Exercises 2: Generalized linear models

Exponential families

We say that a distribution $f(y;\theta,\phi)$ is in an exponential family if we can write its PDF or PMF in the form

$$f(y;\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y;\phi)\right\}$$

for some known functions a, b and c. We refer to θ as the canonical parameter of the family, and (for reasons that will become clear) to ϕ as the dispersion parameter.

(A) Starting from the "standard" form of each PDF/PMF, show that the following distributions are in an exponential family, and find the corresponding b, c, θ, and $a(\phi)$.

- $Y \sim N(\mu, \sigma^2)$ for known σ^2.
- $Y = Z/N$ where $Z \sim \text{Binom}(N, P)$ for known N.
- $Y \sim \text{Poisson}(\lambda)$

(B) We want to characterize the mean and variance of a distribution in the exponential family. To do this, we'll take an unfamiliar route, involving a preliminary lemma (that holds much more generally than just the exponential family). Define the *score* $s(\theta)$ as the gradient of the log likelihood with respect to the parameter of interest:

$$s(\theta) = \frac{\partial}{\partial\theta}\log L(\theta)\,, \quad L(\theta) = \sum_{i=1}^{n} f(y_i;\theta)\,.$$

We've written this in multivariate form for the sake of generality, but of course it just involves an ordinary partial derivative (w.r.t. θ) in the case where θ is one-dimensional.

While we think of the score as a function of θ, clearly (just like the likelihood) the score also depends on the data. So a natural question is: what can we say about the *distribution* of the score over different random realizations of the data under the true data-generating process, i.e. at the true θ? It turns out we can say the following, sometimes referred to as the score equations:

$$E\{s(\theta)\} = 0$$
$$\mathcal{I}(\theta) \equiv \text{var}\{s(\theta)\} = -E\left\{H(\theta)\right\}$$

where the mean and variance are taken under the true θ. **Prove the score equations.** Hints: prove the first equation first. You can assume that it's OK to switch the order of differentiation and integration (i.e. that any necessary technical conditions are met). To prove the second equation, differentiate both sides of the first equation with respect to θ^T and switch the order of differentiation and integration again. Expand out and simplify.

(C) Use the score equations you just proved to show that, if $Y \sim f(y;\theta,\phi)$ is in an exponential family, then

$$E(Y) = b'(\theta)$$
$$\text{var}(Y) = a(\phi)b''(\theta)$$

Thus the variance of Y is a product of two terms. One of these terms, $b''(\theta)$, depends only on the canonical parameter θ, and hence on the mean, since you showed that $E(Y) = b'(\theta)$. The other, $a(\phi)$, is independent of θ. Note that the most common form of a is $a(\phi) = \phi/w$, where ϕ is called a dispersion parameter and where w is a known prior weight that can vary from one observation to another; we'll see this below.

(D) To convince yourself that your result in (C) is correct, use these results to compute the mean and variance of the $N(\mu,\sigma^2)$ distribution.

Generalized linear models

Suppose we observe data like in the typical regression setting: that is, pairs $\{y_i, x_i\}$ where y_i is a scalar response for case i, and x_i is a p-vector of predictors or features for that same case i. We say that the y_i's follow a *generalized linear model* (GLM) if two conditions are met. First, the PDF (or PMF, if discrete) can be written as:

$$f(y_i;\theta_i,\phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi/w_i} + c(y_i;\phi/w_i)\right\}$$

where the weights w_i are all known. This is referred to as the stochastic or random component of the model. Second, for some known invertible function g we have

$$g(\mu_i) = x_i^T\beta$$

where $\mu_i = E(Y_i;\theta_i,\phi)$. This is the systematic component of the model, and g is referred to as a link function, since it links the mean of the response μ_i with the *linear predictor* $\eta_i = x_i^T\beta$.

(A) Deduce from your results above that, in a GLM,

$$\theta_i = (b')^{-1}\Big(g^{-1}(x_i^T\beta)\Big)$$
$$\text{var}(Y_i) = \frac{\phi}{w_i}V(\mu_i)$$

for some function V that you should specify in terms of the building blocks of the exponential family model. V is often referred to as a the *variance function*, since it explicitly relates the mean and the variance in a GLM.

(B) Take two special cases.

(1) Suppose that Y is a Poisson GLM, i.e. that the stochastic component of the model is a Poisson distribution. Show that $V(\mu) = \mu$.

