CSC311 A3 Writeup

Richard Yan

October 2022

1 Backprop

(a)

$$\begin{tikzcd}
 & & & v \arrow[d] & \\
x \arrow[rrr] \arrow[dr] & & & y \arrow[r] & L \\
w \arrow[r] & z \arrow[r] & h \arrow[ur] & & t \arrow[u] \\
\eta \arrow[r] & s \arrow[ur] & & r \arrow[uu] & \\
U \arrow[ur] & & & &
\end{tikzcd}$$

(b)

$$\begin{aligned}
\frac{d\sigma(x)}{dx} &= -\frac{-e^{-x}}{(1+e^{-x})^2} \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1}{1+e^{-x}}\frac{e^{-x}}{1+e^{-x}} \\
&= \sigma(x)\frac{1+e^{-x}-1}{1+e^{-x}} \\
&= \sigma(x)(1-\frac{1}{1+e^{-x}}) \\
&= \sigma(x)(1-\sigma(x))
\end{aligned}$$

(c) We will compute them step by step:
$\bar{y} = \frac{t}{y} - \frac{1-t}{1-y}$

$\bar{\mathbf{v}} = \bar{y} \cdot y(1-y)\mathbf{h} = \bar{y} \cdot \sigma(\mathbf{v}^T\mathbf{h} + \mathbf{r}^T\mathbf{x})(1 - \sigma(\mathbf{v}^T\mathbf{h} + \mathbf{r}^T\mathbf{x})) \cdot \mathbf{h}$

$\bar{\mathbf{r}} = \bar{y} \cdot y(1-y)\mathbf{x} = \bar{y} \cdot \sigma(\mathbf{v}^T\mathbf{h} + \mathbf{r}^T\mathbf{x})(1 - \sigma(\mathbf{v}^T\mathbf{h} + \mathbf{r}^T\mathbf{x})) \cdot \mathbf{x}$

$\bar{\mathbf{h}} = \bar{y} \cdot y(1-y)\mathbf{v} = \bar{y} \cdot \sigma(\mathbf{v}^T\mathbf{h} + \mathbf{r}^T\mathbf{x})(1 - \sigma(\mathbf{v}^T\mathbf{h} + \mathbf{r}^T\mathbf{x})) \cdot \mathbf{v}$

$\bar{\mathbf{z}} = \bar{\mathbf{h}} \cdot \mathrm{diag}(\mathbf{s})$

$\bar{\mathbf{s}} = \bar{\mathbf{h}} \cdot \mathrm{diag}(\mathbf{z})$

$\bar{\mathbf{x}} = \bar{y} \cdot y(1-y)\mathbf{r} + \bar{\mathbf{z}} \cdot \mathbf{W} = \bar{y} \cdot \sigma(\mathbf{v}^T\mathbf{h} + \mathbf{r}^T\mathbf{x})(1 - \sigma(\mathbf{v}^T\mathbf{h} + \mathbf{r}^T\mathbf{x})) \cdot \mathbf{x} + \bar{\mathbf{z}} \cdot \mathbf{W}$

$\bar{\mathbf{W}} = \bar{z} \cdot \mathbf{x}^T$

$\bar{\eta} = \bar{s} \cdot \mathbf{U}$

$\bar{\mathbf{U}} = \bar{s} \cdot \eta^T$

2 Fitting a Naïve Bayes Model

(a) For θ:

To learn $\hat{\theta}_{jc}$, we should maximize $\sum_{i=1}^{N} \log p(x_j^{(i)} | c^{(i)})$. Which is:

$$\begin{aligned} &\sum_{i=1}^{N}\sum_{k=0}^{9} t_k^{(i)} \log\{\theta_{jk}^{x_j^{(i)}} (1-\theta_{jk})^{(1-x_j^{(i)})}\} \\ =&\sum_{i=1}^{N}\sum_{k=0}^{9} t_k^{(i)} \{x_j^{(i)} \log\theta_{jk} + (1-x_j^{(i)})\log(1-\theta_{jk})\} \end{aligned}$$

Then by calculating the derivative and setting it to 0 gives:

$$\begin{aligned} \frac{d}{d\theta_{jc}} \sum_{i=1}^{N} \log p(x_j^{(i)}|c^{(i)}) &= \sum_{i=1}^{N} t_c^{(i)} (\frac{x_j^{(i)}}{\theta_{jc}} - \frac{1-x_j^{(i)}}{1-\theta_{jc}}) = 0 \\ \implies \hat{\theta}_{jc} &= \frac{\sum_{i=1}^{N} t_c^{(i)} x_j^{(i)}}{\sum_{i=1}^{N} t_c^{(i)}} \end{aligned}$$

In which $\hat{\theta}_{jc}$ counts among all image that is in class c, how many images of those has the pixel $x_j = 1$.

