Homework # 4

TJ Bai

December 10, 2023

1. We show that the posterior $p(y = c \mid x)$ resulting from a Gaussian generative model with equal, isotropic covariance matrices is equivalent to the posterior in logistic regression.

$$p(y = c \mid x) = \frac{p(x \mid y = c)p(y = c)}{p(x)} = \frac{p(x \mid y = c)p(y = c)}{p(x \mid y = 1)p(y = 1) + p(x \mid y = 0)p(y = 0)}$$

$$p(x|y = 0) = \prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x_j - \mu_{0j}}{\sigma}\right)^2\right\} = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^d \exp\left\{-\frac{1}{2\sigma^2}\sum_{j=1}^{d}\left(x_j - \mu_{0j}\right)^2\right\}$$

Similarly, $p(x|y = 1) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^d \exp\left\{-\frac{1}{2\sigma^2}\sum_{j=1}^{d}\left(x_j - \mu_{1j}\right)^2\right\}$

We then have $\frac{p(x \mid y = 0)}{p(x \mid y = 1)} = \exp\left\{-\frac{1}{2\sigma^2}\sum_{j=1}^{d}\left[(x_j - \mu_{0j})^2 - (x_j - \mu_{1j})^2\right]\right\}$

$$= \exp\left\{-\frac{1}{2\sigma^2}\sum_{j=1}^{d}\left[x_j^2 - x_j^2 - 2x_j\mu_{0j} + 2x_j\mu_{1j} + \mu_{0j}^2 - \mu_{1j}^2\right]\right\}$$

$$= \exp\left\{-\frac{1}{2\sigma^2}\sum_{j=1}^{d}\left[x_j(2\mu_{1j} - 2\mu_{0j}) + \mu_{0j}^2 - \mu_{1j}^2\right]\right\}$$

Similarly, $\frac{p(x \mid y = 1)}{p(x \mid y = 0)} = \exp\left\{-\frac{1}{2\sigma^2}\sum_{j=1}^{d}\left[x_j(2\mu_{0j} - 2\mu_{1j}) + \mu_{1j}^2 - \mu_{0j}^2\right]\right\}$

We first consider $p(y = 1 \mid x) = \frac{1}{1 + \frac{p(x|y=0)p(y=0)}{p(x|y=1)p(y=1)}}$

$$\frac{p(x \mid y = 0)p(y = 0)}{p(x \mid y = 1)p(y = 1)} = \exp\left\{-\frac{1}{2\sigma^2}\sum_{j=1}^{d}\left[x_j(2\mu_{1j} - 2\mu_{0j}) + \mu_{0j}^2 - \mu_{1j}^2\right]\right\}\exp\left\{\log\frac{p(y = 0)}{p(y = 1)}\right\}$$

$$= \exp\left\{\sum_{j=1}^{d} x_j\left(\frac{\mu_{0j} - \mu_{1j}}{\sigma^2}\right) - \frac{1}{2\sigma^2}\sum_{j=1}^{d}(\mu_{0j}^2 - \mu_{1j}^2) + \log\frac{p(y = 0)}{p(y = 1)}\right\}$$

This is equivalent to logistic regression for. . .

$$w_0 = -\frac{1}{2\sigma^2}\sum_{j=1}^{d}(\mu_{0j}^2 - \mu_{1j}^2) + \log\frac{p(y = 0)}{p(y = 1)}$$

$$w_j = \frac{\mu_{0j} - \mu_{1j}}{\sigma^2},\ \forall j = 1, \ldots, d$$

We can follow the same steps for $p(y=0|x) = \dfrac{1}{1+\frac{p(x|y=1)p(y=1)}{p(x|y=0)p(y=0)}}$

$$\frac{p(x \mid y=1)p(y=1)}{p(x \mid y=0)p(y=0)} = \exp\left\{\sum_{j=1}^{d} x_j\left(\frac{\mu_{1j}-\mu_{0j}}{\sigma^2}\right) - \frac{1}{2\sigma^2}\sum_{j=1}^{d}(\mu_{1j}^2-\mu_{0j}^2) + \log\frac{p(y=1)}{p(y=0)}\right\}$$

This is equivalent to logistic regression for. . .

$$w_0 = \sum_{j=1}^{d}(\mu_{1j}^2-\mu_{0j}^2) + \log\frac{p(y=1)}{p(y=0)}$$

$$w_j = \frac{\mu_{1j}-\mu_{0j}}{\sigma^2},\ \forall j = 1,\ldots,d$$

2. Although both models have the same posterior form, they will not necessarily produce the same classifier because they optimize different objective functions. Whereas logstic regression relies on the conditional log-likelihood, the generative model relies on the *joint* log-likelihood, bringing in information from the marginal distribution $p(x)$ to create predictions. As a result, LDA necessarily makes stronger assumptions about the form of the data, such as being normally distributed with equal, isotropic covariance matrices. Resultingly, the produced classifiers will differ.

