Aaron Dardik

CS567 Problem Set #2

1.1) Given $q_i: 1 \le i \le K, C = \frac{1}{\sum q_i}$ we will denote Cq_i by p_i. We will let $[y = k] \stackrel{\text{def}}{=} 1\{y = k\}$. Now we must show that $p = \prod_{i=1}^{K} (p_i)^{[y=i]}$ is in the exponential family. Now, $p = e^{\log p} = e^{\log \prod_{i=1}^{K} p_i^{[y=i]}}$. By using the fact that the logarithm of a product is the sum of the logarithms the above expression simplifies to $e^{\sum_i \log\left(p_i^{[y=i]}\right)} = e^{\sum_i [y=i] \log p_i}$. Let $\eta = \begin{bmatrix} \log p_1 \\ \vdots \\ \log p_{K-1} \end{bmatrix}, t(y) = \begin{bmatrix} [y = 1] \\ \vdots \\ [y = K-1] \end{bmatrix}, b(y) = 1, a(\eta(C)) = -\log(C-1)$(written here using the variable q, C for convenience instead of p. This shows that the categorical distribution is in the exponential family.

1.2) $p(y|x) \sim$ Bernoulli $\Rightarrow p(y|x) = p^y(1-p)^{1-y} = e^{\log p^y(1-p)^{1-y}} = e^{y \log p + (1-y)\log(1-p)}$. This simplifies to $e^{y \log \frac{p}{1-p} + \log(1-p)}$. Now, $\eta = Wx = \log \frac{p}{1-p}, t(y) = y, b(y) = 1, a(\eta) = -\log(1-p)$. Now, $h(x; W) = E_y t(y) = 1 * e^{1 \log \frac{p}{1-p} + \log(1-p)} + 0 = p = p(x; W), \Rightarrow\ e^{Wx} = \frac{p}{1-p}$ $\Rightarrow e^{-Wx} = \frac{1-p}{p} = \frac{1}{p} - 1 \Rightarrow p(x; W) = \frac{1}{1+e^{-Wx}}$.

1.3) From part 1.1 we see that for the categorical distribution we have $\eta = \begin{bmatrix} \log p_1 \\ \vdots \\ \log p_K \end{bmatrix}, t(y) = \begin{bmatrix} [y = 1] \\ \vdots \\ [y = K] \end{bmatrix}, b(y) = 1, a(\eta) = 0$. From this we see that $p_i = e^{\eta_i}, \sum p_i = \sum e^{\eta_i} = 1$. By substituting back in and dividing we see that $p_i = \frac{e^{\eta_i}}{\sum_j e^{\eta_j}}$. Now, as $\eta = Wx \Rightarrow p(y = i|x) = e^{(Wx)_i} \big/ \sum_j e^{(Wx)_j} = e^{w_i^T x} \big/ \sum_j e^{w_j^T x}$. Now, we note that p_K can be expressed as a linear combination of $p_1 \dots p_{K-1}$ so we have that $E[t(y)|x] = E\left[\begin{matrix} [y = 1] \\ \vdots \\ [y = K-1] \end{matrix} \middle| x\right] = \begin{bmatrix} e^{w_1^T x} \big/ \sum e^{w_j^T x} \\ \vdots \\ e^{w_{K-1}^T x} \big/ \sum e^{w_j^T x} \end{bmatrix} = \begin{bmatrix} p_1 \\ \vdots \\ p_{K-1} \end{bmatrix}$ with p_K being a parameter expressed in terms of the first $p_1, \dots, p_{K-1}$ and not an independent variable itself.

1.4) $p(y; \lambda) = \frac{\lambda^y e^{-\lambda}}{y!} = e^{\log \frac{\lambda^y e^{-\lambda}}{y!}} = e^{\log \lambda^y + \log e^{-\lambda} - \log y!} = \frac{1}{y!} * e^{y \log \lambda - \lambda}$. From this we conclude that the Poisson distribution is in the exponential family with $b(y) = \frac{1}{y!}, \eta = \log \lambda,\ t(y) = y, a(\eta) = \lambda$. Now, $\eta = w^T x = \log \lambda \Rightarrow e^{w^T x} = \lambda$. As the expected value of a Poisson distribution is λ we have $\hat{y_i} = E[y|x; w] = \lambda = e^{w^T x}$.

