1 Logistic Function and Log Odds

Problem 1. Divide the numerator and the denominator by $p(t=1|\mathbf{x})$ (assume both class probabilities are positive), then write the ratio as exp ln of itself:

$$\begin{aligned} p(t=1|\mathbf{x}) &= \frac{p(t=1|\mathbf{x})}{p(t=1|\mathbf{x}) + p(t=2|\mathbf{x})} \\ &= \frac{1}{1 + \dfrac{p(t=2|\mathbf{x})}{p(t=1|\mathbf{x})}} \\ &= \frac{1}{1 + \exp\left(\ln \dfrac{p(t=2|\mathbf{x})}{p(t=1|\mathbf{x})}\right)} \\ &= \frac{1}{1 + \exp\left(-\ln \dfrac{p(t=1|\mathbf{x})}{p(t=2|\mathbf{x})}\right)} \\ &= \frac{1}{1+\exp(-z)} = \sigma(z), \end{aligned}$$

where $z = \ln \dfrac{p(t=1|\mathbf{x})}{p(t=2|\mathbf{x})}$. The fourth line uses $\ln(1/r) = -\ln r$.

Problem 2. Logistic regression models $p(t=1|\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x})$. Problem 1 shows that $p(t=1|\mathbf{x}) = \sigma(z)$ with z the log odds. The sigmoid is strictly increasing, so as a map from $\mathbb{R}$ onto $(0,1)$ it is invertible, and its inverse is the log-odds (logit) function $\sigma^{-1}(s) = \ln \frac{s}{1-s}$ for $s \in (0,1)$. Matching the two expressions therefore gives

$$\boxed{\mathbf{w}^\top \mathbf{x} \text{ models } \ln \frac{p(t=1|\mathbf{x})}{p(t=2|\mathbf{x})},}$$

the log odds of class 1 against class 2. Logistic regression assumes the log odds are a linear function of $\mathbf{x}$. With the lecture labels $t \in \{0,1\}$, the same argument shows that $\mathbf{w}^\top \mathbf{x}$ models $\ln \frac{p(t=1|\mathbf{x})}{p(t=0|\mathbf{x})}$.

2 Multiclass Logistic Regression

Throughout, assume every class probability is positive, so the log odds are defined.

Problem 1. Multiply and divide by $p(t = K|\mathbf{x})$, then write the ratio as exp ln of itself:

$$\begin{aligned} p(t = j|\mathbf{x}) &= p(t = K|\mathbf{x}) \cdot \frac{p(t = j|\mathbf{x})}{p(t = K|\mathbf{x})} \\ &= p(t = K|\mathbf{x}) \cdot \exp\left(\ln \frac{p(t = j|\mathbf{x})}{p(t = K|\mathbf{x})}\right) \\ &= p(t = K|\mathbf{x}) \cdot \exp(z_j). \end{aligned} \tag{1}$$

Problem 2. The class probabilities sum to one. Using Equation (1) for each $i \leq K - 1$,

$$\begin{aligned} p(t = K|\mathbf{x}) &= 1 - \sum_{i=1}^{K-1} p(t = i|\mathbf{x}) \\ &= 1 - p(t = K|\mathbf{x}) \sum_{i=1}^{K-1} \exp(z_i). \end{aligned}$$

Move the $p(t = K|\mathbf{x})$ terms to one side and factorise:

$$p(t = K|\mathbf{x}) \left(1 + \sum_{i=1}^{K-1} \exp(z_i)\right) = 1 \quad \Longrightarrow \quad p(t = K|\mathbf{x}) = \frac{1}{1 + \sum_{i=1}^{K-1} \exp(z_i)}. \tag{2}$$

Problem 3. Substitute Equation (2) into Equation (1):

$$p(t = j|\mathbf{x}) = \frac{\exp(z_j)}{1 + \sum_{i=1}^{K-1} \exp(z_i)}.$$

The definition of z_j also makes sense for $j = K$:

$$z_K = \ln \frac{p(t = K|\mathbf{x})}{p(t = K|\mathbf{x})} = \ln 1 = 0, \qquad \exp(z_K) = 1.$$

So the 1 in the denominator is $\exp(z_K)$, and

$$p(t = j|\mathbf{x}) = \frac{\exp(z_j)}{\exp(z_K) + \sum_{i=1}^{K-1} \exp(z_i)} = \frac{\exp(z_j)}{\sum_{i=1}^{K} \exp(z_i)}.$$

By Equation (2), the same formula also holds for $j = K$. Stacking the K probabilities gives the softmax function, $\text{softmax}(\mathbf{z})_j = \exp(z_j) / \sum_{i=1}^{K} \exp(z_i)$.

