Multinomial logistic regression

In such cases we use multinomial logistic regression, also called softmax regression. n (or, historically, the maxent classifier). In multinomial logistic regression the target y is a variable that ranges over more than two classes; we want to know the probability of y being in each potential class c ∈ C, p(y = c|x).

The multinomial logistic classifier uses a generalization of the sigmoid, called the softmax function, to compute the probability p(y = c|x). The softmax function takes a vector z = [z1,z2,...,zk] of k arbitrary values and maps them to a probability distribution, with each value in the range (0,1), and all the values summing to 1. Like the sigmoid, it is an exponential function. For a vector z of dimensionality k, the softmax is defined as:

$$\text{softmax}(z_i) \;=\; \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad 1 \leq i \leq k$$

The softmax of an input vector z = [z1,z2,...,zk] is thus a vector itself:

$$\text{softmax}(z) \;=\; \left[\frac{e^{z_1}}{\sum_{i=1}^{k} e^{z_i}}, \frac{e^{z_2}}{\sum_{i=1}^{k} e^{z_i}}, \ldots, \frac{e^{z_k}}{\sum_{i=1}^{k} e^{z_i}}\right]$$

$\sum_{i=1}^{k} e^{z_i}$ is used to normalize all the values into probabilities. Thus for

The denominator example
given a vector:

z = [0.6,1.1,−1.5,1.2,3.2,−1.1]

the result softmax(z) is

[0.055,0.090,0.0067,0.10,0.74,0.010]

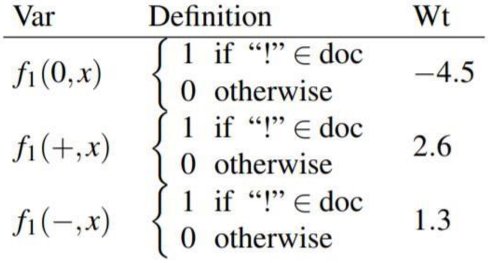
Again like the sigmoid, the input to the softmax will be the dot product between a weight vector w and an input vector x (plus a bias). But now we’ll need separate weight vectors (and bias) for each of the K classes.

$$p(y=c|x) = \frac{e^{w_c \cdot x + b_c}}{\sum_{j=1}^{k} e^{w_j \cdot x + b_j}}$$

Like the sigmoid, the softmax has the property of squashing values toward 0 or 1. Thus if one of the inputs is larger than the others, it will tend to push its probability toward 1, and suppress the probabilities of the smaller inputs.

Features in Multinomial Logistic Regression

For multiclass classification the input features need to be a function of both the observation x and the candidate output class c. Thus instead of the notation xi , fi or fi(x), when we're discussing features we will use the notation fi(c, x), meaning feature i for a particular class c for a given observation x. In binary classification, a positive weight on a feature pointed toward y=1 and a negative weight toward y=0, but in multiclass classification a feature could be evidence for or against an individual class. Let's look at some sample features for a few NLP tasks to help understand this perhaps unintuitive use of features that are functions of both the observation x and the class c. Suppose we are doing text classification, and instead of binary classification our task is to assign one of the 3 classes +, −, or 0 (neutral) to a document. Now a feature related to exclamation marks might have a negative weight for 0 documents, and a positive weight for + or − documents:



Var	Definition	Wt
$f_1(0,x)$	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	-4.5
$f_1(+,x)$	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	2.6
$f_1(-,x)$	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1.3

Learning in Multinomial Logistic Regression

Multinomial logistic regression has a slightly different loss function than binary logistic regression because it uses the softmax rather than the sigmoid classifier. The loss function for a single example x is the sum of the logs of the K output classes:

$$
\begin{aligned}
L_{CE}(\hat{y}, y) &= -\sum_{k=1}^{K} 1\{y = k\} \log p(y = k|x) \\
&= -\sum_{k=1}^{K} 1\{y = k\} \log \frac{e^{w_k \cdot x + b_k}}{\sum_{j=1}^{K} e^{w_j \cdot x + b_j}}
\end{aligned}
$$

This makes use of the function 1{} which evaluates to 1 if the condition in the brackets is true and to 0 otherwise. The gradient for a single example turns out to be very similar to the gradient for logistic regression, although we don’t show the derivation here. It is the difference between the value for the true class k (which is 1) and the probability the classifier outputs for class k, weighted by the value of the input xk :

$$
\begin{aligned}
\frac{\partial L_{CE}}{\partial w_k} &= -(1\{y = k\} - p(y = k|x))x_k \\
&= -\left(1\{y = k\} - \frac{e^{w_k \cdot x + b_k}}{\sum_{j=1}^{K} e^{w_j \cdot x + b_j}}\right) x_k
\end{aligned}
$$

What is Simple Logistic Regression?

Use simple logistic regression when you have one nominal variable with two values (male/female, dead/alive, etc.) and one measurement variable. The nominal variable is the dependent variable, and the measurement variable is the independent variable.

What is Ordinal Logistic Regression?

Ordinal regression is used to predict the dependent variable with ‘ordered’ multiple categories and independent variables. In other words, it is used to facilitate the interaction of dependent variables (having multiple ordered levels) with one or more independent variables.