AML (CS5785) HW2

Jiwon Jeong

October 2025

1 Written Exercises

1.1 Maximum likelihood and KL divergence

Maximizing the likelihood that the model generates the data distribution and minimizing the KL divergence between the data distribution and the model predictions are the same. See Figure 12 for derivation.

1.2 Gradient and log-likelihood for logistic regression

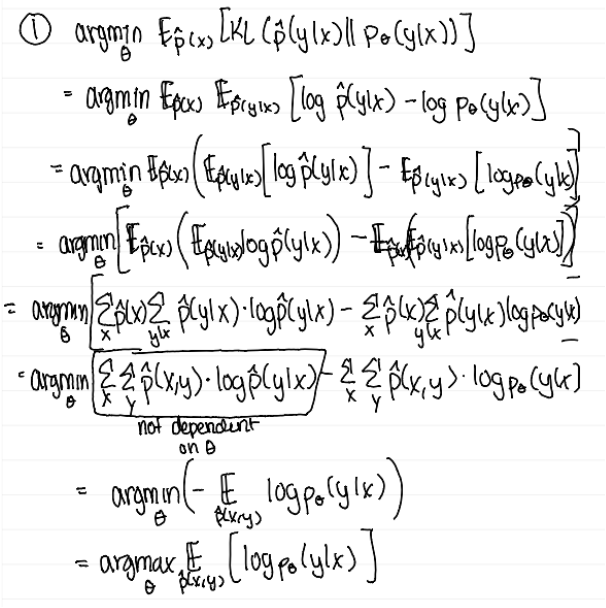
See Figure 13 and 14 for derivation on gradient of the log likelihood loss function.

1.3 Problem with single learning rate

Using a single learning rate for all components in a gradient descent means that all dimensions will use the same learning rate. This can be an issue because a too-high learning rate can break convergence, causing either divergence or jittering. Especially with higher dimensions, a single learning rate might mean that some dimensions converge well, but other dimensions cannot converge. One naive solution would be to just lower the single learning rate until all these dimensions converge, but that could mean that some dimensions now converge too slowly and will not reach convergence in a reasonable time.

1.4 Gradient descent can fail to reach global minimum

Previously alluded, gradient descent can fail to converge if the learning rate for that dimension is too high. There are two versions. If step size is extremely large, then the global minimum cannot be reached, because the steps will cause the solution to overstep and shoot out of convex loss surfaces. This is called divergence. Alternatively, the step size can still be large, causing the solution to step across the minimum to the other size of the convex surface and keep stepping around the minimum in the convex surface. While it may eventually converge, it will take a very long time. This is called jittering.



$$\textcircled{1}\ \underset{\theta}{\operatorname{argmin}}\ \mathbb{E}_{\hat{p}(x)}\left[KL\left(\hat{p}(y|x) \| p_\theta(y|x)\right)\right]$$

$$= \underset{\theta}{\operatorname{argmin}}\ \mathbb{E}_{\hat{p}(x)}\ \mathbb{E}_{\hat{p}(y|x)}\left[\log \hat{p}(y|x) - \log p_\theta(y|x)\right]$$

$$= \underset{\theta}{\operatorname{argmin}}\ \mathbb{E}_{\hat{p}(x)}\left(\mathbb{E}_{\hat{p}(y|x)}\left[\log \hat{p}(y|x)\right] - \mathbb{E}_{\hat{p}(y|x)}\left[\log p_\theta(y|x)\right]\right]$$

$$= \underset{\theta}{\operatorname{argmin}}\left[\mathbb{E}_{\hat{p}(x)}\left(\mathbb{E}_{\hat{p}(y|x)} \log \hat{p}(y|x)\right) - \mathbb{E}_{\hat{p}(x)}\left(\mathbb{E}_{\hat{p}(y|x)}\left[\log p_\theta(y|x)\right]\right)\right]$$

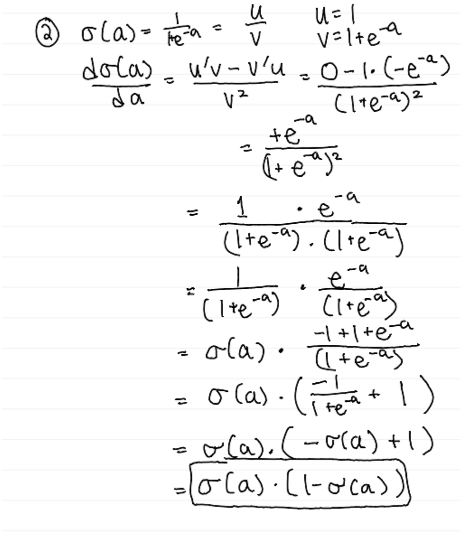
$$= \underset{\theta}{\operatorname{argmin}}\left[\sum_x \hat{p}(x) \sum_{y|x} \hat{p}(y|x) \cdot \log \hat{p}(y|x) - \sum_x \hat{p}(x) \sum_{y|x} \hat{p}(y|x) \log p_\theta(y|x)\right]$$

$$= \underset{\theta}{\operatorname{argmin}}\left[\underbrace{\boxed{\sum_x \sum_y \hat{p}(x,y) \cdot \log \hat{p}(y|x)}}_{\text{not dependent on } \theta} - \sum_x \sum_y \hat{p}(x,y) \cdot \log p_\theta(y|x)\right]$$

$$= \underset{\theta}{\operatorname{argmin}}\left(- \mathbb{E}_{\hat{p}(x,y)} \log p_\theta(y|x)\right)$$

$$= \underset{\theta}{\operatorname{argmax}}\ \mathbb{E}_{\hat{p}(x,y)}\left[\log p_\theta(y|x)\right]$$

