

LASSO/Poisson DML implementation

Thomas R. Covert, Yixin Sun

September 2, 2020

Contents

1	Setup from Chernozhukov et al - Finite Nuisance Parameter Approach	1
1.1	The Linear Setting - LASSO Implementation	2
1.2	The Poisson Setting - LASSO Implementation	4
2	Concentrating Out Approach	6
2.1	The Linear Setting - LASSO implementation	6
2.2	The Poisson Setting - LASSO implementation	8
2.3	Regression Forest Setting	10

1 Setup from Chernozhukov et al - Finite Nuisance Parameter Approach

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

$$(\theta_0, \beta_0) = \operatorname{argmax}_{\theta, \beta} E_{\mathcal{W}} [l(W, \theta, \beta)]$$

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:

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

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

1.1 The Linear Setting - LASSO Implementation

The linear regression, the function l is

$$l(W; \theta, \beta) = -\frac{(Y - D\theta - X'\beta)^2}{2}$$

and the necessary gradients needed for ψ are:

$$\partial\ell_\theta(W; \theta, \beta) = (Y - D\theta - X'\beta) D$$

$$\partial\ell_\beta(W; \theta, \beta) = (Y - D\theta - X'\beta) X$$

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

$$J_{\theta\beta} = -E [DX']$$

$$J_{\beta\beta} = -E [XX']$$

yielding this for μ :

$$\mu = E [DX'] (E [XX'])^{-1}$$

The Neyman orthogonal score is then given by:

$$\psi(W; \theta, \eta) = (Y - D\theta - X'\beta) (D - \mu X)$$

Implementation:

These steps give a single point estimate, $\hat{\theta}$ 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 k splits of the data into training and estimation sets. Default is $k = 5$ folds in our implementation

For each $k = 1, \dots, K$, implement the following steps:

2. In a **training** set k , use a linear LASSO of Y on D and X to select the covariates (making sure D is always included), \hat{X}_k . Then fit a linear regression of Y on \hat{X}_k and D , and let $\hat{\beta}_k$ be the estimated coefficients on \hat{X}_k .
3. In the **estimation** set k , compute $s_k = X\hat{\beta}_k$.
4. In the **training** set k , compute a linear LASSO of D_j on X to select covariates $\tilde{X}_{k,j}$, for each variable of interest. Fit a linear regression of D_j on $\tilde{X}_{k,j}$, and denote this $\tilde{\mu}_{k,j}$. Collect the $\tilde{\mu}_{k,j}$ into a vector to form $\tilde{\mu}_k$.
5. In the **estimation** set, construct

$$m_k = X\tilde{\mu}_k$$

6. Using the DML2 algorithm, for each k , construct the average of the moment:

$$\frac{1}{n} \sum_{i=1}^n (D_i - m_i)' (Y_i - s_i - D_i \theta')$$

Then average over each of these folds to get the final objective function, which we use to compute $\hat{\theta}$ by minimizing squared deviations from zero (if D is univariate, we can just do root-finding). 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. To get a covariance matrix for this estimate of θ , we first compute J_0 defined by:

$$\begin{aligned} J_0 &= \frac{\partial}{\partial \theta} E_W [\psi(W, \hat{\theta}, \hat{\beta})] \\ &= -E_W [D'(D - m)] \end{aligned}$$

Next we compute Ψ :

$$\begin{aligned} \Psi &= E_W [\psi(W, \theta, \tilde{\theta}, \tilde{\beta}) \psi(W, \theta, \tilde{\theta}, \tilde{\beta})'] \\ \psi(W, \hat{\theta}, \tilde{\theta}, \tilde{\beta}) &= (Y_i - s_i - D_i \hat{\theta}') (D_i - m_i) \end{aligned}$$

And we can compute:

$$\widehat{\text{Var}}(\hat{\theta}) = \frac{1}{n} J_0^{-1} \Psi J_0^{-1}$$

1.2 The Poisson Setting - LASSO Implementation

In Poisson regression, the function l is

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

and its associated gradients needed for the definition of ψ are

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

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

$$\begin{aligned} J_{\theta, \theta} &= -E [D' D \exp(D\theta + X\beta)] \\ J_{\theta, \beta} &= -E [D' X \exp(D\theta + X\beta)] \\ J_{\beta, \beta} &= -E [X' X \exp(D\theta + X\beta)] \end{aligned}$$

yielding this for μ :

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

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:

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

Implementation:

1. Make a bunch of splits of the data into training and estimation sets.
2. In a **training** set k , use Poisson LASSO of Y on D and X to select the covariates, \hat{X}_k (making sure D is always included). Then fit a Poisson regression of Y on \hat{X}_k and D , and let $\hat{\beta}_k$ be the estimated

coefficients on \hat{X}_k .

3. In the corresponding **estimation** set k , compute $s_k = X\hat{\beta}_k$.
4. Back in the **training** set k , compute weights $w_k = \exp(D\hat{\theta}_k + X\hat{\beta}_k)$ using the results from step 2. Compute a linear LASSO of D_j on X using those weights to select covariates $\tilde{X}_{k,j}$ for each variable of interest. Fit a linear regression of D_j on $\tilde{X}_{k,j}$, and denote this $\tilde{\mu}_{k,j}$. Collect the $\tilde{\mu}_{k,j}$ into a vector to form $\tilde{\mu}_k$.

