Non-Gaussian responses (week 3?)

31 Jan 2023

Table of contents

Non-Gaussian responses

- Why worry about it?
- Isn't least-squares good enough?
- **poll** (polleverywhere)

Some answers

- heteroscedasticity (Gauss-Markov only applies to homog. variance)
- still unbiased but no longer minimum variance
- maybe we shouldn't (e.g. **linear probability model** in econometrics)
    - adjust for heteroscedasticity with **robust/sandwich estimators** etc. (White):

      $$\hat{\mathbf{V}} = (\mathbf{X}^\top \mathbf{X})^{-1} (\mathbf{X}^\top \mathbf{G} \mathbf{X}) (\mathbf{X}^\top \mathbf{X})^{-1}$$

      where $\mathbf{G} = \text{Diag}(\hat{\varepsilon}_i^2)$ (contrast with $s^2 (\mathbf{X}^\top \mathbf{X})^{-1}$)
- if we have
- if we have **nonlinear** models, MLEs are no longer unbiased

Why not linear?

- actual nonlinear patterns (but can handle these by transformation/basis expansion)
- unrealistic predictions (e.g. probabilities outside of $[0, 1]$
- varying effects (e.g. effect of a 1-unit change in x on probability must differ depending on baseline probability)
- Why not transform? **poll** (polleverywhere)

Logistic regression (ESL § 4.4)

- Worst-case scenario (farthest from Gaussian)
- ESL starts with a *multinomial* model:

$$\log \left(\frac{\Pr(G=i|X=x)}{\Pr(G=K|X=x)} \right) = \beta_{i0} + \beta_i^\top x, \quad i \in 1 \ldots K-1$$

(and so $\Pr(G=K|X=x) = 1 / \left(1 + \sum_{i=1}^{K-1} \exp(\beta_{i0} + \beta_i^\top x)\right)$)

- independent of baseline/reparameterization
- log-likelihood $\sum \log p_i(x_i; \theta)$ where θ is the complete set of parameters

Log-likelihood

- for two categories, log-likelihood simplifies to

$$\begin{aligned} & \sum \left(y_i \beta^\top x_i - \log\left(1 + e^{\beta^\top x_i}\right)\right) \\ = & \sum \left(y_i \eta_i - \log\left(1 + \exp(\eta_i)\right)\right) \end{aligned}$$

- **weight matrix** $\mathbf{W} = \mathrm{Diag}(p(1-p))$
  - more generally, $\mathrm{Diag}(1/\mathrm{Var}(\mu))$
- score equation:
  - $\sum_i = 1^N x_i(y_i - p(x_i; \beta))$
  - Newton update is $\beta^* - \mathbf{H}^{-1}\mathbf{g}$
  - gradient: $\mathbf{X}^\top(\mathbf{y} - \mathbf{p})$
  - generally $\mathbf{X}^\top(\mathbf{y} - \mu) = \mathbf{X}^\top(\mathbf{y} - g^{-1}(\eta))$
- Hessian: $-\mathbf{X}^\top \mathbf{W} \mathbf{X}$
- solution is $(\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \mathbf{z}$
- where $\mathbf{z} = \mathbf{X}\beta_0 + \mathbf{W}^{-1}(\mathbf{y} - \mathbf{p})$ is the adjusted response

Newton step

- **iteratively reweighted least squares**
- solve

$$\mathbf{X}^\top \mathbf{W} \mathbf{X} \beta^* = \mathbf{X}^\top \mathbf{W} \mathbf{z}$$

- 4C03 notes
- 4C03 notes 2

Newton vs IRLS

- Newton vs *Fisher scoring* (expected value of the Hessian); equivalent for the *canonical link*
 (e.g. logistic for binary data, log for Poisson data
- link mostly important for interpretation
- can be disregarded (?) if we are going to handle nonlinearity by basis expansion
- convergence? (Mount 2012)

Mount, John. 2012. “How Robust Is Logistic Regression?” *Win Vector LLC*. https://win-vector.com/2012/08/23/how-robust-is-logistic-regression/.

