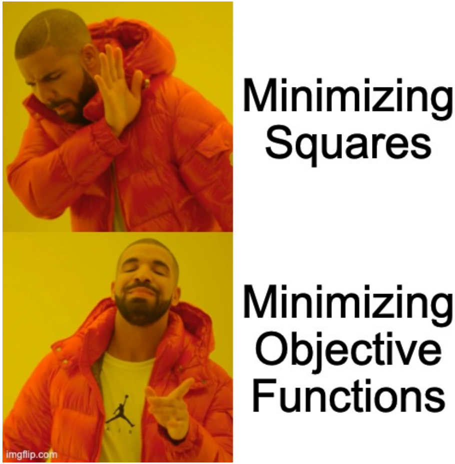
Likelihood Methods: Binary Discrete Choice, GLM and Computational Methods

Paul Goldsmith-Pinkham

March 2, 2021

Today’s topic: minimizing objection functions and an application

- Today: Two topics
    1. Minimizing objective functions instead of minimizing squares
    2. Studying binary choice model
- Minimizing objective functions: examples include minimizing squared distances, or maximizing likelihoods
- Most estimation issues can be framed as general objective function minimization problems
- Highlight this with example non-linear problems
    - Generalized linear models



Our setup

- Consider the following binary outcome problem: let Y_i denote if person i is a homeowner, and X_i includes three covariates: income, age and age^2 (plus a constant)

- A relatively general form of this relationship is

$$Y_i = F(X_i, \beta) + \epsilon_i$$

  In many ways, no different from our other estimation problems with linear regression!

- We can talk about an estimand for this setup based on assumptions on F and ϵ_i

Binary model – what's the right functional form?

- We could model this outcome using a linear regression – why not? Assume strong ignorability (or just $E(\epsilon_i|X_i) = 0$) and

$$E(Y_i|X_i) = X_i\beta \qquad \rightarrow Y_i = X_i\beta + \epsilon_i$$

- The canonical problem with this is twofold:
  1. The errors will be unusual – since it's binary, $V(Y|X) = X_i\beta(1 - X_i\beta)$, and you'll have pretty significant heteroskedasticity (this is obviously solveable using robust SE)
  2. Except under some special circumstances, it's very likely that the predicted values of Y_i will be outside of $[0, 1]$

- What's an example where they will not be? Discrete exhaustive regressors!
  - Why? No extrapolation. Extrapolation is what causes values outside support.

- How does this impact our causal estimates?
  - If the model is correctly specified, we can generate counterfactuals
  - If not, then we get a linear approximation

Linear Probability Model estimates on homeownership

- If income were strictly ignorable, we could say that 10k increase in income leads to 0.8 p.p. increase in the probability of homeownership

- Predicted values of homeownership are on support of $[0.283, 1.78]$
    - Oops.

variable	linear est.	std.error
Intercept	0.0242	0.0410
age	0.0220	0.0017
age^2	-0.0002	0.0000
income /10k	0.0069	0.0007

Modeling discrete choice

- There are two ways to think about how we think about this estimation problem. These are not mutually exclusive.

- The first is a statistical view. E.g. can we model the statistical process better (e.g. the counterfactual). One way to consider this is $X\beta$ is the conditional mean of some process – what is a statistical error term that fits with this?
  - Special case of what's termed "Generalized Linear Models" (GLM)
  - Will discuss in a bit

- A second way to view this is as an structural (economic) choice problem. Most models of limited dependent variables (e.g. binary) instead assume a latent index.

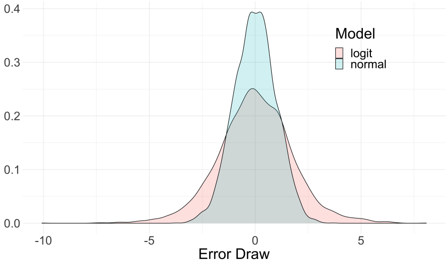
$$Y^* = X\beta + \epsilon, \qquad Y = \begin{cases} 1 & Y^* > 0 \\ 0 & Y^* \leq 0 \end{cases}$$

All about the epsilons

$$Y^* = X\beta + \epsilon, \qquad Y = \begin{cases} 1 & Y^* > 0 \\ 0 & Y^* \leq 0 \end{cases}$$

- A natural approach is to make a distributional assumption about ϵ to do estimation (and fix the support problem). Two common assumptions:
    1. ϵ is conditionally normally distributed (probit), such that $Pr(Y_i = 1|X_i) = \Phi(X_i\beta)$
    2. ϵ is conditionally extreme value (logistic) such that $Pr(Y_i = 1|X_i) = \frac{\exp(X_i\beta)}{1+\exp(X_i\beta)}$

- Note that these are not, in the binary setting, deeply substantive assumptions.
    - A challenge for probit models is that there's no closed form solution for Φ



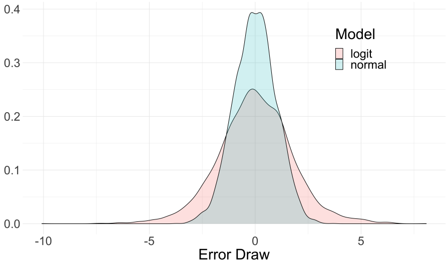
Identification up to scale

$$Y^* = X\beta + \epsilon,\ \Pr(Y_i = 1|X_i) = F(X_i\beta)$$

- Important caveat: these modes only identify β up to scale.

