Machine Learning 1 - Homework Week 3

Minh Ngo 10897402[1]

September 28, 2015

[1] University of Amsterdam
minh.ngole@student.uva.nl

Task 1

1. Likelihood for the general two class naive Bayes classifier is following (Bishop, 4.89):

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} p(C_1|\mathbf{x}, \boldsymbol{\theta})^{t_n} p(C_2|\mathbf{x}, \boldsymbol{\theta})^{1-t_n} \\ where\ t_n \in \{0, 1\}, \mathbf{t} = \{t_n\}_{n=1}^{N} \tag{1}$$

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} \Big(p(C_1)p(\mathbf{x}|C_1)\Big)^{t_n} \Big(p(C_2)p(\mathbf{x}|C_2)\Big)^{1-t_n} \tag{2}$$

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} \Big(p(C_1) \prod_{d=1}^{D} p(x_{nd}|C_1, \theta_{dk})\Big)^{t_n} \Big(p(C_2) \prod_{d=1}^{D} p(x_{nd}|C_2, \theta_{dk})\Big)^{1-t_n} \tag{3}$$

2. For the Poisson model it likelihood is:

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} \Big(\pi_1 \prod_{d=1}^{D} \frac{\lambda_{d1}^{x_{nd}}}{x_{nd}!} exp(-\lambda_{d1})\Big)^{t_n} \Big((1-\pi_1) \prod_{d=1}^{D} \frac{\lambda_{d2}^{x_{nd}}}{x_{nd}!} exp(-\lambda_{d2})\Big)^{1-t_n} \tag{4}$$

3. Log-likelihood is:

$$
\begin{aligned}
\log p(\mathbf{t}|\mathbf{w}) = \sum_{n=1}^{N} \Bigg(\log \pi_1 + \sum_{d=1}^{D} (x_{nd} \log \lambda_{d1} - \log(x_{nd}!) - \lambda_{d1}) \Bigg)^{t_n} \\
+ \sum_{n=1}^{N} \Bigg(\log(1-\pi_1) + \sum_{d=1}^{D} (x_{nd} \log \lambda_{d2} - \log(x_{nd}!) - \lambda_{d2}) \Bigg)^{1-t_n}
\end{aligned}
\tag{5}
$$

4. Now we solve MLE estimators for λ_{dk}. Let $t_{1n} = t_n$, $t_{2n} = 1 - t_n$

$$
\frac{\partial p(\mathbf{t}|\mathbf{w})}{\partial \lambda_{dk}} = \sum_{n=1}^{N} (\frac{x_{nd}}{\lambda_{dk}} - 1)^{t_{kn}} = 0 \tag{6}
$$

$$
\Leftrightarrow \sum_{n=1}^{N} (\frac{x_{nd}}{\lambda_{dk}})^{t_{kn}} = \sum_{n=1}^{N} 1^{t_{kn}} \Leftrightarrow \lambda_{dk} = \frac{1}{\sum_{n=1}^{N} 1^{t_{kn}}} \sum_{n=1}^{N} (x_{nd})^{t_{kn}} = \frac{1}{N_k} \sum_{n=1}^{N} x_{nd}^{t_{kn}} \tag{7}
$$

5. $p(C_1|\mathbf{x})$ for the general two class naive Bayes classifier:

$$
\begin{aligned}
p(C_1|\mathbf{x}) &= \frac{p(C_1)p(\mathbf{x}|C_1)}{p(C_1)p(\mathbf{x}|C_1) + p(C_2)p(\mathbf{x}|C_2)} \\
&= \frac{1}{1 + \frac{p(C_2)p(\mathbf{x}|C_2)}{p(C_1)p(\mathbf{x}|C_1)}}
\end{aligned}
\tag{8}
$$

6. For the Poisson model:

$$
p(C_1|\mathbf{x}) = \frac{1}{1 + \frac{(1-\pi_1) \prod_{d=1}^{D} \lambda_{d2}^{x_{nd}} exp(-\lambda_{d2})}{\pi_1 \prod_{d=1}^{D} \lambda_{d1}^{x_{nd}} exp(-\lambda_{d1})}} \tag{9}
$$

7. If we rewrite $p(C_1|\mathbf{x})$ as a sigmoid $\sigma(a) = \dfrac{1}{1+exp(-a)}$ then:

$$exp(-a) = \frac{(1-\pi_1)\prod_{d=1}^{D}\lambda_{d2}^{x_{nd}}exp(-\lambda_{d2})}{\pi_1\prod_{d=1}^{D}\lambda_{d1}^{x_{nd}}exp(-\lambda_{d1})} \tag{10}$$

$$\begin{aligned} a &= -\log\left(\frac{(1-\pi_1)\prod_{d=1}^{D}\lambda_{d2}^{x_{nd}}exp(-\lambda_{d2})}{\pi_1\prod_{d=1}^{D}\lambda_{d1}^{x_{nd}}exp(-\lambda_{d1})}\right) \\ &= \log(\frac{\pi_1}{1-\pi_1}) + \sum_{d=1}^{D}(x_{nd}\log\frac{\lambda_{d1}}{\lambda_{d2}} + (\lambda_{d2}-\lambda_{d1})) \end{aligned} \tag{11}$$

8. Assume $a = \mathbf{w}^T\mathbf{x} + w_0$ we can solve for $\mathbf{w}$ and w_0:

$$w_0 = \log\frac{\pi_1}{1-\pi_1} + \sum_{d=1}^{D}(\lambda_{d2}-\lambda_{d1}) \tag{12}$$

$$\mathbf{w} = \{\log\frac{\lambda_{d1}}{\lambda_{d2}}\}_{d=1}^{D} \tag{13}$$

9. The decision boundary is a linear function of x because we have linear dependency between $\mathbf{x}$ and a with a constant offset w_0 as the formula (12) and (13) show.