(2) Suppose that $Y = Z/N$ is a Binomial GLM, i.e. that the stochastic component of the model is a Binomial distribution $Z \sim \text{Binom}(N, P)$ and that Y is the fraction of yes outcomes (1's). Show that $V(\mu) = \mu(1-\mu)$.

(C) To specify a GLM we must choose the link function $g(\mu_i)$. Recall that g links the predictors with the mean of the response: $g(\mu_i) = x_i^T\beta$. Since you've shown that

$$\theta_i = (b')^{-1}\left\{g^{-1}(x_i^T\beta)\right\},$$

a "simple" choice of link function is one where $g^{-1} = b'$, or equivalently $g(\mu) = (b')^{-1}(\mu)$. This is known as the *canonical link*, in which case the canonical parameter simplifies to

$$\theta_i = (b')^{-1}\left\{b'(x_i^T\beta)\right\} = x_i^T\beta\,.$$

So under the canonical link $g(\mu) = b'^{-1}(\mu)$, we have the model

$$f(y_i;\beta,\phi)\exp\left\{\frac{y_i x_i^T\beta - b(x_i^T\beta)}{\phi/w_i} + c(y_i;\phi/w_i)\right\}$$

Now return to the two special cases from the previous problem.

(1) Suppose that Y is a Poisson GLM, i.e. that the stochastic component of the model is a Poisson distribution. Show that the canonical link is the log link, $g(\mu) = \log\mu$.

(2) Suppose that $Y = Z/N$ is a Binomial GLM, i.e. that the stochastic component of the model is a Binomial distribution $Z \sim \text{Binom}(N, P)$. Show that the canonical link is the logistic link $g(\mu) = \log\{\mu/(1-\mu)\}$.

Fitting GLMs

The regression coefficients β in a GLM are typically fit using some variation on likelihood-based inference. To this end, define the likelihood function for a given GLM as

$$L(\beta, \phi) = \prod_{i=1}^{n} \exp\left\{ \frac{y_i \theta_i - b(\theta_i)}{\phi / w_i} + c(y_i; \phi / w_i) \right\},$$

where based on results you proved above, we define $\theta_i = (b')^{-1}(\mu_i)$ and $\mu_i = g^{-1}(x_i^T \beta)$. This allows us to define the score function $s(\beta, \phi)$ as the gradient of the log likelihood with respect to β:

$$s(\beta, \phi) = \nabla_\beta \log L(\beta, \phi) = \frac{\partial}{\partial \beta} \log L(\beta, \phi).$$

(A) Using the chain rule

$$\frac{\partial}{\partial \beta} = \frac{\partial}{\partial \theta} \times \frac{\partial \theta}{\partial \mu} \times \frac{\partial \mu}{\partial \beta},$$

show that

$$s(\beta, \phi) \equiv \nabla_\beta \log L(\beta, \phi) = \sum_{i=1}^{n} \frac{w_i (Y_i - \mu_i) x_i}{\phi V(\mu_i) g'(\mu_i)}$$

where x_i is the vector of predictors for case i (i.e. row i of the predictor matrix X, transposed to be a column vector).

(B) Show that under the canonical link, $g'(\mu) = 1/V(\mu)$, so that the score function simplifies to:

$$s(\beta, \phi) = \sum_{i=1}^{n} \frac{w_i (Y_i - \mu_i) x_i}{\phi}.$$

Hint: remember from calculus that

$$(g^{-1})'(x) = \frac{1}{g' \{g^{-1}(x)\}}$$

(C) Let's take the specific case of a GLM for a binomial outcome, where $Y_i \sim \text{Binom}(N_i, \mu_i)$ for known sample size N_i, $Y_i = Z_i / N_i$ is the observed success fraction, and where μ_i is related to the predictors $x_i \in \mathcal{R}^p$ via the (canonical) logistic link. This is called the logistic regression model. But how should we fit the parameters?

Read up on the method of steepest descent, i.e. gradient descent[1] Write your own function that will fit a logistic regression model by

[1] if you want a textbook reference, see *Numerical optimization*, by Nocedal and Wright. This should be available in electronic form through from the UT Library website.

gradient descent. For extra coding brownie points, try to maintain some level of generality to your code, i.e. so that it could also work with different GLMs, assuming your wrote different sub-routines.