- Why? The “true” model of ϵ could have variance σ^2 that is unknown.

- Consider if $F(X_i\beta) = \Phi(X_i\beta)$. If this were a general normal (rather than standardized with variance 1), we could just scale up the coefficients proportinoate to σ and the realized binary outcome would identical. Hence, we normalize $\sigma = 1$ in most cases. This is *not* a meaningful assumption.



Comparing with Logit

- Consider now the same homeowner problem, but estimated with logit

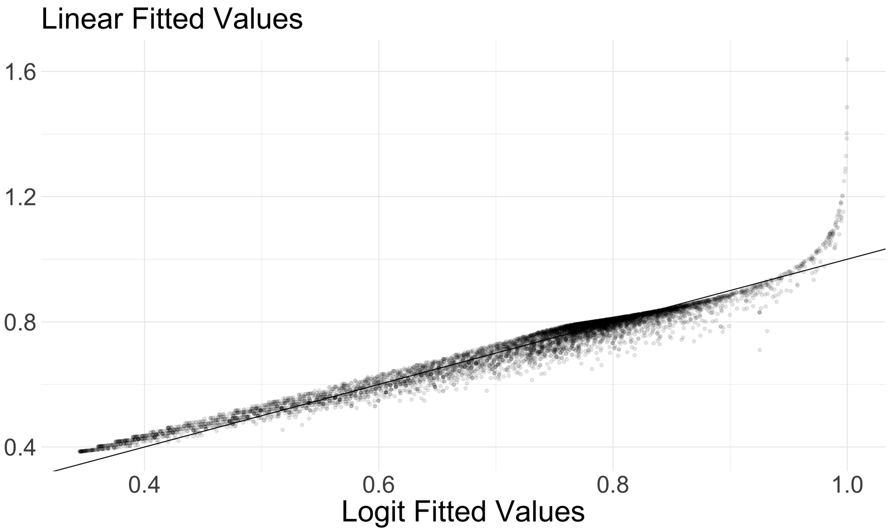
- These coefficients are harder to interpret – we can instead consider the average derivative:

$$n^{-1}\sum_i \frac{\partial E(Y|X)}{\partial X} =$$

$$n^{-1}\sum_i \frac{\exp(X_i\beta)}{1+\exp(X_i\beta)}\frac{\beta}{1+\exp(X_i\beta)}$$

- Avg. deriv is comparable but not identical

term	logit est.	linear est.	avg. deriv.
constant	-2.14	0.0242	-0.392
age	0.0903	0.022	0.0166
age^2	-0.0006	-0.0002	-0.0001
income/10k	0.0716	0.0069	0.0131


Linear Fitted Values
1.6
1.2
0.8
0.4
0.4
0.6
0.8
1.0
Logit Fitted Values

Aside: Generalized Linear Models

- Important aside: Generalized Linear Models (GLM)
  - General setting considering errors that are non-normal (and may have restricted support)
  - Very common terminology in non-economics settings

- Key pieces with a linear model $X_i\beta$:
  1. Link function g such that $E(Y|X) = g^{-1}(X_i\beta)$
  2. Error distribution drawn from exponential family (includes normals, binomial, Poisson)

- Some simple examples:
  - Logit (we just did this), with a link function $log\left(\frac{X_i\beta}{1-X_i\beta}\right)$
  - Normal (we just did this), with an identity link function

- In essence, we can enforce a linear functional form to the *mean*, and allow the error distribution to fit the form of the data
  - Important underused case: Poisson regression for non-negative numbers
  - Key point: even if model is “wrong”, can construct robust s.e. that are robust to the misspecification

How do we estimate these problems?