2.1.1) For ease of notation we will denote our cost function by C instead of l. To find $\frac{\partial C}{\partial \boldsymbol{a}}$ it suffices to find $\frac{\partial C}{\partial a_i}, \forall i$. We will consider two cases: $i \ne y, i = y$. In the first case, we have $\frac{\partial C}{\partial a_i} = \frac{-1}{z_y} * \frac{\partial}{\partial a_i}(z_y)$. The latter derivative we will solve via the quotient rule to get $0 - e^{a_y}e^{a_i} / (\sum e^{a_j})^2 = -z_y * z_i$. Combining these results we have $i \neq y \Rightarrow \frac{\partial C}{\partial a_i} = z_i$. In the latter case where $i = y$, we have $\frac{\partial C}{\partial a_y} = (\sum e^{a_j}) * e^{a_y} - e^{a_y}e^{a_y} / (\sum e^{a_j})^2 = z_y - (z_y)^2$. Thus, letting $\boldsymbol{y} \in \mathbb{R}^K$ be the vector whose kth coordinate is 1 if $k = y$ and 0 otherwise, we have that $\frac{\partial C}{\partial \boldsymbol{a}} = \boldsymbol{z} - z_y^2\boldsymbol{y}$.

2.1.2) Now, $\frac{\partial C}{\partial \boldsymbol{W}^{(2)}} = \frac{\partial C}{\partial \boldsymbol{a}}\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{W}^{(2)}} = \frac{\partial C}{\partial \boldsymbol{a}}\boldsymbol{h}$. Additionally, $\frac{\partial C}{\partial \boldsymbol{b}^{(2)}} = \frac{\partial C}{\partial \boldsymbol{a}}\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{b}^{(2)}} = \frac{\partial C}{\partial \boldsymbol{a}}$.

2.1.3) $\frac{\partial C}{\partial \boldsymbol{u}} = \frac{\partial C}{\partial \boldsymbol{a}}\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{h}}\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{u}} = \frac{\partial C}{\partial \boldsymbol{a}}\boldsymbol{W}^{(2)}\boldsymbol{H}(\boldsymbol{u})$ where $\boldsymbol{H}(\boldsymbol{u}) = \begin{bmatrix} H(u_1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & H(u_M) \end{bmatrix}$, $H(u_i) = \begin{cases} 1 \Leftrightarrow u > 0 \\ 0 \Leftrightarrow u \leq 0 \end{cases}$.

2.1.4) $\frac{\partial C}{\partial \boldsymbol{W}^{(1)}} = \frac{\partial C}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{W}^{(1)}} = \frac{\partial C}{\partial \boldsymbol{u}}\boldsymbol{x}$. Additionally, we have $\frac{\partial C}{\partial \boldsymbol{b}^{(1)}} = \frac{\partial C}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{b}^{(1)}} = \frac{\partial C}{\partial \boldsymbol{u}}$.

2.2) $\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)}, \boldsymbol{b}^{(1)}$ being initialized to zero implies that $\boldsymbol{u} = 0, \boldsymbol{h} = 0, \boldsymbol{a} = \boldsymbol{b}^{(2)}$ and $\boldsymbol{z}$ is constant. Now, plugging these values in we have $\boldsymbol{h} = 0 \Rightarrow \frac{\partial C}{\partial \boldsymbol{W}^{(2)}} = 0, \boldsymbol{W}^{(2)} = 0 \Rightarrow \frac{\partial C}{\partial \boldsymbol{u}} = 0 \Rightarrow \frac{\partial C}{\partial \boldsymbol{W}^{(1)}} = 0, \frac{\partial C}{\partial \boldsymbol{b}^{(1)}} = 0$. Now, learning in the hidden layer would adjust $\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}$ along the gradient of cost with respect to those variables, but as the partial derivatives along those variables are zero, the gradient is zero and no learning occurs.

2.3) Removing the nonlinear operation $\boldsymbol{h}$ we have $\boldsymbol{x} \in \mathbb{R}^D, \boldsymbol{u} = \boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)} \in \mathbb{R}^M$. Letting $\boldsymbol{a} = \boldsymbol{W}^{(2)}\boldsymbol{u} + \boldsymbol{b}^{(2)}, \boldsymbol{W}^{(2)} \in \mathbb{R}^{KxM}, \boldsymbol{b}^{(2)} \in \mathbb{R}^K \Rightarrow \boldsymbol{a} = \boldsymbol{W}^{(2)}(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}$. Let $\mathcal{U} = \boldsymbol{W}^{(2)}\boldsymbol{W}^{(1)} \in \mathbb{R}^{KxD}$ and $\mathcal{v} = \boldsymbol{W}^{(2)}\boldsymbol{b}^{(1)} + \boldsymbol{b}^{(2)} \in \mathbb{R}^K \Rightarrow \boldsymbol{a} = \mathcal{U}\boldsymbol{x} + \mathcal{v}$and is therefore linear in $\boldsymbol{x}$.

