LASSO/Poisson DML implementation

Thomas R. Covert, Yixin Sun

July 20, 2020

1 Setup from Chernozhukov et al

Let θ be the thing we care about and β be the nuisance parameters (location, time etc). The data is $W = (Y, D, X)$ where Y is an outcome, D is the vector of stuff we care about and X is the stuff we don't care about. The true values of θ and β, denoted as θ_0 and β_0, fit the data best, in the sense that

$$(\theta_0, \beta_0) = \arg\max_{\theta,\beta} \mathbb{E}_W \left[l(W, \theta, \beta) \right]$$

where $l(W, \theta, \beta)$ is some criterion (squared deviation, log likelihood etc).

The *Neyman Orthogonal Score* ψ is defined by:

$$\psi(W, \theta, \beta, \mu) = \frac{\partial}{\partial \theta} l(W, \theta, \beta) - \mu \frac{\partial}{\partial \beta} l(W, \theta, \beta)$$

The vector μ above is defined by the hessian of this criterion function. Let J be:

$$J = \begin{pmatrix} J_{\theta,\theta} & J_{\theta,\beta} \\ J_{\beta,\theta} & J_{\beta,\beta} \end{pmatrix} = \frac{\partial}{\partial \theta \partial \beta} \mathbb{E}_W \left[\frac{\partial}{\partial \theta \partial \beta} l(W, \theta, \beta) \right]$$

Then we define μ as $\mu = J_{\theta,\beta} J_{\beta,\beta}^{-1}$.

2 The Poisson Setting

In Poisson regression, the function l is

$$l(Y, D, X, \theta, \beta) = Y(D\theta + X\beta) - \exp(D\theta + X\beta)$$

and its associated gradients needed for the definition of ψ are

$$\frac{\partial}{\partial \theta} l(W, \theta, \beta) = (Y - \exp(D\theta + X\beta))D$$
$$\frac{\partial}{\partial \beta} l(W, \theta, \beta) = (Y - \exp(D\theta + X\beta))X$$

The entries in the Hessian matrix that we need to compute μ are:

$$J_{\theta,\theta} = -\mathbb{E}\left[D'D \exp(D\theta + X\beta)\right]$$
$$J_{\theta,\beta} = -\mathbb{E}\left[D'X \exp(D\theta + X\beta)\right]$$
$$J_{\beta,\beta} = -\mathbb{E}\left[X'X \exp(D\theta + X\beta)\right]$$

yielding this expression for μ:

$$\mu = \mathbb{E}\left[D'X \exp(D\theta + X\beta)\right] \left(\mathbb{E}\left[X'X \exp(D\theta + X\beta)\right]\right)^{-1}$$

I *think* this constructing is revealing, since it looks like weighted least squares, with D as the outcome, X as the covariates, and weights equal to $\exp(D\theta + X\beta)$.

The Neyman Orthogonal moment for Poisson regression is then:

$$\psi = (Y - \exp(D\theta + X\beta))(D - X\mu)$$

How would we implement this? These steps give a single point estimate $\widehat{\theta}$ and an associated covariance matrix for a given split structure. See below for how we combine point estimates and covariance matrices across many split structures into a single point estimate/covariance matrix that should be less sensitive to the monte carlo nature of splitting.

1. Make a bunch of splits of the data into training and estimation sets.

2. In a **training** set k, use Poisson LASSO and regular Poisson regression to get initial estimates of θ and β that we'll call $\widetilde{\theta}$ and $\widetilde{\beta}$.

    - Use the LASSO step to pick the X's that count.

    - Use the regular step to estimate $\widetilde{\theta}_k$ and $\widetilde{\beta}_k$ with all of D and the chosen subset of X.

3. In the corresponding **estimation** set k, compute $s_k = X\widetilde{\beta}_k$.

4. Back in the **training** set k, compute weights $w_k = \exp(D\widetilde{\theta}_k + X\widetilde{\beta}_k)$. Compute a linear LASSO of D on X using those weights. Based on the selected covariates there, do weighted OLS, again with those weights, on the selected covariates. The coefficients of this are μ_k.
   - Note, the STATA package for this doesn't do weighted OLS in the second step, they do regular OLS. I guess we want a flag here to possibly mimic STATA.
5. Finally, in the **estimation** set K, construct the moment $(Y - \exp(D\theta + s))(D - X\mu_k)$.
6. Since we'll (probably?) focus on the DML2 algorithm, for each k, compute the average of that moment, as a function of θ, and then average over each of those averages to get the final objective function we want.
   - Note, this is **also** different from what STATA does. It seems like they compute this moment in one step using all the data, and ignore the hold out structure.
7. If D is univariate, we can just do root-finding. If D is multivariate, we won't be able to match this exactly, so lets minimize squared deviations from zero.

To get a covariance matrix for this estimate of θ, we first compute J_0, defined by:

$$\begin{aligned} J_0 &= \frac{\partial}{\partial\theta}\mathbb{E}_W \psi(Y, D, X, \widehat{\theta}, \widetilde{\theta}, \widetilde{\beta}) \\ &= -\mathbb{E}_W\left[D' \exp(D\widehat{\theta} + s)(D - X\widetilde{\mu})\right] \end{aligned}$$

Next we compute Ψ:

$$\begin{aligned} \Psi &= \mathbb{E}_W\left[\psi(W, \widehat{\theta}, \widetilde{\theta}, \widetilde{\beta})\psi(W, \widehat{\theta}, \widetilde{\theta}, \widetilde{\beta})'\right] \\ &= \mathbb{E}_W\left[(Y - \exp(D\widehat{\theta} + s))^2 (D - X\widetilde{\mu})(D - X\widetilde{\mu})'\right] \end{aligned}$$

In both cases, I think we'd compute each of these as average within

2.1 How would the linear version work?

Exactly the same way. Get rid of the exp's, and wherever the above says "Poisson" replace with "OLS". Moment is now $(Y - D\theta - s)(D - X\mu)$. I **think** we ignore the weights in the step where we compute μ?