Logistic/Softmax Regression by Moment Matching

David S. Rosenberg

1 Logistic regression

Consider the conditional probability modeling setting with input space $\mathcal{X} = \mathbf{R}^d$ and outcome space $\mathcal{Y} = \{0, 1\}$. We want to predict the probability of the outcome 1 for any input $x \in \mathcal{X}$. The logistic regression model is $\mathbb{P}\left(Y = 1 \mid X = x; w\right) = \phi(w^T x)$, where $\phi(\eta) = 1/\left(1 + e^{-\eta}\right)$ (the standard logistic function). We fit $w \in \mathbf{R}^d$ by minimizing the negative log-likelihood (NLL) of w for some data $\mathcal{D} = \left((x_i, y_i)\right)_{i=1}^n$. The NLL is

$$\text{NLL}(w) \quad = \quad -\left[\sum_{i=1}^n y_i \log \phi(w^T x_i) + (1 - y_i) \log\left(1 - \phi(w^T x_i)\right)\right]$$

In Appendix A we show that

$$\nabla_w \text{NLL}(w) = \sum_{i=1}^n \left(\phi(w^T x_i) - y_i\right) x_i,$$

and in Appendix B we show that the objective function $\text{NLL}(w)$ is convex. So we'll get close to the global minimizer by gradient descent, and the minimizer would have $\nabla_w \text{NLL}(w) = 0$. Note that this is a vector equation, with one entry for each of d features. Let's consider one entry at a time and write x_i^j for the j'th feature in the i'th example. Then we can write the optimality conditions for w as

$$\sum_{i=1}^n \phi(w^T x_i) x_i^j \quad = \quad \sum_{i=1}^n y_i x_i^j \qquad \forall j \in \{1, \ldots, d\}\,.$$

As a simplest possible example, let's suppose that $x_i^j \equiv 1$ for all i. This corresponds to putting an intercept into the logistic regression model. In this case, the optimality condition becomes

$$\sum_{i=1}^{n} \phi(w^T x_i) \quad = \quad \sum_{i=1}^{n} y_i.$$

Note that the LHS is the expected number of x's that have $y = 1$ in the training set, where the expectation is w.r.t. the distribution given by our logistic regression model. The RHS is the actual number of examples with $y = 1$. Thus if we apply a fitted logistic regression model to its own training data, and add up the predicted probabilities of $y = 1$, we get the actual number positive examples in the training set.

For another example, suppose $x_i^j = 1(i$ has red hair$)$ and suppose that $y_i = 1$ means that i likes ice cream. Then on the RHS, each summand $y_i x_i^j$ is 1 if individual i has red hair AND likes ice cream, and 0 otherwise. On the LHS, $\phi(w^T x_i) x_i^j$ is 0 if individual i does not have red hair (because x_i^j is 0) and otherwise it's the predicted probability that the [red-haired] individual likes ice cream. So then the LHS is the expected number of individuals in the training set that have red hair and like ice cream, where the expectation is w.r.t. the model. The RHS is the number of individuals in the training set who have red hair and who like ice cream.

For the optimal w, these counts and expected counts must be equal for all d features.

2 Multinomial logistic regression with compatibility feature functions

The multinomial logistic regression model is a special case of the following probability model

$$p(y|x; w) = \frac{\exp(\sum_r w_r g_r(x, y))}{\sum_{y'} \exp(\sum_r w_r g_r(y', x))},$$

where $(x, y) \in \mathcal{X} \times \mathcal{Y}$ ($\mathcal{X}$ arbitrary, $\mathcal{Y}$ finite), the $g_i : (x, y) \mapsto \mathbf{R}$ for $i = 1, \ldots, d$ are the "compatibility features", or we can call all the g's together the "class-sensitive feature map" (depending on who you ask).

Exercise: How is multinomial logistic regression a special case of this? Answer: Let $\mathcal{X} = \mathbf{R}^D$ and let $\mathcal{Y} = \{1, \ldots, k\}$. For each $j \in \{1, \ldots, D\}$ and for each $c \in \mathcal{Y}$, create a compatibility feature $g_k(x, y) = x^j 1(y = c)$, where x^j is the jth component of the original feature vector $x \in \mathbf{R}^D$. So in the end there will be kD compatibility features, and $w \in \mathbf{R}^d$ where $d = kD$.

