MSE loss and classification problems

Akash Garg

Version 1.0 - October 7, 2020

One needs a loss function. For classification problems, cross-entropy is a popular choice, but why not mean-squared error (MSE) To answer this we need to consider where MSE comes from. To understand MSE we need to understand some basic concepts from bayesian inference.

Suppose we have our model such that $F(x, \theta) \mapsto y$ where x are the inputs, y outputs and θ are the model parameters. Furthermore, let $D = (x_n, y_n)$ are the observations / data that is drawn from an unknown probability distribution $p(D)$. Our goal is to approximate $p(D)$ through a $q(D|\theta)$. Reall that one way to compare two distributions is by using KL-Divergence:

$$D_{\mathrm{KL}}(p(D)\|q(D|\theta)) = \sum_D p(D) \log\left(\frac{p(D)}{q(D|\theta)}\right)$$

We want to minimize the divergence between p and q. Since $p(D)$ does not depend on the model parameters θ we can safely ignore those terms and we are left with:

$$\mathrm{argmax}_\theta q(D|\theta).$$

We have observations x_i, outputs y_i and the predictions from our model $F(x, \theta) = \hat{y}_i$. Note that we have some unknown error between our observations y and our predictions $\hat{y}$ s.t.

$$y_i = \hat{y}_i + \epsilon.$$

If we assume that the error ϵ has a normal distribution around the mean with variance σ then we can have:

$$p(y|x) = \mathcal{N}(y; \hat{y}, \sigma^2).$$

Using our formulation to minimize KL divergence and the product rule of conditional probability, we get:

$$L(y|x) = \mathrm{argmax}_\theta \prod_n p(y_n|x_n, \theta) = \mathrm{argmax}_\theta \prod_n \frac{1}{2\sigma^2\pi} \exp\left(-\frac{(y_n - \hat{y_n})^2}{2\sigma^2}\right).$$

Where L is our "loss" function. All we have done here is subsitute the gaussian for $p(y|x)$. We can then take the log of both sides to simplify further:

$$\log L(y|x) = \log \prod_n \frac{1}{2\sigma^2\pi} \exp\left(-\frac{(y_n - \hat{y}_n)^2}{2\sigma^2}\right)$$
$$= \sum \log\left(\frac{1}{2\sigma^2\pi}\right) + \left(-\frac{(y_n - \hat{y}_n)^2}{2\sigma^2}\right)$$

Since we want to maximize L we can ignore the constant terms and assume that $\sigma = 1$ since it won't affect the function extrema. Thus we get:

$$\text{argmax} L(y|x) = \text{argmax}_{\hat{y}} -\frac{1}{2n}\sum (y_n - \hat{y}_n)^2 = \text{argmin}_{\hat{y}} \frac{1}{2}\sum (y_n - \hat{y}_n)^2. \tag{1}$$

where we changed the argmax into an argmin and removed the minus sign. We now have our MSE equation.

The salient point here is that we got here by assuming a normal distribution on the prediction. However, for classification problems the true distribution is not normal, but closer to a Binomial distribution with values of 1 or 0. There is a proof of this that leads to the binary cross entropy term by Rafay Khan [Kha19].

The second reason for not using MSE for classification problems is that when combined with the sigmoid activation (which seems common practice), we do not get a convex function. One can see this by evaluating the second derivative of MSE with sigmoid activation and note that it is not positive everywhere. A proof is given here [Bha19].

References

[Bha19] Rajesh Bhat. *Why not Mean Squared Error(MSE) as a loss function for Logistic Regression?* 2019. URL: https://towardsdatascience.com/why-not-mse-as-a-loss-function-for-logistic-regression-589816b5e03c.

[Kha19] Rafay Khan. *Where did the Binary Cross-Entropy Loss Function come from?* 2019. URL: https://towardsdatascience.com/where-did-the-binary-cross-entropy-loss-function-come-from-ac3de349a715.

Revision	Date	Description
1.0	October 7, 2020	Initial draft