Approximating Class Posteriors With a Model Function Whose Parameters Are Estimated Using Maximum Likelihood Estimation Method

Given dataset $D = \{(x_1, l_1), \ldots, (x_N, l_N)\}$ and model $h_i(x;\theta)$ for $P(l(x)=i|x)$, use MLE to optimize (train) θ.

$$\hat{\theta}_{ML} = \arg\max_{\theta} p(D|\theta) = \arg\max_{\theta} \ln p(D|\theta)$$

$$= \arg\max_{\theta} \ln \left[p((x_1, l_1), \ldots, (x_N, l_N) | \theta) \right]$$

Assume iid samples

$$= \arg\max_{\theta} \ln \prod_{n=1}^{N} p(x_n, l_n | \theta)$$

$$= \arg\max_{\theta} \sum_{n=1}^{N} \ln p(x_n, l_n | \theta)$$

Bayes Rule

$$= \arg\max_{\theta} \sum_{n=1}^{N} \left(\ln p(l_n | x_n \theta) + \ln p(x_n | \theta) \right)$$

$x_n \perp\!\!\!\perp \theta$

$$= \arg\max_{\theta} \sum_{n=1}^{N} \ln p(l_n | x_n \theta) + \underbrace{\cancel{\sum_{n=1}^{N} \ln p(x_n)}}_{\text{constant}}$$

Now introduce the model ~~[illegible]~~ $p(l_n|x_n\theta) = h_{l_n}(x_n,\theta)$

$$\hat{\theta}_{ML} = \underset{\theta}{\operatorname{argmax}} \sum_{n=1}^{N} \ln h_{l_n}(x_n,\theta)$$

Our model $h: \mathbb{R}^d \to \mathbb{R}^c$ where $x \in \mathbb{R}^d$ and $l \in \{1, .., c\}$ is a multi-input multi-output function

$$\underset{\in \mathbb{R}^d}{x} \longrightarrow \boxed{h(x,\theta)} \longrightarrow \begin{cases} h_1(x,\theta) \approx P(l=1|x) \\ \vdots \\ h_c(x,\theta) \approx P(l=c|x) \end{cases}$$

$$\hat{\theta}_{ML} = \underset{\theta}{\operatorname{argmin}} -\frac{1}{N} \sum_{n=1}^{N} \ln h_{l_n}(x_n,\theta)$$

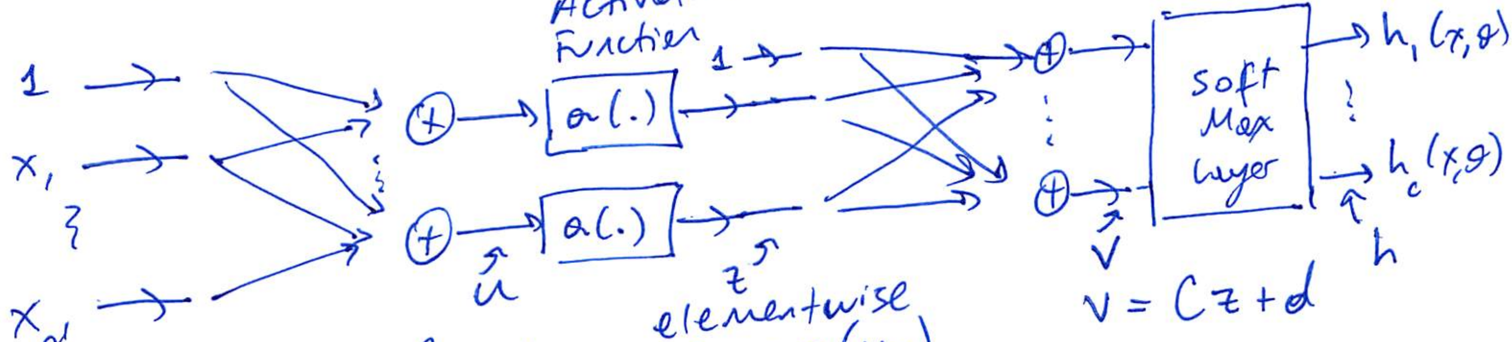
If we define $y_{(l_n)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ $\begin{array}{l} \} \text{ 0's} \\ \leftarrow 1 \text{ at } l_n^{th} \text{ entry} \\ \} \text{ 0's} \end{array}$

$$\hat{\theta}_{ML} = \underset{\theta}{\operatorname{argmin}} -\frac{1}{N} \sum_{n=1}^{N} \left[\sum_{l=1}^{c} y_{(l_n)l} \ln h_l(x_n,\theta) \right]$$

Note that for the inner summation only one term is non-zero. This expression warrents the phrase minimum-cross-entropy training, which is commonly used. It allows us to also use soft labels in y (e.g. if labels are uncertain).

Simple MLP Model

Consider the following simple multilayer perceptron model



$u = Ax + b$

elementwise $z_j = \alpha(u_j)$

or simply $z = \alpha(u)$

$v = Cz + d$

The softmax layer produces $h_i = \dfrac{e^{v_i}}{\sum_{j=1}^{c} e^{v_j}}$ for $i \in \{1, \dots, c\}$

Let's denote it as $h = m(v)$.

The overall model function is

$$h(x,\theta) = m\Big(\underbrace{d + C\, \underbrace{\alpha(b + Ax)}_{1^{st}\text{ layer}}}_{2^{nd}\text{ layer}} \Big)$$

soft max layer

θ = vectorize $\{b, A, d, C\}$ all parameters.