For π:
To learn $\hat{\pi}_c$, we should maximize $\sum_{i=1}^{N} \log p(c^{(i)})$. Which is:

$$\begin{aligned}
&\sum_{i=1}^{N} \log\{\prod_{k=1}^{9} \pi_k^{t_k^{(i)}}\} \\
=&\sum_{i=1}^{N}\sum_{k=1}^{9} t_k^{(i)} \log \pi_k \\
=&\sum_{i=1}^{N}\sum_{k=1}^{8} t_k^{(i)} \log \pi_k + t_9^{(i)} \log(1-\sum_{k=0}^{8} \pi_k)
\end{aligned}$$

Then by calculating the derivative w.r.t. π_c for $c \neq 9$ and setting it to 0 gives:

$$\begin{aligned}
\frac{d}{d\pi_c}\sum_{i=1}^{N} \log p(c^{(i)}) &= \sum_{i=1}^{N} \frac{t_c^{(i)}}{\pi_c} - \sum_{i=1}^{N} \frac{t_9^{(i)}}{1-\sum_{k=0}^{8}\pi_k} = 0 \\
\implies \frac{\hat{\pi}_c}{\hat{\pi}_9} &= \frac{\sum_{i=1}^{N} t_c^{(i)}}{\sum_{i=1}^{N} t_9^{(i)}}
\end{aligned}$$

Then with all those $\frac{\hat{\pi}_c}{\hat{\pi}_9}$, we know that if we add them up all together, that's equivalent to $\frac{1}{\hat{\pi}_9} - 1$, so we can solve for $\hat{\pi}_9$, which is:

$$\hat{\pi}_9 = \frac{\sum_{i=1}^{N} t_9^{(i)}}{N}$$

Thus giving that all of the $\hat{\pi}_c$'s is just counting among all images, how many of those are classified as class c.

(b) Applying Bayes rule, we get:

$$\begin{aligned}
p(\mathbf{t}|\mathbf{x},\boldsymbol{\theta},\boldsymbol{\pi}) &= \frac{p(\mathbf{x}|\mathbf{t},\boldsymbol{\theta},\boldsymbol{\pi})p(\mathbf{t},\boldsymbol{\theta},\boldsymbol{\pi})}{p(\mathbf{x},\boldsymbol{\theta},\boldsymbol{\pi})} \\
&= \frac{p(c|\boldsymbol{\theta},\boldsymbol{\pi})\prod_{j=1}^{784} p(x_j|c,\theta_{jc})}{\sum_{c'=0}^{9} p(c'|\boldsymbol{\pi})p(\mathbf{x}|c',\boldsymbol{\theta},\boldsymbol{\pi})} \\
&= \frac{p(c|\boldsymbol{\pi})\prod_{j=1}^{784} p(x_j|c,\theta_{jc})}{\sum_{c'=0}^{9} p(c'|\boldsymbol{\pi})\prod_{j=1}^{784} p(x_j|c',\theta_{jc'})} \\
&= \frac{\pi_c \cdot \prod_{j=1}^{784} p(x_j|c,\theta_{jc})}{\sum_{c'=0}^{9} \pi_{c'}\prod_{j=1}^{784} p(x_j|c',\theta_{jc'})}
\end{aligned}$$

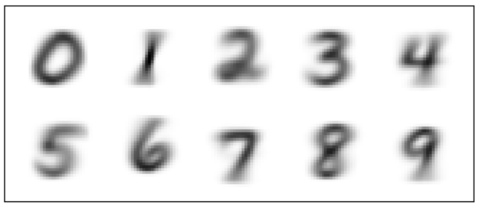
Using results in part (a) will give us:

$$\begin{aligned}
\log p(\mathbf{t}|\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\pi}) &= \log \pi_c + \log\{\prod_{j=1}^{784} p(x_j|c, \theta_{jc})\} - \log\{\sum_{c'=0}^{9} \pi_{c'} \prod_{j=1}^{784} p(x_j|c^{'}, \theta_{jc'})\} \\
&= \log \pi_c + \sum_{j=1}^{784} x_j \log \theta_{jc} + (1 - x_j) \log(1 - \theta_{jc}) \\
&\qquad - \log\{\sum_{c'=0}^{9} \pi_{c'} \prod_{j=1}^{784} p(x_j|c^{'}, \theta_{jc'})\} \\
&= \log \pi_c + \sum_{j=1}^{784} x_j \log \theta_{jc} + (1 - x_j) \log(1 - \theta_{jc}) \\
&\qquad - \log\{\sum_{c'=0}^{9} \exp\{\log \pi_{c'} + \sum_{j=1}^{784} x_j \log \theta_{jc'} + (1 - x_j) \log(1 - \theta_{jc'})\}\}
\end{aligned}$$

Notice that the last term (denominator before log) is written as exp to the log (since $a = e^{\log a}$), which will help us to calculate (under numpy), as sum is much better than product.