Figure 1: Derivation for equality of minimizing KL divergence and maximum likelihood estimation



$$\textcircled{2}\ \sigma(a) = \frac{1}{1+e^{-a}} = \frac{u}{v} \qquad u = 1,\ v = 1+e^{-a}$$

$$\frac{d\sigma(a)}{da} = \frac{u'v - v'u}{v^2} = \frac{0 - 1 \cdot (-e^{-a})}{(1+e^{-a})^2}$$

$$= \frac{+e^{-a}}{(1+e^{-a})^2}$$

$$= \frac{1 \cdot e^{-a}}{(1+e^{-a}) \cdot (1+e^{-a})}$$

$$= \frac{1}{(1+e^{-a})} \cdot \frac{e^{-a}}{(1+e^{-a})}$$

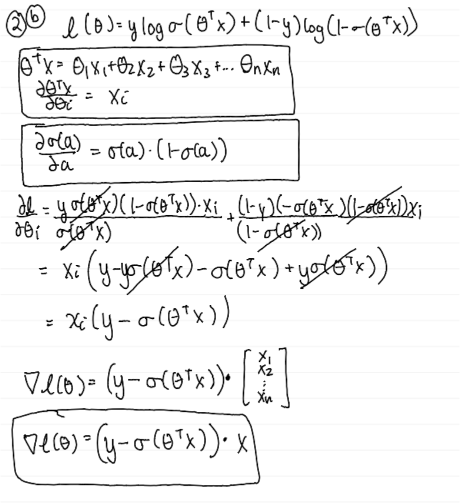
$$= \sigma(a) \cdot \frac{-1+1+e^{-a}}{(1+e^{-a})}$$

$$= \sigma(a) \cdot \left(\frac{-1}{1+e^{-a}} + 1\right)$$

$$= \sigma(a) \cdot \left(-\sigma(a) + 1\right)$$

$$= \boxed{\sigma(a) \cdot (1 - \sigma(a))}$$

Figure 2: Derivative of sigmoid function



②ⓑ $\ell(\theta) = y \log \sigma(\theta^T x) + (1-y) \log(1-\sigma(\theta^T x))$

$$\boxed{\begin{array}{l} \theta^T x = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots \theta_n x_n \\ \frac{\partial \theta^T x}{\partial \theta_i} = x_i \end{array}}$$

$$\boxed{\frac{\partial \sigma(a)}{\partial a} = \sigma(a) \cdot (1-\sigma(a))}$$

$$\frac{\partial \ell}{\partial \theta_i} = \frac{y \cancel{\sigma(\theta^T x)}(1-\sigma(\theta^T x)) \cdot x_i}{\cancel{\sigma(\theta^T x)}} + \frac{(1-y)(-\sigma(\theta^T x))\cancel{(1-\sigma(\theta^T x))} x_i}{\cancel{(1-\sigma(\theta^T x))}}$$

$$= x_i \left(y - \cancel{y\sigma(\theta^T x)} - \sigma(\theta^T x) + \cancel{y\sigma(\theta^T x)}\right)$$

$$= x_i (y - \sigma(\theta^T x))$$

$$\nabla \ell(\theta) = (y - \sigma(\theta^T x)) \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\boxed{\nabla \ell(\theta) = (y - \sigma(\theta^T x)) \cdot x}$$

Figure 3: Gradient of log likelihood

However, gradient descent can also fail to reach global minima under other conditions. If the learning rate is too low and/or the loss surface is very flat in regions away from the minimum, then the solution will step so slowly that the minimum is never reached in realistic time. Particularly concerning this idea of the loss surface being very flat, the loss surface can have flat points or stationary points if there are many collinear features.

Thus, ideally the step size is optimized to not be too large for divergence/jittering or too small for unrealistic convergence times. In fact for convex loss surfaces, there is an optimal step size for which we can reach the minimum in one step.

However, even gradient descents with optimized step sizes can still fail to reach a global minimum. This can happen for two main reasons. First, if there are local minima, it is possible that the specific initialization brings the solution to converge to a local minimum. This is the most concern for nonconvex loss surfaces. Second, if parts of the loss surface is nondifferentiable, the vanilla computation for gradient descent can get stuck at nondifferentiable points.

1.5 Decaying learning rate

When getting close to a (hopefully global) minimum and within the local convex surface, we need a small step size to guarantee convergence. Otherwise, a large step size could cause divergence or very slow convergence through jittering.

However, having a small step size throughout the whole optimization can be a bad idea, because we want to quickly approach the minima. The loss surface might be relatively flat when farther away from the minima, so a larger step size is necessary to reach the minima quickly. In addition, for complex non-convex loss surfaces, a large initial step size can help escape local minima.

Thus, we initially set a high learning rate to take faster steps towards the minima and possibly escape local minima. Then we gradually decrease the learning rate, so that it is eventually possible to converge in a local convex surface towards the (hopefully global) minimum. This change in learning rate is typically performed through 3 decay schedules. Starting from a large learning rate of greater than 2, the learning rate is decreased with each training loop iteration. The linear decay schedule decreases rate based on number of iterations in a $1/x$ rational decay. The quadratic decay schedule decreases rate based on number of iterations in a $1/x^2$ rational decay. The exponential decay schedule decreases rate based on number of iterations in a $1/e^x$ exponential decay.

Using these decaying learning rates means that convergence will initially not occur. There will be divergence away from (local) minima and large step sizes. With iterations, the steps will get smaller and hopefully allow the solution to approach a minima. It is important to choose the right decay schedule so that the step size does not decay too quickly (leading to slow optimization from too small steps) or decay too slowly (leading to slow optimization from unnecessary divergence/jittering).