**Regression**

**Multivariable Logistic Regression**

Seems that now we are just generalizing our curve to include multiple independent variables **x**i = (x1i, x2i, x3i, etc.).

Chart, line chart

Description automatically generated

And our curve would generalize to, for instance:



The best fit curve can be ascertained by adjusting the parameters to maximize the cumulative likelihood:



or minimizing the log-loss,



And like before,

Looks like we can take LLf to be analogous to the SSEf from linear models. It will be zero when there is perfect fit, and ∞ for really bad fit.

As suggested in the examples in the previous file, if we can physically separate all the no’s and yes’s into separate volumes of phase space, then we ought to be able to perfectly fit the data. In this cae LLf = 0. On the other hand, the ‘worst’ fit scenario should still be when we just approximate it with a hyperplane, setting all the m’s = 0. Then we have f(x) = 1/(1+e-(0·x\_1 + 0·x\_2 + 0·x\_3 + b) = p. Should find again that p = n1/(n0 + n1) where n0,1 = total number of no’s, yes’s. And LLf will be in this case, like before,



Okay. This is just some large positive number.

**Goodness of Fit: R2 value**

R2 is a measure of the goodness of fit, of the regression curve. There’s apparently no consensus on a definition of R2. But one analogous to the linear models expression is this. We define



where LLf is the log-loss of the logistic regression curve, and LLm is the log-loss of the curve that just fits the ‘mean’ of the data, i.e., it is f(**x**) = p = n1/(n0 + n1), where n1 is the total number of yi = 1’s and n0 = the total number of yi = 0’s. As we argued in the example above, LLm forms the upper bound of the log-loss for any logistic function. So we should always have LLf < LLm. And so 0 < R2 < 1.

**Hypothesis Testing**

Now that we are allowing multiple fitting parameters, it’s useful to know how to tell if the fitting parameters are predictive – if they are fitting the behavior, or just the noise. In analogy with the multivariable linear regression stuff, let’s say we have a model with f d.o.f./fitting parameters, which is taken as the standard model so-to-speak. Then let our Null Hypothesis be as follows:

H0 = assumption that the data is described by model f(**x**) which has f0 degrees of freedom.

And let LLf0 be the log-loss for this model.Then let’s compare to another model with f > f0 degrees of freedom. And let LLf be its log-loss. We would anticipate this to be smaller of course, i.e., LLf0 > LLf. The alternative hypothesis would be:

HA = assumption that at least one of the extra f – f0 parameters in the new model is non-zero.

Given the Null Hypothesis, turns out LLf follows a known probability distribution. Well we can form a test statistic,



And this follows a χ2 distribution with f – f0 d.o.f..



which is the probability density of getting an Z-value of x, given the null hypothesis is true. So we can calculate a p-value,



which would be the probability that we’d get an Z-statistic Z\* or higher, out of the new model, if the Null hypothesis were true. So if the f model has true explanatory power, then we should find Z\* >> 0 and the p-value should be small (less than 0.05 at the 95% significance level).

**Appendix 1**

We should be able to easily generalize this discussion to include categarical variables which can take on discrete values, like we did for linear models. We would just generalize our f(x) to something like,



and A, B, C can be 0/1.

**Appendix 2**

Since f(x1, x2, x3) → probabilities. Can we say f(x1,x2,x3) = P(y|x1x2x3)? Yes I think so. Then what is P(x1,x2,x3,y)? Well,



and I guess P(x1,x2,x3) would just be 1/n, where n = # data points, presuming there are no duplicate xi’s. So then we could say,

