pi)p(\mathbf{x}_{top}|c, \boldsymbol{\theta}, \pi)}{\sum_{c=0}^{9} p(\mathbf{x}_{top}|c, \boldsymbol{\theta}, \pi)}, \ \textit{condionally independent} \\ &= \frac{\sum_{c=0}^{9} \theta_{ci}^{x_i}(1-\theta_{ci})^{(1-x_i)} \prod_{d=1}^{392} \theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}}{\sum_{c=0}^{9} \prod_{d=1}^{392} \theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}} \end{aligned} \tag{4}$$

(f)



Problem 3

(a)
784 weights for each digit, 10 digit classes in total. 784 * 10 = 7840 parameters.

(b)

$$\begin{aligned}\nabla_w log(p(c|\mathbf{x},\mathbf{w})) &= \nabla_w log(\frac{exp(\mathbf{w}_c^T\mathbf{x})}{\sum_{c'=0}^{9} exp(\mathbf{w}_{c'}^T\mathbf{x})}) \\ &= \nabla_w(log(exp(\mathbf{w}_c^T\mathbf{x})) - log(\sum_{c'=0}^{9} exp(\mathbf{w}_{c'}^T\mathbf{x}))) \\ &= \nabla_w(\mathbf{w}_c^T\mathbf{x} - log(\sum_{c'=0}^{9} exp(\mathbf{w}_{c'}^T\mathbf{x}))) = F\end{aligned} \tag{5}$$

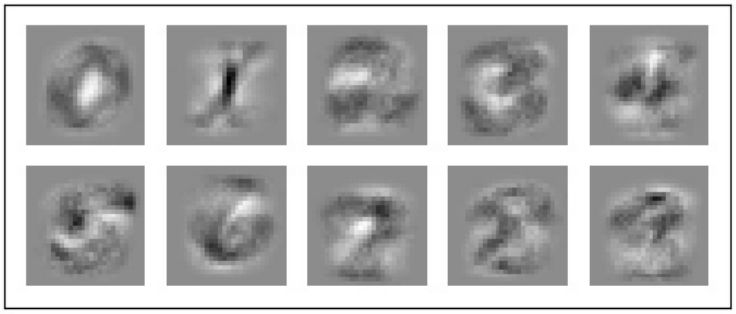
For c,

$$\begin{aligned}\frac{\partial F}{\partial w_{cd}} &= \frac{\partial}{\partial w_{cd}}(\mathbf{w}_c^T\mathbf{x} - log(\sum_{c'=0}^{9} exp(\mathbf{w}_{c'}^T\mathbf{x}))) \\ &= x_d - \frac{\partial}{\partial w_{cd}}(log(\sum_{c'=0}^{9} exp(\mathbf{w}_{c'}^T\mathbf{x}))) \\ &= x_d - (\frac{exp(\mathbf{w}_c^T\mathbf{x})x_d}{\sum_{c'=0}^{9} exp(\mathbf{w}_{c'}^T\mathbf{x})}) \\ &= x_d - x_d \cdot p(c|\mathbf{x},\mathbf{w})\end{aligned} \tag{6}$$

For $\tilde{c} \neq c$,

$$\begin{aligned}\frac{\partial F}{\partial w_{\tilde{c}d}} &= \frac{\partial}{\partial w_{\tilde{c}d}}(\mathbf{w}_c^T\mathbf{x} - log(\sum_{c'=0}^{9} exp(\mathbf{w}_{c'}^T\mathbf{x}))) \\ &= 0 - \frac{\partial}{\partial w_{\tilde{c}d}}(log(\sum_{c'=0}^{9} exp(\mathbf{w}_{c'}^T\mathbf{x}))) \\ &= 0 - (\frac{exp(\mathbf{w}_{\tilde{c}}^T\mathbf{x})x_d}{\sum_{c'=0}^{9} exp(\mathbf{w}_{c'}^T\mathbf{x})}) \\ &= -x_d \cdot p(\tilde{c}|\mathbf{x},\mathbf{w})\end{aligned} \tag{7}$$

(c)
Code in q3.py. Notice that here, it is identical to using softmax with cross-entropy loss.