Task 2

$$K > 2 \quad y_k = p(C_k|\boldsymbol{\phi}) = \frac{exp(a_k)}{\sum_i exp(a_i)} \quad a_k = -\mathbf{w}_k^T\boldsymbol{\phi} \tag{14}$$

1. For computing the derivative $\dfrac{\partial y_k}{\partial \mathbf{w}_j}$ we use this formula:

$$\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2} \tag{15}$$

$$\begin{aligned}
\Rightarrow \frac{\partial y_k}{\partial \mathbf{w}_j} &= \frac{exp(a_k)\dfrac{\partial ak}{\partial \mathbf{w}_j}(\sum\limits_i exp(a_i)) - exp(a_k)exp(a_j)\dfrac{\partial a_j}{\partial \mathbf{w}_j}}{(\sum\limits_i exp(a_i))^2} \\
&= \frac{exp(a_k)\dfrac{\partial a_k}{\partial \mathbf{w}_j}}{\sum\limits_i exp(a_i)} - \frac{exp(a_k)exp(a_j)\dfrac{\partial a_j}{\partial \mathbf{w}_j}}{(\sum\limits_i exp(a_i))^2} \\
&= y_k(\boldsymbol{\phi})\boldsymbol{\phi}^{[k=j]} - y_k(\boldsymbol{\phi})y_j(\boldsymbol{\phi})\boldsymbol{\phi} \\
&= y_k(\boldsymbol{\phi})(\mathbf{I}_{kj} - y_j(\boldsymbol{\phi}))\boldsymbol{\phi}
\end{aligned} \tag{16}$$

2. Likelihood will be:

$$p(\mathbf{T}|\mathbf{\Phi}, \mathbf{W}) = \prod_{n=1}^{N}\prod_{k=1}^{K} y_k(\boldsymbol{\phi}_n)^{t_{nk}} \tag{17}$$

   And log likelihood:

$$\log p(\mathbf{T}|\mathbf{\Phi}, \mathbf{W}) = \sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\log y_k(\boldsymbol{\phi}_n) \tag{18}$$

3.

$$\begin{aligned}
\frac{\partial \log p(\mathbf{T}|\mathbf{\Phi}, \mathbf{W})}{\partial \mathbf{w}_j} &= \sum_{n=1}^{N}\sum_{k=1}^{K} \frac{t_{nk}}{y_k(\boldsymbol{\phi}_n)}\frac{\partial y_k}{\partial \mathbf{w}_j} \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} \frac{t_{nk}}{y_k(\boldsymbol{\phi}_n)} y_k(\boldsymbol{\phi}_n)(I_{kj} - y_j(\boldsymbol{\phi}_n))\boldsymbol{\phi}_n \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}(\mathbf{I}_{kj} - y_j(\boldsymbol{\phi}_n))\boldsymbol{\phi}_n \\
&= \sum_{n=1}^{N}(t_{nj} - y_j(\boldsymbol{\phi}_n))\boldsymbol{\phi}_n
\end{aligned} \tag{19}$$

4. We are minimizing a cross entropy error:

$$\begin{aligned} E(\mathbf{W}) &= -\log p(\mathbf{T}|\boldsymbol{\Phi}, \mathbf{W}) \\ &= -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \log y_k(\boldsymbol{\phi}_n) \end{aligned} \tag{20}$$

$$\frac{\partial E(\mathbf{W})}{\partial \mathbf{w}_j} = \sum_{n=1}^{N}(y_j(\boldsymbol{\phi}_n) - t_{nj})\boldsymbol{\phi}_n \tag{21}$$

5. SGD algorithm for Logistic Regression:
   (a) Initialize weights $\mathbf{W}$, learning rate γ
   (b) do
      i. Randomly choose $n \sim U(1, N)$.
      ii.

$$\begin{aligned} \mathbf{w}_j^{(t+1)} &= \mathbf{w}_j^{(t)} - \gamma^{(t)} \nabla \mathbf{e}_n \\ &= \mathbf{w}_j^{(t)} - \gamma^{(t)} (y_j(\boldsymbol{\phi}_n) - t_{nj})\boldsymbol{\phi}_n \end{aligned} \tag{22}$$

   until convergence

6. It's a stochastic optimization procedure because instead of the full gradient we are picking random point from the uniform distribution and compute gradient from it.

7.

$$\begin{aligned} E(\mathbf{W}) &= -\log p(\mathbf{T}|\boldsymbol{\Phi}, \mathbf{W}) + \frac{\lambda}{2}\sum_{j=1}^{K} \mathbf{w}_j^T \mathbf{w}_j \\ &= -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \log y_k(\boldsymbol{\phi}_n) + \frac{\lambda}{2}\sum_{j=1}^{K} \mathbf{w}_j^T \mathbf{w}_j \end{aligned} \tag{23}$$

   This would be the MAP estimator.