We'll find w by maximum likelihood. The log-likelihood of w for a dataset $(x_1, y_1), \ldots, (x_n, y_n)$ is

$$\begin{aligned} L(w) &= \sum_{i=1}^{n} \log p(y_i | x_i; w) \\ &= \sum_{i=1}^{n} \left(\sum_{\text{ftr } r} w_r g_r(x_i, y_i) - \log \left[\sum_{\text{label } y} \exp \left(\sum_{\text{ftr } r} w_r g_r(x_i, y) \right) \right] \right) \end{aligned}$$

Let's compute the partials:

$$\begin{aligned} \frac{\partial}{\partial w_r} L(w) &= \sum_{i=1}^{n} \left(g_r(x_i, y_i) - \frac{\sum_{\text{label } y} g_r(x_i, y) \exp\left(\sum_{\text{ftr } r} w_r g_r(x_i, y)\right)}{\sum_{\text{label } y} \exp\left(\sum_{\text{ftr } r} w_r g_r(x_i, y)\right)} \right) \\ &= \sum_{i=1}^{n} \left(g_r(x_i, y_i) - \sum_{\text{label } y} g_r(x_i, y) p(y \mid x_i; w) \right) \\ &= \sum_{i=1}^{n} \left(g_r(x_i, y_i) - \mathbb{E}\left[g_r(x_i, Y) \mid X = x_i; w \right] \right), \end{aligned}$$

where the last expectation is over the distribution for Y predicted by our model given input x_i and parameter vector w.

Note that the first term of $L(w)$ is linear in w and the second term is the negative of the log-sum-exp of linear functions of w, so the whole thing is concave (cf. Boyd&Vandenberghe Example 3.14, p. 87). That means we should be able to get close to the global minimum with gradient descent. So let's examine what happens at the solution to the first order conditions, since that's what we'll have at the optimum:

$$\begin{aligned} \frac{\partial}{\partial w_r} L(w) &= 0 \\ \iff \frac{1}{n} \sum_{i=1}^{n} g_r(x_i, y_i) &= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}\left[g_r(x_i, Y) \mid X = x_i; w \right] \end{aligned}$$

So if $g_r(x, y) = 1(x \text{ has red hair}) 1(y = \text{ likes ice cream})$, then this condition is telling us that the fraction of the sample that has red hair and likes ice cream is equal to the expected fraction of sample that has red hair and likes ice cream, as predicted by model. More generally, this is telling us that in fitting the model, we are attempting to match, for each compatibility feature g_r, the empirical average of g_r over the data with the expected average, where the expectation is based on the label distributions predicted by the model.

We might call this "moment matching", because a [generalized] moment is the expectation of some function of your random variables. In this case, the function is $g_r(x_i, Y)$.

A Gradient of NLL for logistic regression

We'll now compute the gradient of the NLL. It's helpful to first note that for $\phi(\eta) = 1/\left(1 + e^{-\eta}\right)$ we have $\phi'(\eta) = \phi(\eta)(1 - \phi(\eta)$.

$$\begin{aligned}
\nabla_w \mathrm{NLL}(w) &= \sum_{i=1}^{n} \left[-y_i \nabla_w \log \phi(w^T x_i)\right] + (y_i - 1)\, \nabla_w \log\left(1 - \phi(w^T x_i)\right) \\
&= \sum_{i=1}^{n} \left[-y_i \frac{\phi'(w^T x_i) x_i}{\phi(w^T x_i)}\right] - (y_i - 1)\, \frac{\phi'(w^T x_i) x_i}{1 - \phi(w^T x_i)} \\
&= \sum_{i=1}^{n} \left[-y_i \frac{\phi(w^T x_i)\left[1 - \phi(w^T x_i)\right] x_i}{\phi(w^T x_i)}\right] - (y_i - 1)\, \frac{\phi(w^T x_i)\left[1 - \phi(w^T x_i)\right] x_i}{1 - \phi(w^T x_i)} \\
&= \sum_{i=1}^{n} \left[-y_i \left[1 - \phi(w^T x_i)\right] x_i\right] - (y_i - 1)\, \phi(w^T x_i) x_i \\
&= \sum_{i=1}^{n} \left(-y_i x_i + y_i \phi(w^T x_i) x_i - y_i \phi(w^T x_i) x_i + \phi(w^T x_i) x_i\right) \\
&= \sum_{i=1}^{n} \left(-y_i x_i + \phi(w^T x_i) x_i\right) \\
&= \sum_{i=1}^{n} \left(\phi(w^T x_i) - y_i\right) x_i
\end{aligned}$$

B NLL for logistic regression is convex

$$\nabla_w^2 \mathrm{NLL}(w) = \sum_{i=1}^{n} \phi(w^T x_i) \left(1 - \phi(w^T x_i)\right) x_i x_i^T$$

And for any $z \in \mathbf{R}^d$, we have

$$\begin{aligned} z^T \left[\nabla_w^2 \text{NLL}(w)\right] z &= \sum_{i=1}^{n} \phi(w^T x_i) \left(1 - \phi(w^T x_i)\right) z^T x_i x_i^T z \\ &= \sum_{i=1}^{n} \phi(w^T x_i) \left(1 - \phi(w^T x_i)\right) \left(x_i^T z\right)^2 \\ &\geq 0. \end{aligned}$$