Computing the Gradient for Logistic Regression

Math 3180

Binary Logistic Regression: Setup

Log-Likelihood

L = probability that the image is a cat or not for a particular choice of W

Pick W so that Likelihood of data is a maximum

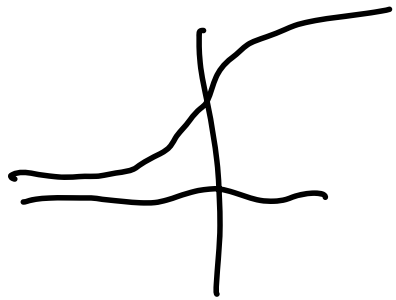
$$\log L = Y^T \log(\sigma(XW)) + (1 - Y)^T \log(1 - \sigma(XW))$$

Dimensions:

- X is $N \times K$ (data matrix)
- W is $K \times 1$ (weight vector)
- Y is $N \times 1$ (labels: 0s and 1s)

The Sigmoid Function

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



Its derivative:

$$\begin{aligned}
\frac{d}{dx}\sigma(x) &= -\frac{1}{(1 + e^{-x})^2} \cdot (-e^{-x}) \\
&= \left(\frac{e^{-x}}{1 + e^{-x}}\right)\left(\frac{1}{1 + e^{-x}}\right) \\
&= \sigma(x)\big(1 - \sigma(x)\big)
\end{aligned}$$

Component Form of $\sigma(XW)$

$N \times K \quad K \times 1 \qquad XW \qquad N \times 1$

$\sigma(XW)$ is an N-vector whose i-th entry is

$$\sigma(XW)_i = \sigma\left(\sum_{j=1}^{K} x_{ij}\, w_j\right).$$

Therefore the first term in L expands as

$$Y^T \log \sigma(XW) = \sum_{i=1}^{N} y_i \log \sigma\left(\sum_{j=1}^{K} x_{ij}\, w_j\right).$$

$$\frac{\partial}{\partial w_s}$$

$s = 1, \ldots, K$

Gradient: Differentiating the First Term

Applying the chain rule to $\frac{\partial}{\partial w_s}$:

$$\frac{\partial}{\partial w_s} \sum_{i=1}^{N} y_i \log \sigma\left(\sum_{j=1}^{K} x_{ij} w_j\right)$$

$$= \sum_{i=1}^{N} y_i \cdot \frac{1}{\sigma(\sum_{j=1}^{K} x_{ij} w_j)} \cdot \sigma(\sum_{j=1}^{K} x_{ij} w_j)\big(1 - \sigma(\sum_{j=1}^{K} x_{ij} w_j)\big) \cdot x_{is}$$

$$= \sum_{i=1}^{N} y_i \left(1 - \sigma\left(\sum_{j=1}^{K} x_{ij} w_j\right)\right) x_{is}$$

Gradient: Differentiating the Second Term

$$\frac{\partial}{\partial w_s}(1-Y)^T \log\big(1-\sigma(XW)\big)$$

$$= \frac{\partial}{\partial w_s}\sum_{i=1}^{N}(1-y_i)\,\log\Big(1-\sigma\Big(\textstyle\sum_{j=1}^{K} x_{ij}w_j\Big)\Big)$$

$$= \sum_{i=1}^{N}(1-y_i)\cdot\frac{-\sigma(\sum_{j=1}^{K} x_{ij}w_j)\big(1-\sigma(\sum_{j=1}^{K} x_{ij}w_j)\big)}{1-\sigma(\sum_{j=1}^{K} x_{ij}w_j)}\cdot x_{is}$$

$$= -\sum_{i=1}^{N}(1-y_i)\,\sigma\Big(\textstyle\sum_{j=1}^{K} x_{ij}w_j\Big)\,x_{is}$$

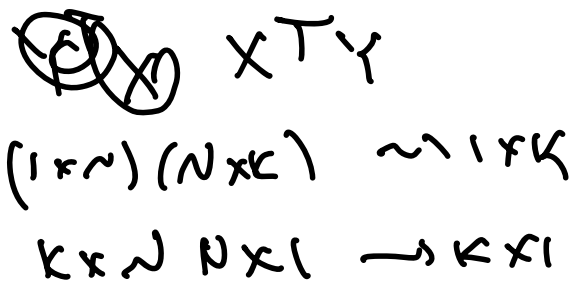
Combining Both Terms

$$\frac{\partial L}{\partial w_s} = \sum_{i=1}^{N} y_i \big(1 - \sigma(XW)_i\big) x_{is} - \sum_{i=1}^{N} (1 - y_i)\, \sigma(XW)_i\, x_{is}$$

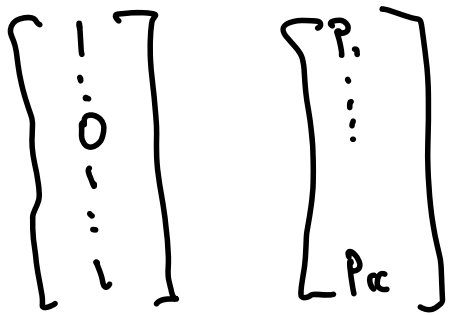
$$= \sum_{i=1}^{N} y_i\, x_{is} - \sum_{i=1}^{N} y_i\, \sigma(XW)_i\, x_{is}$$

$$- \sum_{i=1}^{N} \sigma(XW)_i\, x_{is} + \sum_{i=1}^{N} y_i\, \sigma(XW)_i\, x_{is}$$

$$= \sum_{i=1}^{N} y_i\, x_{is} - \sum_{i=1}^{N} \sigma(XW)_i\, x_{is}$$



Matrix Form: Binary Case



Collecting all components $s = 1, \ldots, K$ gives

$$\boxed{\nabla_W L = X^T Y - X^T P = X^T (Y - P)}$$

where $P = \sigma(XW)$ is the $N \times 1$ vector of predicted probabilities.