- How do we estimate these types of problems? Consider the likelihood function for logit:

$$
\begin{aligned}
Pr(Y_i = 1|X_i) &= \frac{\exp(X_i\beta)}{1+\exp(X_i\beta)} \\
l(\beta|\mathbf{Y},\mathbf{X}) &= \Pi_{i=1}^{n} Pr(Y_i = 1|X_i)^{Y_i}(1 - Pr(Y_i = 1|X_i)^{1-Y_i} \\
L(\beta|\mathbf{Y},\mathbf{X}) &= \sum_{i=1}^{n} Y_i \log(Pr(Y_i = 1|X_i)) \\
&\quad + (1 - Y_i)\log(1 - Pr(Y_i = 1|X_i))
\end{aligned}
$$

- Rule of thumb: the likelihood is the joint probability of the data
  - We are exploiting the independent nature of the data
  - Joint probability of two independent values is the product of their marginals

- Recall that we can take the log of the likelihood when considering extremes of the function because any maximum will be identical irrespective of monotone transformations

Plug in Logit to the ML

- With some simple rewriting:

$$
\begin{aligned}
L(\beta|\mathbf{Y},\mathbf{X}) &= \sum_{i=1}^{n} Y_i \log(\frac{\exp(X_i\beta)}{1+\exp(X_i\beta)} + (1-Y_i)\log(\frac{1}{1+\exp(X_i\beta)}) \\
&= \sum_{i=1}^{n} Y_i X_i\beta - Y_i \log(1+\exp(X_i\beta)) + (1-Y_i)\log(1+\exp(X_i\beta))) \\
&= \sum_{i=1}^{n} Y_i X_i\beta - \log(1+\exp(X_i\beta))
\end{aligned}
$$

- Great, so how would one estimate this? We have a likelihood, we want to maximize it!
- Take derivatives and find the maximum!
  - Finally that calculus is paying off!
- Good news and bad news...

The bad news and the good news

$$L(\beta|\mathbf{Y},\mathbf{X}) = \sum_{i=1}^{n} Y_i X_i \beta - \log(1 + \exp(X_i \beta))$$

- There's no analytic solution for this β. Unlike with OLS, we can't get a closed-form solution for our estimate – this is true of most estimators. In fact, this is a well-behaved problem, relative to most.
    - Well-behaved because it's globally concave and has easily calculated first and second derivative

- So, What's the good news? We have computers!

The bad news and the good news

$$\frac{\partial L(\beta)}{\partial \beta} = \sum_{i=1}^{n} X_i(Y_i - \frac{\exp(X_i\beta)}{1+\exp(X_i\beta)})$$

- While there is not an analytic solution, if there is a maximum where $\hat{\beta}$ satisfies $\frac{\partial L(\hat{\beta})}{\partial \beta} = 0$, then there are sets of conditions such that
  - $\lim_{n\to\infty} \Pr(||\hat{\beta}_n - \beta_0|| > \epsilon) = 0$ (weak consistency)
  - $\lim_{n\to\infty} \sqrt{n}(\hat{\beta}_n - \beta_0) \to^d \mathcal{N}\left(0, -E\left[\frac{\partial^2}{\partial\beta\beta'} L(\beta_0)\right]\right)$ (asymptotic normality)

- The challenge is that the conditions for when this is satisfied vary from problem to problem

- Most general results in this put high-level assumptions on the problem, and then the conditions need to be checked

The bad news and the good news

$$\frac{\partial L(\beta)}{\partial \beta} = \sum_{i=1}^{n} X_i (Y_i - \frac{\exp(X_i\beta)}{1+\exp(X_i\beta)})$$

- While there is not an analytic solution, if there is a maximum where $\hat{\beta}$ satisfies $\frac{\partial L(\hat{\beta})}{\partial \beta} = 0$, then there are sets of conditions such that
  - $\lim_{n\rightarrow\infty} \Pr(||\hat{\beta}_n - \beta_0|| > \epsilon) = 0$ (weak consistency)
  - $\lim_{n\rightarrow\infty} \sqrt{n}(\hat{\beta}_n - \beta_0) \rightarrow^d \mathcal{N}\left(0, -E\left[\frac{\partial^2}{\partial\beta\beta'}L(\beta_0)\right]\right)$ (asymptotic normality)

- The conditions for when this is satisfied vary from problem to problem

- Most general results in this put high-level assumptions on the problem, and then the conditions need to be checked for a particular problem
  - These general types of problems are classified into M-estimation and Z-estimation
  - M-estimation is a general problem where $\beta_0 = \arg\max_\beta E(m(\beta))$
  - Z-estimation $\subset$ M-estimation focused on exploiting features of the derivative of $m(\beta)$

How to compute - Newton-Raphson

- In our applications, very well-defined solutions. We'll instead focus on the actual computation of these maxima

- There are many numerical optimization methods. I'll outline info on the few I know, but this is in no way exhaustive
    - This draws from my own graduate school notes!

