STA 410/2102, Fall 2015 — Assignment #1

Due at the start of class on October 22. Please hand it in on 8 1/2 by 11 inch paper, stapled in the upper left, with no other packaging.

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. In particular, you should not leave any discussion with someone else with any written notes (either paper or electronic).

Suppose you are interested in what proportion of adults in Toronto play Minecraft regularly. You send out surveys to 130 Toronto adults selected uniformly at random from all adults in Toronto. Amazingly, you get responses from everyone sent a survey. (In real surveys, non-response is of course a big problem, but we're ignoring it here.) Of the $n = 130$ people surveyed, $x = 75$ say they play Minecraft.

At this point, the maximum likelihood estimate for the proportion of adult Torontonians who play Minecraft is easily obtained as $x/n = 75/130$. Unfortunately, you now realize that this isn't all you are interested in. You'd really like to know the proportion of men who play Minecraft and the proportion of women who play Minecraft, but you do not know the genders of the respondents.

To address this, you send out more surveys, to 25 randomly selected Toronto men and to 25 randomly selected Toronto women. (Your money is running out, so you can't afford bigger samples). Fortunately, your good fortune in having everyone respond continues, so non-response is also not a problem in these surveys. Of the $m_1 = 25$ men surveyed, $x_1 = 20$ say they play Minecraft. Of the $m_2 = 25$ women surveyed, $x_2 = 6$ say they play Minecraft.

You would now like to find the maximum likelihood estimates for the proportion, p_1, of men who play Minecraft and the proportion, p_2, of women who play Minecraft, based on all the data you have. You assume that the population of Toronto is equally divided into men and women, and that whether one person plays Minecraft is independent of whether another plays Minecraft.

The likelihood function is based on x having the binomial distribution with parameters n and $(p_1 + p_2)/2$, on x_1 having the binomial distribution with parameters m_1 and p_1, and on x_2 having the binomial distribution with parameters m_2 and p_2. So we can write the likelihood function as follows:

$$\begin{aligned} \ell(p_1, p_2) &= \binom{n}{x} \left((p_1 + p_2)/2 \right)^x \left(1 - (p_1 + p_2)/2\right)^{n-x} \\ &\quad \times \binom{m_1}{x_1} p_1^{x_1} (1 - p_1)^{m_1 - x_1} \times \binom{m_2}{x_2} p_2^{x_2} (1 - p_2)^{m_2 - x_2} \end{aligned}$$

Your task is to write a script and associated functions to find the maximum likelihood estimates for p_1 and p_2 using several methods, and compare how well they work. All the methods should work by maximizing the log of the likelihood, not the likelihood itself. The methods you should try are as follows:

- Alternating maximization (non-linear Gauss-Siedel iteration), using bisection to alternately find the maximum with respect to p_1 and with respect to p_2.
- Multivariate Newton iteration.
- Multivariate method of scoring.
- Just using R's built-in `nlm` function, passing it a function that computes the log-likelihood (but not its derivatives).

Your implementations of these methods should be designed to work for any data set (ie, any valid values of n, m_1, m_2, x, x_1, and x_2), not just the particular data mentioned above, though that is the data you will use when presenting results.

To implement these methods, you will need to write several functions involving the log likelihood for this model. They should have names and parameters as follows, where `p` is a vector containing the parameters p_1 and p_2, and `n`, `m1`, `m2`, `x`, `x1`, and `x2` are scalars corresponding to the other quantities mentioned above:

```
log_likelihood <- function (p, n, m1, m2, x, x1, x2)

log_likelihood_gradient <- function (p, n, m1, m2, x, x1, x2)

log_likelihood_hessian <- function (p, n, m1, m2, x, x1, x2)

fisher_information <- function (p, n, m1, m2, x, x1, x2)
```

You should use these functions to implement four functions for finding the maximum likelihood estimate using the four methods mentioned above, as follows:

```
mle_alt <- function (n,m1,m2,x,x1,x2,initial)

mle_mvn <- function (n,m1,m2,x,x1,x2,initial,iters)

mle_mos <- function (n,m1,m2,x,x1,x2,initial,iters)

mle_nlm <- function (n,m1,m2,x,x1,x2,initial)
```

where `initial` is an initial guess for the parameters.

For `mle_alt`, you should use the `bisect2` function from the Week 2 examples on the course web page, which continues until the zero has been found to the maximum possible accuracy. (You may want to comment out the call of the `cat` function that prints trace information.) You should keep iterating your alternating maximization procedure until the estimate is no longer changing.

For `mle_mvn` and `mle_mos`, you should use the `mvnewton` function from the Week 3 examples on the course web page. The `iters` argument for `mle_mvn` and `mle_mos` specifies how many iterations to do. You should set this argument to a value large enough that the final estimate is not changing, or is just flipping among a small number of nearby values.

For `mle_nlm`, you should let `nlm` decide when to stop, using its default criterion.

Your functions (except for `mle_nlm`) should print the estimates after each iteration, so that you can see how rapidly the method is converging. You should use `options(digits=17)` to see these estimates (and other quantities) to the full precision of the floating-point numbers used by R.

You should try out these functions on the data described above, and discuss how well they worked. In particular, you should discuss at least the following:

- How sensitive the methods are to the initial guess.
- Whether the methods produce the same answer (with a suitable initial guess), and if not, which answer is better (ie, comes closer to maximizing the log likelihood).
- How rapidly the methods (except `mle_nlm`) converge, and in particular whether convergence is linear, quadratic, or of some other form.
- How easy it was to implement the methods.

You should hand in a listing of your R functions and script, which should be in two files. One file should contain the definitions of with the functions of general use (the ones listed above), which someone might use to find estimates for some other data set of this form. It should contain `source` commands to read in the `bisect.r` and `mvnewton.r` files from the course web page. The other file should contain the script you used to apply these functions to the particular data set described above, and to produce any output needed for your discussion. It should contain a `source` command to read in your file of function definitions.

You should hand in your derivation of the log likelihood function and its gradient vector and Hessian matrix (which may be hand-written).

You should also hand in the output of the script you ran, including the estimates found, as well as other text or graphical output that helps support your discussion.

Finally, you should hand in your discussion, which may also be hand-written (though it's probably easier to type it in, particularly if you want to include numeric output from running your script).

Extra question for grad students in STA 2102 (bonus for STA 410 students):

Use the Hessian matrix computed at the maximum likelihood estimate to obtain standard errors for p_1 and p_2. Note in this regard that statistical theory suggests that the inverse of minus the Hessian of the log likelihood is an estimate of the covariance matrix of the sampling distribution of the estimate, or alternatively, of the covariance matrix of the posterior distribution (if the prior information is not strong). Also find standard errors using the Fisher information matrix, and compare them to those found using the Hessian (observed information). Do they differ much for the data used here? Might they differ more for other data?

[For grad students, this part is worth 10 marks and the main part is worth 90 marks. For undergrads, this part is worth a bonus of up to 10 marks, and the main part is worth 100 marks.]