3.1) $L(w) = \frac{1}{2}\sum_{i=1}^{n}\|w^T\phi(x_i) - y_i\|_2^2 + \frac{1}{2}\lambda\|w\|_2^2, \lambda > 0$. Taking derivatives and setting equal to zero, we have $L'(w) = \sum_{i=1}^{n}\phi^T(x_i)(w^T\phi(x_i) - y_i) + \lambda w = 0$. Now, $w_{t+1} \leftarrow w_t + \alpha\nabla_w L$. We now plug in, to get the update rule $w_{t+1} \leftarrow w_t + \alpha(\sum_{i=1}^{n}\phi^T(x_i)(w^T\phi(x_i) - y_i) + \lambda w)$. From this we can simplify to the update rule $w_{t+1} \leftarrow w_t(1 + \alpha\lambda) + \alpha\sum\phi^T(x_i)(w^T\phi(x_i) - y_i)$.

3.2) If we attempt to do gradient descent on regularized linear regression without kernel, then we have to recalculate $\phi(x_i)$ for all x_i each time.

3.3.1) Note $w_{t+1} \leftarrow w_t(1 + \alpha\lambda) + \alpha\sum\phi^T(x_i)(w^T\phi(x_i) - y_i)$. If $w_0 = 0$ we see that $w_1 = -\alpha\sum\phi^T(x_i) * (y_i) = \sum\beta_i\phi(x_i)$ is a linear combination of $\phi(x_i)$. Assume then that the inductive hypothesis holds for time step up to w_t. Now, as $w_{t+1} \leftarrow w_t(1 + \alpha\lambda) + \alpha\sum\phi^T(x_i)(w^T\phi(x_i) - y_i)$ we can substitute back into the expression to get $w_{t+1} = (1 + \alpha\lambda)\sum\beta_i\phi(x_i) + \alpha\sum\phi^T(x_i)(w^T\phi(x_i) - y_i)$ $= \sum[(1 + \alpha\lambda)\beta_i + \alpha(w^T\phi(x_i) - y_i)]\phi(x_i)$ which is a linear combination of $\phi(x_i)$ and so the result holds.

3.3.2) Substituting for w we get $B_i^{t+1} = \beta_i^t - \alpha y_i + \alpha\sum_j\beta_j^t K(x_i, x_j)$.

4.1) $w^* = \arg\max_{w\in\mathbb{R}^D}\sum_i y_i w^T x_i - \lambda(w^T w - 1)$. Taking the derivative of the expression “inside” the arg max we have $\sum_i y_i x_i - 2\lambda w = 0 \Rightarrow w^* = \frac{1}{2\lambda}\sum_i y_i x_i = \frac{1}{2\lambda}(\sum_{i:x_i\in C_1} x_i - \sum_{j:x_j\in C_{-1}} x_j) = w^*$.

4.2) $\|w\| = 1 = \frac{1}{4\lambda^2}\left(\sum_{i:x_i\in C_1} x_i^2 + \sum_{j:x_j\in C_{-1}} x_j^2 - 2(\sum_{i:x_i\in C_1} x_i)(\sum_{j:x_j\in C_{-1}} x_j)\right)$. By multiplying both sides of the equation by $4\lambda^2$ and simplifying notation, we have $4\lambda^2 = \sum_{C_1} x_i^2 + \sum_{C_{-1}} x_j^2 - 2\left(\sum_{C_1} x_i\right)\left(\sum_{C_{-1}} x_j\right)$. Dividing by 4 and remembering that we know the square root of the right expression (as we calculated it by squaring another expression) we can conclude the following result: $\lambda = \frac{1}{2}(\sum_{i:x_i\in C_1} x_i - \sum_{j:x_j\in C_{-1}} x_j)$.

4.3) We cannot always solve a problem in terms of its dual formulation, if strong duality does not hold then the w^*that minimizes training error may not be the same as that which maximizes $f(w)$.