(d)
Training set accuracy: 0.9177
Training set average log prob: -0.312122612435
Test set accuracy: 0.8991
Test set average log prob: -0.359125675241
The accuracy and average log-likelihhood is better than the naive bayes's because using logistic regression, we drop the assumption that each pixel is independent.

Problem 4

(a)
θ : 784 * K params
π: K params (K-1 params strictly, since π should sum to 1)
Total: 785K params (If we use K-1, the total would be 785K - 1)

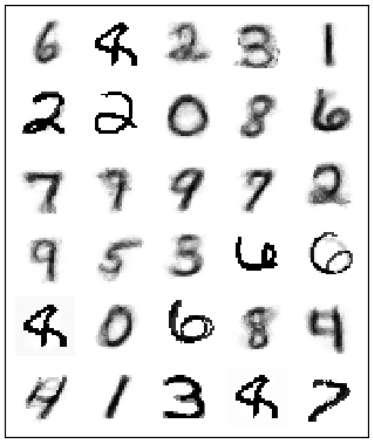
(b)

$$log(p(\mathbf{x}|\theta,\pi)) = log(\sum_{c=1}^{k}\pi_c\prod_{d=1}^{784}\theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}) = F$$

For an arbitray c', d':

$$\begin{aligned}
\frac{\partial F}{\partial \theta_{c'd'}} &= \frac{\partial}{\partial \theta_{c'd'}} log(\sum_{c=1}^{k}\pi_c\prod_{d=1}^{784}\theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}) \\
&= \frac{1}{\sum_{c=1}^{k}\pi_c\prod_{d=1}^{784}\theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}}\frac{\partial}{\partial \theta_{c'd'}}(\sum_{c=1}^{k}\pi_c\prod_{d=1}^{784}\theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}) \\
&= \frac{1}{\sum_{c=1}^{k}\pi_c\prod_{d=1}^{784}\theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}}\frac{\partial}{\partial \theta_{c'd'}}(\pi_{c'}\prod_{d=1}^{784}\theta_{c'd}^{x_d}(1-\theta_{c'd})^{(1-x_d)}) \\
&= \frac{\pi_{c'}\prod_{d=1,d\neq d'}^{784}\theta_{c'd}^{x_d}(1-\theta_{c'd})^{(1-x_d)}}{\sum_{c=1}^{k}\pi_c\prod_{d=1}^{784}\theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}}\frac{\partial}{\partial \theta_{c'd'}}(\theta_{c'd'}^{x_{d'}}(1-\theta_{c'd'})^{(1-x_{d'})}) \\
&= \frac{\pi_{c'}\prod_{d=1,d\neq d'}^{784}\theta_{c'd}^{x_d}(1-\theta_{c'd})^{(1-x_d)}}{\sum_{c=1}^{k}\pi_c\prod_{d=1}^{784}\theta_{cd}^{x_d}(1-\theta_{cd})^{(1-x_d)}}(x_{d'}\theta_{c'd'}^{(x_{d'}-1)}(1-\theta_{c'd'})^{(1-x_{d'})} - (1-x_{d'})(1-\theta_{c'd'})^{-x_{d'}}\theta_{c'd'}^{x_{d'}})
\end{aligned} \tag{8}$$

(c)
Code in starter.py
Compare to the previous ones from the supervised model, the θs here are not as decisive and clear. The previous θs are clearly correspond to a specific digit, while in here, some of the θs seems to the a mix of different digits. Also, since K = 30 in our case, by dividing the training set into 30 clusters, different ways of writing a digit has its own θ map.



(d)
Compare to the previous ones from the supervised model, the the generated bottom part here is actually not bad. Through more digits are paired with the bottom half from another digit, for the ones that are correct, the bottom half is more clear compare the the previous.