Homework 1: Solutions
600.482/682 Deep Learning
Fall 2018

Kaushik Srinivasan

September 28, 2018

With collaboration with Nathan Vallapureddy

1. (a)

$$
\begin{aligned}
P(y|x) = \mathcal{N}(ax, s^2) &\Rightarrow \prod_i p(y_1|x) \longrightarrow \max \\
&= \log\left(\prod_i^N \frac{1}{\sqrt{2\pi s^2}} e^{\frac{-(y_i - ax)^2}{2x^2}}\right) \\
&= \min \frac{1}{\sqrt{2\pi s^2}} \sum_i \frac{(y_i - ax)^2}{\underbrace{2s^2}_{\text{constant}}} \\
&= \boxed{\min \sum_i^N (y_i - ax)^2} \Longrightarrow x = \frac{\sum_{i=1}^N y_i}{aN}
\end{aligned}
$$

(b) Maximum a posteriori : $\max P(x|y) = \frac{P(y|x)P(x)}{P(y)}$

$$P(Y) = \sum_j P(y|x_j) \cdot p(x_j) \leftarrow \text{this is a normalizing constant, so we can ignore}$$

This means we can instead compute

$$
\begin{aligned}
\operatorname{argmax} \prod p(y_i|x) \cdot p(x) &= \operatorname{argmax}\left(\prod_i^N \frac{1}{\sqrt{2\pi s^2}} e^{\frac{-(y_i - ax)^2}{s^2}} \cdot \frac{1}{\sqrt{2\pi r^2}} e^{\frac{-x^2}{r^2}}\right) \\
&= \min \sum_i \frac{-(y_i - ax)^2}{2s^2} + \frac{-x^2}{2r^2} \\
&= \min\left(\frac{x^2}{r^2} + \frac{1}{s^2}\sum_{i=1}^N (y_i - ax)^2\right)
\end{aligned}
$$

Differentiate w.r.t. x and set to 0

$$
\begin{aligned}
\frac{2x}{r^2} + \frac{1}{s^2}\sum_{i=1}^N -2a(y_i - ax) = 0 \\
\rightarrow \frac{x}{r^2} = \frac{a}{s^2}\sum_{i=1}^N (y_i - ax) \\
\rightarrow \frac{x}{r^2} + \frac{a^2 N x}{s^2} = \frac{a}{s^2}\sum_{i=1}^N y_i \\
\boxed{x = \frac{\frac{a}{s^2}\sum_{i=1}^N y_i}{\frac{1}{r^2} + \frac{a^2 N}{s^2}}}
\end{aligned}
$$

(c) When $a = 1, s = 1$, and $r = 1$ and $y \in \{0.45, 0.13, -0.26, 1.27, -0.87, -0.49, -0.12, 0.23\}$. Then maximum likelihood estimation is.

$$\textbf{MLE:} \qquad \sum_{i=1}^{N} y_i = 0.34 \quad \text{hence} \quad x = \frac{0.34}{8} = \boxed{0.0425}$$

$$\textbf{MAP:} \qquad \sum_{i=1}^{N} y_i = 0.34 \quad \text{hence} \quad x = \frac{0.34}{1+8} \approx \boxed{0.0378}$$

2. (a)

$$f(x;\theta)y = y\sum_j \theta_j x^{(j)}$$

$$\boxed{\frac{\partial}{\partial\theta_j} f(x;\theta)y = x^{(j)}y}$$

(b) let M be the set with all misclassified samples, then

$$\boxed{\sum_{i \in M} x_i^{(j)} y}$$

(c) Code in file

(d) Code in file

(e) The Answers are

- Data 1 - converges in 5 epochs - [[0.00312885], [0.02170075], [-0.025]]
- Data 2 - doesn't converge, - min error rate after 29 epochs (0.25) - [[0.01895395],[0.00178105],[-0.02]]
- Data 3 - doesn't converge - min error rate function after 66 epochs (0.15) - [[0.00988327], [-0.01209465], [-0.00631251], [0.01]]
- Data 4 - converges in 7 epochs - [[0.00564839],[0.00030848], [0.02565065], [-0.03]]
- Data 5 - converges in 6 epochs - [[0.03937346],[-0.00899502], [0.00273425], [-0.03372524], [0.01]]
- Data 6 - doesn't converge - min error rate after 11 epochs (0.15) - [[0.00790035], [0.01804779], [-0.02766167], [-0.02559074], [0.01]]