Families

- Gaussian, Poisson, binomial (binary)
- May need to compute *scale/dispersion* parameter
  - for exponential families, calculate as $\sqrt{D/(n-p)}$ where D is the *deviance* (-2 log likelihood, equal to SSQ for Gaussian)
  - not exactly the MLE but good enough
- **over-dispersion**: quasi-likelihood
- more complex familes (negative binomial etc.) have an additional, non-collapsible parameters, need to estimate by MLE (or **profiling**)

Regularized versions

- lasso, ridge, or elasticnet
- score equations: $\mathbf{x}^{|}top(\mathbf{y}-\mathbf{p}) = \lambda \cdot \text{sign}(\beta_j)$ for **active** variables (non-zero coeffs)

revisiting ridge by data augmentation

- we want to minimize $||\mathbf{y}-\mathbf{X}\beta||_2^2 + \lambda||\beta||_2^2$
- the solution to the original regression equations was $\hat{\beta} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$
- Set

$$\mathbf{B} = \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I} \end{pmatrix}$$

- ridge regression should still be solvable by data expansion, i.e. in the IRLS loop use

$$\mathbf{B} = \begin{pmatrix} \mathbf{X}^\top\mathbf{W}\mathbf{X} \\ \sqrt{\lambda}\mathbf{I} \end{pmatrix}$$

  and

$$\mathbf{y}^* = (\mathbf{y} \quad 0)$$

- so that $\mathbf{B}^\top\mathbf{B} = \mathbf{X}^\top\mathbf{X}+\lambda I$ and the $\mathbf{X}^\top\mathbf{y}$ term is unchanged

ridge + IRLS

- recall that we need to iteratively solve

$$\mathbf{X}^\top \mathbf{W} \mathbf{X} \beta^* = \mathbf{X}^\top \mathbf{W} \mathbf{z}$$

- if we want to solve the **weighted** least-squares problem from IRLS, we would normally take the QR decomposition of $\mathbf{X}' = \mathbf{X}\sqrt{\mathbf{W}}$ (so that $\mathbf{X}^\top \mathbf{X} = \mathbf{X}^\top \mathbf{W} \mathbf{X}$)
- enhance this by adding $\sqrt{\lambda} I$ to X and zeros to $\mathbf{z}$ (no longer $\mathbf{y}$) as before
- ¿ try out enhanced GLM ?

proximal gradient descent/Newton

- solving the optimization problem for non-differentiable penalties
- previous solution (cyclic coordinate descent)
- simpler strategies (cyclic coordinate descent) may not work as well
- **proximal** gradient descent or **proximal** IRLS:
- like the pathwise coordinate descent solution from lasso:
- **solution**:

$$\tilde{\beta}_j(\lambda) \leftarrow S\left(\sum_{i=1}^{N} x_{ij}(y_i - \tilde{y}_i^{(j)}), \lambda\right)$$

- where $S(t, \lambda) = \text{sign}(t)(|t| - \lambda)$
- except that we can no longer jump straight to the correct solution.

proximal operator

- separate objective function into **smooth** part (likelihood/RSS/etc., **plus** ridge penalty) and **non-smooth** part (typically an L1 regularization term)
- **proximal operator**:

$$\text{argmin}_u \left(\underbrace{h(u)}_{nonconvexpart} + \frac{1}{2}||u - x||_2^2 \right)$$

- for $h = \lambda||\beta||_1$ (lasso penalty), we get the soft-threshold operator

$$\begin{cases} \beta_i - \lambda & \text{if } \beta_i < -\lambda \\ 0 & \text{if } -\lambda < \beta_i < \lambda \\ \beta_i + \lambda & \text{if } \beta_i < -\lambda \end{cases}$$

(from Ryan Tibshirani's notes on optimization)

proximal Newton (IRLS)

- take a Newton/IRLS step
- apply the prox operator to soft-threshold
- not going to get into the details! thresholding is more complex than the gradient descent rule (Lee, Sun, and Saunders 2014)
- need to solve

$$\text{argmin}_u \left(\underbrace{h(u)}_{\text{nonconvex part}} + \frac{1}{2}(u-x)^\top H(u-x) \right)$$

i.e. replace $||u-x||_2^2$ with a corresponding quadratic form
- ¿ not sure if the solution corresponds easily to soft-thresholding again?
- also need to be careful about backtracking if necessary
- i.e. taking a Newton step $-1\mathbf{g}$ is better than an uninformed gradient step $t\mathbf{g}$ (where t is the **learning rate**) but might overshoot
- i.e., **don't try this at home**

(Ryan Tibshirani again)

Lee, Jason D., Yuekai Sun, and Michael A. Saunders. 2014. "Proximal Newton-Type Methods for Minimizing Composite Functions." *SIAM Journal on Optimization* 24 (3): 1420–43. https://doi.org/10.1137/130921428.

more computational details

- from glmnet family docs
- using the *name* of the family ("poisson" etc.) uses hard-coded internal algorithms
  - faster (but scaling isn't too bad??)
  - less flexible (alternative families [links, variance functions])
  - slightly less robust (doesn't do backtracking)

sparse model matrices (side note)

- expanding factors (categorical variables) may make p gigantic
- each factor f_i with n_i levels will be expanded via treatment contrasts, so $\mathbf{X}$ will have (at least) $\sum_i (n_i - 1)$ columns
- `glmnet::makeX`, `Matrix::sparse.model.matrix()`