Dimension check: X^T is $K \times N$, $Y - P$ is $N \times 1$, so $\nabla_W L$ is $K \times 1$. ✓

iterate

$$W \longrightarrow W - \lambda \nabla_W L$$

Multiclass Classification: Setup

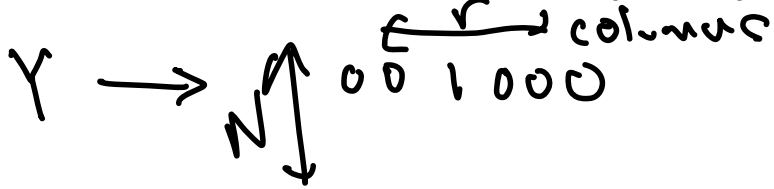
Log-Likelihood

$$\log L = \operatorname{tr}\left(Y^T \log \sigma(XW)\right)$$

where σ is now the **softmax** function (applied row-by-row), and $\operatorname{tr}(\cdot)$ denotes the sum of diagonal entries.

Dimensions:

- X is $N \times K$ (data matrix)
- W is $K \times r$ (weight matrix)
- Y is $N \times r$ (one-hot encoding)
- XW is $N \times r$, $\sigma(XW)$ is $N \times r$



Softmax Function

For a vector $z = (z_1, \ldots, z_r)$, set $F = \sum_{t=1}^{r} e^{z_t}$. Then

$$\sigma(z)_k = \frac{e^{z_k}}{F}.$$

Applied row-by-row to XW, the (i, t) entry of $\sigma(XW)$ is

$$\sigma(XW)_{it} = \frac{e^{z_{it}}}{F_i}, \qquad z_{it} = \sum_{s=1}^{K} x_{is}\, w_{st}, \qquad F_i = \sum_{t'=1}^{r} e^{z_{it'}}.$$

Expanding the Log-Likelihood

Using $\operatorname{tr}(Y^T M) = \sum_{i=1}^{N} \sum_{t=1}^{r} y_{it} m_{it}$:

$$\log L = \sum_{i=1}^{N} \sum_{t=1}^{r} y_{it} \log \sigma(XW)_{it}.$$

Key simplification. Since $\sigma(XW)_{it} = e^{z_{it}} / F_i$:

$$\log \sigma(XW)_{it} = \log e^{z_{it}} - \log F_i = z_{it} - \log F_i.$$

Therefore

$$L = \sum_{i=1}^{N} \sum_{t=1}^{r} y_{it} z_{it} - \sum_{i=1}^{N} \left(\sum_{t=1}^{r} y_{it} \right) \log F_i = \sum_{i=1}^{N} \sum_{t=1}^{r} y_{it} z_{it} - \sum_{i=1}^{N} \log F_i,$$

where the last step uses $\sum_{t=1}^{r} y_{it} = 1$ (one-hot).

Computing $\partial L / \partial w_{pq}$

W is $K \times r$

We want the (p, q) entry of the $K \times r$ gradient matrix.

$$\frac{\partial L}{\partial w_{pq}} = \sum_{i=1}^{N} \sum_{t=1}^{r} y_{it} \frac{\partial}{\partial w_{pq}} (z_{it} - \log F_i).$$

The two derivatives:

- Since $z_{it} = \sum_{s=1}^{K} x_{is} w_{st}$: $\quad \dfrac{\partial z_{it}}{\partial w_{pq}} = \begin{cases} x_{ip} & t = q \\ 0 & t \neq q \end{cases}$
- $\dfrac{\partial \log F_i}{\partial w_{pq}} = \dfrac{e^{z_{iq}} x_{ip}}{F_i} = \sigma(XW)_{iq} \cdot x_{ip}$

Define the Kronecker delta $\delta_{tq} = \begin{cases} 1 & t = q \\ 0 & t \neq q \end{cases}$. Then:

$$\frac{\partial \log \sigma(XW)_{it}}{\partial w_{pq}} = x_{ip}(\delta_{tq} - \sigma(XW)_{iq}).$$

Summing Over Classes: The $t = q$ and $t \neq q$ Cases

$$\frac{\partial L}{\partial w_{pq}} = \sum_{i=1}^{N} \sum_{t=1}^{r} y_{it}\, x_{ip} (\delta_{tq} - \sigma(XW)_{iq})$$

$$= \sum_{i=1}^{N} x_{ip} \underbrace{\left(\sum_{t=1}^{r} y_{it} \delta_{tq} \right)}_{= y_{iq}} - \sum_{i=1}^{N} x_{ip}\, \sigma(XW)_{iq} \underbrace{\left(\sum_{t=1}^{r} y_{it} \right)}_{= 1}$$

$$= \sum_{i=1}^{N} x_{ip} (y_{iq} - \sigma(XW)_{iq}) = \underbrace{\left(X^T (Y - P)\right)_{pq}}$$

where we used the one-hot property: $\sum_{t=1}^{r} y_{it} \delta_{tq} = y_{iq}$ and $\sum_{t=1}^{r} y_{it} = 1$.

Matrix Form: Multiclass Case

Collecting all entries (p, q):

$$\boxed{\nabla_W L = X^T(Y - P)}$$

where $P = \sigma(XW)$ is the $N \times r$ matrix of softmax probabilities.

The gradient has exactly the same form as in the binary case:

	σ	W
Binary	sigmoid	$K \times 1$
Multiclass	softmax	$K \times r$

In both cases: $\nabla_W L = X^T(Y - P)$.