3. We show that least squares regression with dropout is equivalent to ridge regression.

$$\hat{L}_{\text{Dropout}}(w) = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{d}\epsilon_j w_j x_{ij}\right)^2\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left(y_i - \sum_{j=1}^{d}\epsilon_j w_j x_{ij}\right)^2\right]$$

$$\mathbb{E}\left[\left(y_i - \sum_{j=1}^{d}\epsilon_j w_j x_{ij}\right)^2\right] = y_i^2 - \sum_{j=1}^{d}2y_i\mathbb{E}[\epsilon_j]w_j x_{ij} + \sum_{j_1=1}^{d}\sum_{j_2=1}^{d}\mathbb{E}[\epsilon_{j_1}\epsilon_{j_2}]w_{j_1}w_{j_2}x_{ij_1}x_{ij_2}$$

Note that $\mathbb{E}[\epsilon_j] = 1$ and $\mathbb{E}[\epsilon_{j_1}\epsilon_{j_2}] = 1$ when $j_1 \neq j_2$, otherwise $\frac{1}{p}$.

$$\mathbb{E}\left[\left(y_i - \sum_{j=1}^{d}\epsilon w_j x_{ij}\right)^2\right] = y_i^2 - \sum_{j=1}^{d}2y_i w_j x_{ij} + \sum_{j=1}^{d}\frac{1}{p}w_j^2 x_{ij}^2 + \sum_{j_1\neq j_2} w_{j_1}w_{j_2}x_{ij_1}x_{ij_2}$$

$$= \left(y_i - \sum_{j=1}^{d} w_j x_{ij}\right)^2 + \frac{1-p}{p}\sum_{j=1}^{d} w_j^2 x_{ij}^2$$

$$\implies \hat{L}_{\text{Dropout}}(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^\top x_i)^2 + \frac{1-p}{np}\sum_{i=1}^{n}\sum_{j=1}^{d} w_j^2 x_{ij}^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}(y_i - w^\top x_i)^2 + \frac{1-p}{np}\|\Gamma^\top w\|^2$$

This is a form of generalized ridge regression where $\Gamma_j = \sqrt{\sum_{i=1}^{n} x_{ij}^2}$ and $\lambda = \dfrac{1-p}{np}$.

Knowing that $\hat{\mu}_j = 0$, we can understand this form of ridge regression as scaling the jth dimension of the weights by the standard deviation of the jth dimensions of the inputs.

If the data were not normalized, then a higher-than-unit variance would be equivalent to increasing the regularization penalty, while a lower-than-unit variance would decrease the penalty.

Additionally, as the mean of the data increases/decreases away from 0, this would also increase the regularization penalty because the values of the x_{ij}^2 terms would increase.

As p approaches 1, $\lambda \to 0$. As p approaches 0, $\lambda \to \infty$.

This result is also intuitive, because $p = 1$ would be equivalent to no dropout and thus no regularization. Meanwhile, $p = 0$ would be equivalent to infinite regularization.

4. Each update algorithm resulted in relatively similar final validation errors (4.5, 4.65, 4.4). From the training runs, it appears that SGD with momentum and SGD with Nestorov momentum both converged faster than vanilla SGD. Additionally, SGD with Nestorov momentum appeared to handle oscillations in the loss surface better. The validation error rarely if ever increased and converged the quickest out of the 3.

   I implemented the ReLU and logistic activation functions, and also implemented dropout layers. From various ablation tests, ReLU and logistic did not have significantly different results. Rather, the width and number of hidden layers were more relevant (the best performing NN had 2 hidden layers, each dimension 400). When dropout layers were integrated, the validation error strictly increased by 1-2%. Dropout could perhaps be more useful with significantly more training data and longer training runs, where regularization is more important to prevent regularization.

   Ultimately, a NN with baseline hyperparameters, 2 hidden layers each with dimension 400, and ReLU activations performed the best, achieving 95.6% accuracy on the kaggle dataset.