(c) All log-likelihood are nan. I believe this is because that the parameter θ we fitted contained a lot of entries of 0, hence when we calculate log(0), it gives nan.

(d) This is the MLE estimator:



(e) We get that for $\alpha = \beta = 3$:

$$
\begin{aligned}
p(\boldsymbol{\theta}|\mathbf{t},\mathbf{x},\boldsymbol{\pi}) &\propto p(\boldsymbol{\theta})p(\mathbf{x},\mathbf{t}|\boldsymbol{\theta},\boldsymbol{\pi}) \\
&\propto [\theta^{\alpha-1}(1-\theta)^{\beta-1}][p(\mathbf{t}|\boldsymbol{\theta},\boldsymbol{\pi})\prod_{i=1}^{N}\prod_{j=1}^{784} p(x_j^{(i)}|c^{(i)},\theta_{jc})] \\
&= [\theta^2(1-\theta)^2][\prod_{c=0}^{9}\pi_c^{t_c}\prod_{i=1}^{N}\prod_{j=1}^{784}\theta_{jc^{(i)}}^{x_j^{(i)}}(1-\theta_{jc^{(i)}})^{1-x_j^{(i)}}] \\
&\propto 2\log\theta + 2\log(1-\theta) + \sum_{c=0}^{9}[t_c\log\pi_c + \sum_{i=1}^{N}\sum_{j=1}^{784} x_j^{(i)}\log\theta_{jc^{(i)}} + (1-x_j^{(i)})\log(1-\theta_{jc^{(i)}})]
\end{aligned}
$$

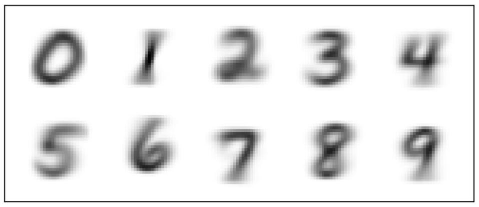
Then calculating the derivative w.r.t. θ and let it equal to 0 will give the estimator, specifically:

$$
\begin{aligned}
&\frac{\partial}{\partial\theta_{jc}} 2\log\theta + 2\log(1-\theta) + \sum_{c=0}^{9}[t_c\log\pi_c + \sum_{i=1}^{N}\sum_{j=1}^{784} x_j^{(i)}\log\theta_{jc^{(i)}} + (1-x_j^{(i)})\log(1-\theta_{jc^{(i)}})] \\
=\ & \frac{2}{\theta_{jc}} - \frac{2}{1-\theta_{jc}} + \sum_{i=1}^{N}\frac{t_c x_j^{(i)}}{\theta_{jc^{(i)}}} - \frac{t_c(1-x_j^{(i)})}{1-\theta_{jc^{(i)}}} = 0 \\
\Longrightarrow\ & \frac{2}{\theta_{jc}} + \sum_{i=1}^{N}\frac{t_c x_j^{(i)}}{\theta_{jc^{(i)}}} = \frac{2}{1-\theta_{jc}} + \sum_{i=1}^{N}\frac{t_c(1-x_j^{(i)})}{1-\theta_{jc^{(i)}}} \\
\Longleftrightarrow\ & (1-\theta_{jc})(2 + \sum_{i=1}^{N} t_c x_j^{(i)}) = \theta_{jc}(2 + \sum_{i=1}^{N} t_c(1-x_j^{(i)})) \\
\Longleftrightarrow\ & \theta_{jc} = \frac{2 + \sum_{i=1}^{N} t_c x_j^{(i)}}{2 + \sum_{i=1}^{N} t_c(1-x_j^{(i)}) + 2 + \sum_{i=1}^{N} t_c x_j^{(i)}} \\
\Longleftrightarrow\ & \theta_{jc} = \frac{2 + \sum_{i=1}^{N} t_c x_j^{(i)}}{4 + \sum_{i=1}^{N} t_c}
\end{aligned}
$$

This count the same thing as shown in part (a), but we add a kind of bias pseudo-count on the numerator (2 here) and a bias pseudo-count on the denominator (4 here), hence we could avoid having 0 in $\boldsymbol{\theta}$ hence to avoid nan calculated afterwards.