Cross-Entropy Loss

$$\hat{\theta}_{ML} = \arg\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \overbrace{\left[- \sum_{l=1}^{C} y_{(l_n)_l} \ln h_l(x_n, \theta) \right]}^{\text{for sample } (x_n,\, l_n) \text{ cross-entropy loss is thus}}$$

For $y_l \neq 0$

$$= \arg\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \left[\sum_{l=1}^{C} y_{(l_n)_l} \ln y_{(l_n)_l} - \sum_{l=1}^{C} y_{(l_n)_l} \ln h_l(x_n, \theta) \right]$$

$$= \arg\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \left[\sum_{l=1}^{C} y_{(l_n)_l} \ln \frac{y_{(l_n)_l}}{h_l(x_n, \theta)} \right]$$

$$= \arg\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} \underbrace{D_{KL}\left(y_{(l_n)} \,\|\, h(x_n, \theta) \right)}_{\text{KL-divergence between desired } y \text{ and model output } h}$$

So if $y_{(l_n)}$ is a desired posterior distribution for x_n, it turns out, under the assumptions we made, the model trained with MLE is minimizing average KLD between desired and model-output posterior values.

Approximating a function that maps X to y using Max Likelihood parameter estimation under the additive Gaussian noise model.

Given dataset $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$

where $x \in \mathbb{R}^d$, $y \in \mathbb{R}^m$; and model

$$y \approx h(x;\theta) + v \quad \text{with} \quad v \sim \mathcal{N}(0, \Sigma_v),$$

find $\hat{\theta}_{ML}$ that maximizes the likelihood of D.

$$\hat{\theta}_{ML} = \arg\max_{\theta} p(D|\theta) = \arg\max_{\theta} \ln p(D|\theta)$$

iid samples

$$= \arg\max_{\theta} \sum_{n=1}^{N} \ln p(x_n, y_n | \theta)$$

Bayes Rule

$$= \arg\max_{\theta} \sum_{n=1}^{N} \left[\ln p(y_n | x_n \theta) + \ln p(x_n | \theta) \right]$$

$x_n \perp\!\!\!\perp \theta$

$$= \arg\max_{\theta} \sum_{n=1}^{N} \ln p(y_n | x_n \theta) + \overbrace{\sum_{n=1}^{N} \cancel{\ln p(x_n)}}^{\text{constant}}$$

substitute model

$$= \arg\max_{\theta} \sum_{n=1}^{N} \ln \left[(2\pi)^{-m/2} |\Sigma_v|^{-1/2} e^{-\frac{1}{2}(y_n - h(x_n,\theta))^T \cdot \Sigma_v^{-1} (y_n - h(x_n,\theta))} \right.$$

Note that our model $y \approx h(x,\theta) + v$ with $v \sim \mathcal{N}(0, \Sigma_v)$ implies $y_n | x_n \theta \sim \mathcal{N}(h(x_n,\theta), \Sigma_v)$ and we used this in the previous step.

Now simplify:

$$\hat{\theta}_{ML} = \arg\max_{\theta} \overbrace{\cancel{N \ln (2\pi)^{-m/2}}}^{\text{constant}} + \overbrace{\cancel{N \ln |\Sigma_v|^{-1/2}}}^{\text{constant}} - \frac{1}{2} \sum_{n=1}^{N} (y_n - h(x_n,\theta))^T \Sigma_v^{-1} (y_n - h(x_n,\theta))$$

$\times \frac{-2}{N}$

$$= \arg\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} (y_n - h(x_n,\theta))^T \Sigma_v^{-1} (y_n - h(x_n,\theta))$$

Now let's assume $\Sigma_v = \sigma_v^2 I$.

$$\Rightarrow \hat{\theta}_{ML} = \arg\min_{\theta} \frac{1}{N\sigma_v^2} \sum_{n=1}^{N} (y_n - h(x_n,\theta))^T (y_n - h(x_n,\theta))$$

$$= \arg\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} (y_n - h(x_n,\theta))^T (y_n - h(x_n,\theta))$$

$$= \arg\min_{\theta} \underbrace{\frac{1}{N} \sum_{n=1}^{N} \| y_n - h(x_n,\theta) \|_2^2}_{\text{Average (Mean) Squared Error}}$$

Simple MLP Model

$$h(x,\theta) = \underbrace{d + C\underbrace{a(b + Ax)}_{\substack{1^{st}\text{ layer} \\ \text{with nonlinear} \\ \text{activation } a(\cdot)}}}_{\text{Linear } 2^{nd}\text{ layer}}$$

The number of perceptrons in the 1^{st} layer can be adjusted using model selection procedures (e.g. cross-validation). For the activation function soft versions of ReLu can be used (e.g. ELu).

If the values of y are known to be bounded, we can use a nonlinear activation function for the second layer, in order to make sure $h(\cdot)$ produces values in the appropriate range.

<u>Fact</u>. If $P_{XY}(x, y)$ is the joint pdf of X and Y, then $\hat{Y}_{MSE}(x) = E_{XY}\{Y \mid X=x\}$, the conditional expectation of Y given $X=x$ is the minimum-MSE estimator of Y from X.

So our MLE-model parameters yield a function that approximates this conditional expectation.