CSC 411, Fall 2006 — Assignment #1

*Due at **start** of tutorial time on October 13. Note that this assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own. Handing in work that is not your own is a serious academic offense. Fabricating results, such handing in fake output that was not actually produced by your program, is also an academic offense.*

In this assignment you will implement maximum penalized likelihood binary logistic regression using a quadratic penalty, and a variant of binary logistic regression which avoids probabilities that are close to 0 or 1. You will also assess how well these methods do compared to ordinary maximum likelihood logistic regression on two artificial datasets that are provided on the web page.

The web page also contains R and Matlab implementations of maximum likelihood binary logistic regression, done using the standard builtin minimization functions (`nlm` for R, `fminunc` for Matlab). You may use these programs as a basis for your implementation.

Recall that in binary logistic regression the probability of a binary response being 1 is modelled as

$$P(y = 1 \,|\, x_1, \ldots, x_p) \;\; = \;\; \left[1 \;+\; \exp\left(- \left(\beta_0 + \sum_{j=1}^{p} \beta_j \, x_j \right) \right) \right]^{-1}$$

The log likelihood for β given n training cases can be written as follows:

$$\ell(\beta) \;\; = \;\; \sum_{i=1}^{n} - \log \left[1 \;+\; \exp\left(-(2y^{(i)} - 1) \left(\beta_0 + \sum_{j=1}^{p} \beta_j \, x_j^{(i)} \right) \right) \right]$$

Here, $2y - 1$ converts the 0/1 response to $-1/+1$. The maximum likelihood estimate for β is the one that maximizes $\ell(\beta)$.

You should implement maximum penalized likelihood estimation for this model, in which the estimate for β maximizes

$$\ell(\beta) \;-\; \lambda \sum_{j=1}^{p} \beta_j^2$$

The value of λ needs to be set somehow. Later, we'll talk about ways of doing this, but in this assignment you should just investigate the effects of different values for λ.

You should also implement a modified form of logistic regression in which the probability of class 1 is modelled as

$$P(y = 1 \,|\, x_1, \ldots, x_p) \;\; = \;\; \nu_1 + (1 - \nu_0 - \nu_1) \left[1 \;+\; \exp\left(- \left(\beta_0 + \sum_{j=1}^{p} \beta_j \, x_j \right) \right) \right]^{-1}$$

In this model, the probability of class 1 is always at least ν_1, and the probability of class 0 is always at least ν_0. Here, ν_0 and ν_1 are additional model parameters that must be estimated from the training data. To avoid the need to constrain these parameters, you should instead use unconstrained parameters θ_0 and θ_1, related to ν_0 and ν_1 by

$$\begin{aligned} \nu_0 &= (1/2) \;/ (1 + \exp(-\theta_0)) \\ \nu_1 &= (1/2) \;/ (1 + \exp(-\theta_1)) \end{aligned}$$

You should also implement the combination of these two modifications — the modified form of the probability with a penalty on $\beta_1, \ldots, \beta_p$. This gives four methods to compare — ordinary logistic regression and the three methods you implement.

You should compare these four methods on two artificial data sets that I created, which are available from the course web page, and also in `/u/radford/411/ass1` on CDF. The first data set has two inputs, 2000 training cases, and 3000 test cases. The second data set has 20 inputs, 100 training cases, and 900 test cases. In both cases, the classes for the test cases are provided so that you can see how good the predictions of the four methods are. (Of course, your programs should *not* look at the classes of test cases except when reporting how good the final results are.)

You should evaluate how good a method is in two ways — its error rate guessing the class by thresholding the probability of class 1 at 1/2, and its average squared error. The squared error for a test cases is defined to be the square of the difference between the class, coded as 0 or 1, and the probability of class 1. (One can prove that the average squared error is minimized when the probabilities produced by the model are the correct ones.) For the two methods with a penalty, you should try various values of λ, and in particular try to find the value of λ that gives the best results.

You should also report the log likelihood obtained with the estimates from the four methods. The likelihood is not a criterion for a method being good, but does give some insight into how it is behaving.

You should try various initial values for the optimization function, to see whether that makes a difference.

You should hand in a paper listing of your functions, written in R, Matlab, Octave, or some other suitable language (if you check with me first). Your program should be written in a readable style, and be reasonably efficient (though there is no need to be fanatical in this regard).

You should also hand in the results of the tests you did on the data sets. In particular, so that the marker can check correctness, you *must* hand in output giving the probability of class 1 for the first ten test cases of the first data set, for each of the four methods, using $\lambda = 1$ for the ones with a penalty. What other output to hand in is left for you to judge. You should hand in enough to justify your conclusions, but not an excessive amount.

Finally, you should hand in a discussion of your results, and what you think can (or cannot) be concluded from them.