- A common simple method is Newton-Raphson

Newton-Raphson Computation of MLE

- Let $Q(\theta) = -L(\theta)$ (denote with θ to highlight that this is a general problem)

- Idea is to take some arbitrary objective function and fit a local quadratic based on derivatives
  - Find the minimum based on this quadratic
  - Take that minimizer and repeat

- Specifically, let

$$\theta_{k+1} = \theta_k - \left[\frac{\partial^2 Q(\theta_k)}{\partial\theta\partial\theta'}\right]^{-1} \frac{\partial Q}{\partial\theta}(\theta_k)$$

- In our Logit application, we already know the first derivative – calculating the second derivative is straightforward. Hence, we can solve for θ
  - We benefit from a convex problem and an easily defined second derivative

More general methods

- What if we don't know our second derivative? (or it is onerous to calculate)

- Then we can reframe to the problem in two pieces. Let A_k be any positive definite matrix. Consider the following iterated estimation:

$$\theta_{k+1} = \theta_k - \lambda_k A_k \frac{\partial Q}{\partial \theta}(\theta_k)$$

- This nests Newton-Raphson:
    - $\lambda_k = 1$
    - $A_k = \left[\frac{\partial^2}{\partial\theta\theta} L(\theta_k)\right]^{-1}$

- Intuitively, there are two pieces:
    - a steplength (defined by λ_k)
    - a direction $d_k = A_k \frac{\partial Q}{\partial \theta}(\theta_k)$ (controlled by A_k, which select a direction of the gradient)
        - A convenient rescaling is $\tilde{d}_k = d_k / (1 + \sqrt{d_k' d_k}$ to ensure $|\tilde{d}_k| < 1$

Simple version of algorithm

- We can choose the direction, then choose how far we want to go

$$\lambda_k = \arg\min_{\lambda} Q(\theta_k + \lambda \tilde{d}_k)$$

- Simplest version verison of this is $A_k = I_k$ (identity matrix) – just go in the direction of steepest descent

- How does one calculate λ_k in these settings? If θ is scalar, it's feasible (but inefficient) to calculate using a simple grid search

- In high-dimensions, too slow (and our next algorithm needs optimal choice to converge)

Two line search algorithms - Newton's Method

- Given a d, recall we need a λ. Redefine $\lambda^* = \arg\min_\lambda Q(\lambda)$
- The simplest method is Newton's method (which finds the root of a function (we want the root of the derivative)
- Begin with an initial guess for λ_0. Then,

  $$\lambda_{k+1} = \lambda_k - \frac{Q'(\lambda_k)}{Q''(\lambda_k)}$$

  Repeat till $|\lambda_{k+1} - \lambda_k|$ is small (e.g. convergence)
- Issue with this approach is you need a second derivative

Two line search algorithms - Golden Search

- Start with two points you know for certain contain the minimum (need unimodality)
    - E.g. $\lambda_l = 0, \lambda_h = 1$ [Picking an abritraily large λ_h is fine – there are ways to check this]

- Two points on the line segment between: $\lambda_{m1} = \lambda_l + 0.392 \times (\lambda_h - \lambda_l)$ and $\lambda_{m2} = \lambda_l + 0.618 \times (\lambda_h - \lambda_l)$

- Now, given the four points, can check two conditions:
    - $Q(\lambda_{m2}) > Q(\lambda_{m1})$: you know that the minimizing value of λ in $[\lambda_l, \lambda_{m2}]$. Update your values: $\lambda'_l = \lambda_l, \lambda'_h = \lambda_{m2}, \lambda'_{m2} = \lambda_{m1}, \lambda'_{m1} = \lambda'_l + (\lambda'_h - \lambda'_{m2}$
    - $Q(\lambda_{m2}) < Q(\lambda_{m1})$: you know that the minimizing value of λ in $[\lambda_{m1}, \lambda_h]$. Update your values: $\lambda'_h = \lambda_h, \lambda'_l = \lambda_{m1}, \lambda'_{m1} = \lambda_{m2}, \lambda'_{m2} = \lambda'_h - (\lambda'_{m1} - \lambda'_l$

- Update until you find the optimal λ

Davidson-Fletcher-Powell

- DFP is more elaborate, and requires all these pieces
  - Commonly used, although not the fastest algorithm out there now

- Its strongest feature is that it is efficient and can work without calculating a second derivative

- Initiate with any positive definite matrix A (e.g. identity matrix)

- Steps (repeat till convergence):
  1. Calculate direction $\tilde{d}_k$
  2. Calculate optimal step length λ_k
  3. Calculate the actual step $p_k = \lambda_k \tilde{d}_k$ and the new parameter $\theta_{k+1} = \theta_k + p_k$
  4. Calculate the change in the derivative q_k from θ_k to θ_{k+1}
  5. Update

$$A_{k+1} = A_k + \frac{p_k p_k'}{p_k' q_{k+1}} - \frac{A_k q_{k+1} q_{k+1}' A_k}{q_{k+1}' A_k q_{k+1}}$$