For $K = 2$, this gives $p(t = 1|\mathbf{x}) = \dfrac{\exp(z_1)}{\exp(z_1) + 1} = \dfrac{1}{1 + \exp(-z_1)} = \sigma(z_1)$, the logistic function from the previous problem.

Clarification of the parameterisation. In Problems 1–3, z_j denotes log odds relative to class K, so $z_K = 0$. To retain this meaning when writing $z_j = \mathbf{w}_{(j)}^\top \mathbf{x}$, we can fix $\mathbf{w}_{(K)} = \mathbf{0}$ and learn the other $K-1$ rows. Alternatively, if all K rows of $\mathbf{W}$ are learned freely, $z_j = \mathbf{w}_{(j)}^\top \mathbf{x}$ is a softmax *score*, and the log odds are

$$\ln \frac{p(t=j|\mathbf{x},\mathbf{W})}{p(t=K|\mathbf{x},\mathbf{W})} = z_j - z_K = (\mathbf{w}_{(j)} - \mathbf{w}_{(K)})^\top \mathbf{x}.$$

Subtracting z_K from every score leaves the softmax probabilities unchanged and recovers the reference-class form. The error function below is valid for either parameterisation.

Problem 4. Let the training set be $\mathcal{D} = \{(\mathbf{x}_n, t_n)\}_{n=1}^N$ with $t_n \in \{1, \ldots, K\}$, and assume the examples are independent given the inputs. Let $\mathbb{I}[A]$ be 1 if A is true and 0 otherwise. For a single example, write the likelihood as

$$p(t_n|\mathbf{x}_n, \mathbf{W}) = \prod_{k=1}^K p(t_n = k|\mathbf{x}_n, \mathbf{W})^{\mathbb{I}[t_n=k]}.$$

Exactly one indicator equals 1, so the product picks out the probability of the observed class. This is the probability mass function of the categorical distribution, which generalises the Bernoulli distribution to K classes.

The log-likelihood of the data is

$$\begin{aligned}
\log p(\mathcal{D}|\mathbf{W}) &= \log \prod_{n=1}^N \prod_{k=1}^K p(t_n = k|\mathbf{x}_n, \mathbf{W})^{\mathbb{I}[t_n=k]} \\
&= \sum_{n=1}^N \sum_{k=1}^K \mathbb{I}[t_n = k] \log p(t_n = k|\mathbf{x}_n, \mathbf{W}) \\
&= \sum_{n=1}^N \sum_{k=1}^K \mathbb{I}[t_n = k] \left(\mathbf{w}_{(k)}^\top \mathbf{x}_n - \log \sum_{j=1}^K \exp(\mathbf{w}_{(j)}^\top \mathbf{x}_n) \right) \\
&= \sum_{n=1}^N \left(\mathbf{w}_{(t_n)}^\top \mathbf{x}_n - \log \sum_{j=1}^K \exp(\mathbf{w}_{(j)}^\top \mathbf{x}_n) \right),
\end{aligned}$$

where the last line uses $\sum_k \mathbb{I}[t_n = k] = 1$. Maximising the likelihood is the same as minimising the negative log-likelihood, so the error function is

$$\boxed{E(\mathbf{W}) = -\log p(\mathcal{D}|\mathbf{W}) = \sum_{n=1}^N \Big(\text{logsumexp}(\mathbf{W}\mathbf{x}_n) - \mathbf{w}_{(t_n)}^\top \mathbf{x}_n \Big),}$$

with $\text{logsumexp}(\mathbf{z}) = \log \sum_{j=1}^K \exp(z_j)$. Machine learning libraries compute this term in a numerically stable way, for example `torch.logsumexp`.

Equivalently, encode the label as a one-hot vector $\mathbf{t}_n \in \{0,1\}^K$ with a 1 in entry t_n (bold $\mathbf{t}_n$ is the vector; plain t_n is the class label). Then

$$E(\mathbf{W}) = -\sum_{n=1}^N \mathbf{t}_n^\top \log \text{softmax}(\mathbf{W}\mathbf{x}_n),$$

where log is applied entrywise. This is the cross-entropy error. For $K = 2$ it reduces to the logistic regression error with weights $\mathbf{w} = \mathbf{w}_{(1)} - \mathbf{w}_{(2)}$.

Remark. When all K rows of $\mathbf{W}$ are learned, adding the same vector to every row leaves all the probabilities unchanged.