3. (a)

$$P(y|x;\theta) = \sigma(\theta^T x)$$

$$O(D;\theta) = \prod_i P(y_i|x_i) = \prod_i z_i^{y_i}(1-z_i)^{(1-y_i)} \qquad (z_i = \sigma(\theta^T x))$$

$$\boxed{-\log(O(D)) = -\sum_i \left[y_i \log(z_i) + (1-y_i)\log(1-z_i)\right]} \quad \leftarrow \text{cross entropy}$$

(b) For simplicity, assume for one example (x, y) where i is fixed, then we can generalize to all the examples. I will also include the negative of the sign in the final step

$$\begin{aligned}
\log(O(D)) &= y\log(\sigma(\theta^T x)) + (1-y)\log(1-\sigma(\theta^T x)) \\
\frac{\partial}{\partial\theta_j}\log(\theta) &= \left(y\frac{1}{\sigma(\theta^T x)} - (1-y)\frac{1}{1-\sigma(\theta^T x)}\right)\cdot\frac{\partial}{\partial\theta_j}\sigma(\theta^T x) \\
&= \left(\frac{y(1-\sigma(\theta^T x)) - (1-y)\sigma(\theta^T x)}{\sigma(\theta^T x)(1-\sigma(\theta^T x))}\right)\cdot\sigma(\theta^T x)\cdot(1-\sigma(\theta^T x))\cdot x^{(j)} \\
&= \left[y(1-\sigma(\theta^T x)) - (1-y)\sigma(\theta^T x)\right]x^{(j)} \\
&= [y - y\sigma(\theta^T x) - \sigma(\theta^T x) + y\sigma(\theta^T x)]x^{(j)} \\
&= [y - \sigma(\theta^T x)]x^{(j)}
\end{aligned}$$

Now if we include all the examples and the negation

$$\boxed{\frac{\partial}{\partial\theta_j} - \log(\theta) = \sum_i [\sigma(\theta^T x_i) - y_i] x_i^{(j)}}$$

(c) Please see code in attached file

(d) The results from the data are:

- Data 1 - converges in 51 epochs - [[0.49006907], [3.83194682], [-4.26828716]]
- Data 2 - doesn't converge, - min error rate after 28 epochs (0.25) - [[1.39005559], [-0.00601565], [-1.33555841]]
- Data 3 - doesn't converge - min error rate function after 3 epochs (0.25) - [[0.43290483], [-0.44922362], [-0.04366681], [-0.02600216]]
- Data 4 - converges in 76 epochs - [[1.32904062], [-0.43986865], [4.88545346], [-4.97437984]]
- Data 5 - converges in 9 epochs - [[2.00545512], [-0.23228308], [0.13208157], [-1.50019973], [0.23684892]]
- Data 6 - doesn't converge - min error rate after 1 epochs (0.25) - [[0.12201986], [0.11603158], [-0.10142059], [-0.43052066], [0.02348147]]

4. (a) Cost function for n examples,

$$\boxed{J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta^T x_i - y_i)^2}$$

(b) the Gradient Descent Rule for this cost function is

$$\frac{\partial}{\partial\theta_j} J(\theta) = \frac{2}{n}\sum_{i=1}^{n}(\theta^T x_i - y_i) \cdot x_i^{(j)}$$

And so the update rule for a given θ_j is

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{n}(\theta^T x_i - y_i) \cdot x_i^{(j)}$$

(c) Code is in attached file

(d) The values of theta (θ): [[3.18654211], [0.79760108]] where the slope = 3.18654211 and intercept = 0.79760108