Grab the data "wdbc.csv" from the course website, or obtain some other real data that interests you, and test out your fitter. The WDBC file has information on 569 breast-cancer patients from a study done in Wisconsin. The first column is a patient ID, the second column is a classification of a breast cell (Malignant or Benign), and the next 30 columns are measurements computed from a digitized image of the cell nucleus. These are things like radius, smoothness, etc. For this problem, use the first 10 features for X, i.e. columns 3-12 of the file. (If you use all 30 features you'll run into trouble.)

Some notes:

- We're trying to maximize the log likelihood function, but the convention in the optimization literature is to minimize things. No big deal; what we're doing is the same as *minimizing* the negative of the log likelihood.
- You need to add an intercept term, and the simplest way is to add a column of 1's as the first column of the feature matrix X. (If you've never seen this trick before, convince yourself why it makes sense.)
- Make sure that, at every iteration of gradient descent, you compute and store the current value of the log likelihood, so that you can track and plot the convergence of the algorithm.
- Be sensitive to the numerical consequences of an estimated success probability that is either very near 0, or very near 1.
- Finally, you can be as clever as you want about the gradient-descent step size. Small step sizes will be more robust but slower; larger step sizes can be faster but may overshoot and diverge; step sizes based on line search (Chapter 3 of Nocedal and Wright) are cool but involve some extra, optional work.

(D) Consider the Hessian matrix, i.e. the matrix of partial second derivatives of the log likelihood:

$$H(\beta, \phi) = \frac{\partial^2}{\partial \beta \partial \beta^T} \log L(\beta, \phi)$$

Give an expression for the Hessian matrix $H(\beta, \phi)$ of a GLM that is as simple as possible, ideally in matrix form. Note: to keep things a little more streamlined, please assume the canonical link function

here; it's the same idea, just with hairier algebra, under an arbitrary link function.

(E) Now consider a point $\beta_0 \in \mathcal{R}^P$, which serves as an intermediate guess for our vector of regression coefficients. Show that, for any GLM, the second-order Taylor approximation of $\log L(\beta, \phi)$, around the point β_0, can be expressed in the form

$$q(\beta; \beta_0) = -\frac{1}{2}(\tilde{y} - X\beta)^T W(\tilde{y} - X\beta) + c\,,$$

where $\tilde{y}$ is a vector of "working responses" and W is a diagonal matrix of "working weights," and c is a constant that doesn't involve β. Give explicit expressions for the diagonal elements W_{ii} and for $\tilde{y}$ (which will necessarily involve the point β_0, around which you're doing the expansion).[2] Again, we're assuming the canonical link to make the algebra a bit simpler.

[2] Remember the trick of completing the square, e.g. `https://davidrosenberg.github.io/mlcourse/Notes/completing-the-square.pdf`.

(F) Read up on Newton's method for optimizing smooth functions (e.g. in Nocedal and Wright, Chapter 2). Implement it for the logistic regression model model and test it out on the same data set you just used to test out gradient descent.[3] Note: while you could do line search, there is a "natural" step size of 1 in Newton's method. Verify that your solution replicates the β estimate you get when using a package solver, e.g. the glm function in R, up to minor numerical differences.

[3] Hey, cool! You should be able to use your own solver for weighted least squares that you wrote for the first set of problems.

(G) Standard asymptotic theory, which we won't go into here, implies that the maximum likelihood estimator is consistent and asymptotically normal around the true value β_0:

$$\hat{\beta} \sim N(\beta_0, I(\beta_0, \phi)^{-1})\,,$$

where $\mathcal{I}(\beta_0, \phi)$, called the *Fisher information matrix*, is the same $\mathcal{I}$ you met all the way back when you proved the score equations:

$$\mathcal{I}(\beta_0, \phi) \equiv \text{var}\{s(\beta_0, \phi)\} = -E\left\{H(\beta_0, \phi)\right\}\,.$$

The fact that Fisher information is the negative of the expected Hessian motivates the following idea: use the inverse of the negative Hessian matrix at the MLE to approximate the inverse Fisher information, i.e. the covariance matrix of the estimator. Happily, you get this Hessian matrix for free when fitting by Newton's method.

For your logistic regression on the WDBC data fit via Newton's method, compute the square root of each diagonal element of the inverse Hessian matrix, evaluated at the MLE.[4] Compare these to the standard errors you get when using a package solver, e.g. the glm function in R.

[4] These are your standard errors for each coefficient, i.e. the square root of the variance of each coefficient's (approximate) sampling distribution.