(f) Average log-likelihood for MAP is -3.3570631378602847
Training accuracy for MAP is 0.8352166666666667
Test accuracy for MAP is 0.816

(g) This is the MAP estimator:



(h) The assumption is reasonable because it reduce the probability we have to calculate by a large amount hence reducing the time it takes to train the classifier (the general advantage of naive Bayes models), and one thing that the assumption not reasonable to this question is that known the class label doesn't really mean the features (pixels) are independent. To elaborate on that on a simplified context, we are classifying handwritten digits, and pixels aren't going to appear randomly as 0 or 1, but more likely in those shapes of digits (kind of like the pixels that's going to be 1 will be continuous, because of handwritten). Hence the assumption may not be reasonable in this way.

3 Categorical Distribution.

(a) The posterior $p(\boldsymbol{\theta}|\mathcal{D})$ is:

$$\begin{aligned}
&= \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})} \\
&\propto \theta_1^{a_1-1}\theta_2^{a_2-1}\dots\theta_K^{a_K-1}\prod_{k=1}^{K}\theta_k^{x_k} \\
&= \prod_{k=1}^{K}\theta_k^{a_k-1+x_k}
\end{aligned}$$

Which is indeed a conjugate prior of the categorical distribution.

(b) Using Q2 tricks, we can see that since MLE estimator $\theta_k = \frac{N_k}{N}$, then we could guess that MAP estimator $\theta_k = \frac{N_k+a_k-1}{N+\sum_{i=1}^{K}a_i-1} = \frac{N_k+a_k-1}{N-K+\sum_{i=1}^{K}a_i}$. We can verify this since we are maximizing $p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$, hence, maximizing:

$$\begin{aligned}
&\prod_{k=1}^{K} \theta_k^{a_k-1+N_k} \\
&\propto \sum_{k=1}^{K} (a_k - 1 + N_k) \log \theta_k \\
&= \sum_{k=1}^{K-1} (a_k - 1 + N_k) \log \theta_k + (a_K - 1 + N_K) \log(1 - \sum_{k=1}^{K-1} \theta_k)
\end{aligned}$$

Then calculating derivative for $k \neq K$ and setting to 0 will give:

$$\begin{aligned}
&\frac{\partial}{\partial \theta_k} \sum_{j=1}^{K-1} (a_j - 1 + N_j) \log \theta_j + (a_K - 1 + N_K) \log(1 - \sum_{j=1}^{K-1} \theta_j) \\
&= \frac{(a_k - 1 + N_k)}{\hat{\theta}_k} - \frac{(a_K - 1 + N_K)}{1 - \sum_{j=1}^{K-1} \hat{\theta}_j} = 0 \\
&\implies \frac{(a_k - 1 + N_k)}{\hat{\theta}_k} = \frac{(a_K - 1 + N_K)}{\hat{\theta}_K} \\
&\implies \frac{\hat{\theta}_k}{\hat{\theta}_K} = \frac{a_k - 1 + N_k}{a_K - 1 + N_K}
\end{aligned}$$

Solving for $\hat{\theta}_K$ and generalize gives:

$$\hat{\theta}_k = \frac{a_k - 1 + N_k}{\sum_{j=1}^{K} (a_j + N_j) - K} = \frac{a_k - 1 + N_k}{N - K + \sum_{j=1}^{K} a_j}$$

for all k from 1 to K.

(c)

$$\begin{aligned}
p(x^{(N+1)} < K) &= \sum_{k=1}^{K-1} p(x^{(N+1)} = k) \\
&= \sum_{k=1}^{K-1} \int p(x^{(N+1)} = k|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta} \\
&= \sum_{k=1}^{K-1} \int \theta_k p(\theta_k|\mathcal{D}) d\theta_k \\
&= \sum_{k=1}^{K-1} \mathbb{E}[\theta_k] \\
&= \sum_{k=1}^{K-1} \frac{a_k}{\sum_{k'} a_{k'}} \\
&= \frac{\sum_{k=1}^{K-1} a_k}{\sum_{k'} a_{k'}}
\end{aligned}$$

4 Gaussian Discriminant Analysis.

(a) Average conditional log likelihood for training: -0.12462443666863006
Average conditional log likelihood for test: -0.1966732032552555

(b) Accuracy on training: 0.9814285714285714
Accuracy on test: 0.97275

(c) We will get that:
Average conditional log likelihood for training with diagonal covariance: -1.2307654222727913
Average conditional log likelihood for test with diagonal covariance: -1.2872603667558402
Accuracy on training with diagonal covariance: 0.85
Accuracy on test with diagonal covariance: 0.84

We can see that both likelihood and prediction accuracy (on both training and test) became worse.
This is because that the diagonal entries of the covariance matrix only measure the variance of the corresponding pixel i (i.e. $\text{Cov}[\text{digit}]_{ii} = \text{Var}(x_i^{\text{digit}})$, hence using it to calculate likelihood and prediction can't take the relationship between pixels into account, as those relationship between different pixels are measured and recorded in the non-diagonal entries of the covariance matrices.