- Should we be using weights in the linear regression too?

5. In **estimation** set, construct

$$m_k = X\tilde{\mu}_k$$

6. Using the DML2 algorithm, for each k , construct the average of the moment:

$$\frac{1}{n} \sum_{i=1}^n (D_i - m_i)' (Y_i - \exp(s_i - D_i\theta'))$$

Then average over each of these folds to get the final objective function, which we use to compute $\hat{\theta}$ by minimizing squared deviations from zero (if D is univariate, we can just do root-finding).

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

$$\begin{aligned} J_0 &= \frac{\partial}{\partial \theta} E_W \left[\psi(Y, D, X, \hat{\theta}, \tilde{\theta}, \tilde{\beta}) \right] \\ &= -E_W \left[D' \exp(D\hat{\theta} + s)(D - X\tilde{\mu}) \right] \end{aligned}$$

Next we compute Ψ :

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

In both cases, I think we'd compute each of these as the average over points in the estimation set

k , and then average over each of the estimation sets within a split structure.¹ We then get:

$$\widehat{Var}(\hat{\theta}) = \frac{1}{n} J_0^{-1} \Psi J_0^{-1}$$

2 Concentrating Out Approach

Chernozhukov et. al write that the approach for constructing Neyman orthogonal scores is closely related to the “concentrating-out approach”. For all $\theta \in \Theta$, let β_θ be the solution of the following optimization problem:

$$\operatorname{argmax}_{\beta \in \mathcal{B}} E[\ell(W; \theta, \beta)]$$

where β_θ satisfies

$$\partial_\beta E[\ell(W; \theta, \beta_\theta)] = 0$$

Differentiating this with respect to θ and interchanging the order of differentiation gives us

$$\begin{aligned} 0 &= \partial_\theta \partial_\beta E[\ell(W; \theta, \beta_\theta)] = \partial_\beta \partial_\theta E[\ell(W; \theta, \beta_\theta)] \\ &= \partial_\beta E[\partial_\theta \ell(W; \theta, \beta_\theta) + [\partial_\theta \beta_\theta]' \partial_\beta \ell(W; \theta, \beta_\theta)] \\ &= \partial_\beta E[\psi(W; \theta, \beta, \partial_\theta \beta_\theta)]|_{\beta=\beta_\theta} \end{aligned}$$

So our score function here is:

$$\psi(W; \theta, \beta, \partial_\theta \beta_\theta) := \partial_\theta \ell(W; \theta, \beta) + [\partial_\theta \beta_\theta]' \partial_\beta \ell(W; \theta, \beta)$$

2.1 The Linear Setting - LASSO implementation

Consider again the function:

$$\ell(W; \theta, \beta) = -\frac{(Y - D\theta - \beta(X))^2}{2}$$

Taking FOC with respect to β , we get:

$$\begin{aligned} 0 &= E[Y - D\theta - \beta(X)|X] \\ \Rightarrow \beta(X) &= E[Y - D\theta|X] \end{aligned}$$

¹For an example of this averaging, see the formula for \hat{J}_0 on page C27 of the original DML paper.

So we get the following Neyman orthogonal score

$$\begin{aligned}
\psi(W; \theta, \beta_\theta) &= -\frac{1}{2} \frac{d \{Y - D\theta - E[Y - D\theta|X]\}^2}{d\theta} \\
&= (D - E[D|X]) \times (Y - E[Y|X] - (D - E[D|X])\theta) \\
&= (D - m(X)) \times (Y - s(x) - (D - m(x))\theta)
\end{aligned}$$

Implementation:

1. Make k splits of the data into training and estimation sets. Default is $k = 5$ folds in our implementation

For each $k = 1, \dots, K$, implement the following steps:

2. In a **training** set k , use a linear LASSO to generate a model of Y on and X to select the covariates, \hat{X}_k . Then fit a linear regression of Y on \hat{X}_k , and let $\hat{\beta}_k$ be the estimated coefficients on \hat{X}_k .
3. In the **estimation** set k , compute $s_k = \hat{X}_k \hat{\beta}_k$
4. In the **training** set k , compute a linear LASSO of D_j on X to select covariates $\tilde{X}_{k,j}$ for each variable of interest. Fit a linear regression of D_j on $\tilde{X}_{k,j}$, and denote the estimated coefficients as $\tilde{\mu}_{k,j}$. Collect the $\tilde{\mu}_{k,j}$ into a vector to form $\tilde{\mu}_k$.
5. In the **estimation** set, construct

$$m_k = X \tilde{\mu}_k$$

6. Using the DML2 algorithm, for each k , construct the average of the moment:

$$\frac{1}{n} \sum_{i=1}^n (D_i - m_i)' (Y_i - s_i - (D_i - m_i)\theta')$$

Then average over each of these folds to get the final objective function, which we use to compute $\hat{\theta}$ by minimizing squared deviations from zero.

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

$$\begin{aligned}
J_0 &= \frac{\partial}{\partial \theta} E_W [\psi(W, \hat{\theta}, \hat{\beta})] \\
&= -E_W [(D - m)'(D - m)]
\end{aligned}$$

Next we compute Ψ :

$$\begin{aligned}\Psi &= E_W \left[\psi(W, \theta, \tilde{\theta}, \hat{\beta}) \psi(W, \theta, \tilde{\theta}, \hat{\beta})' \right] \\ \psi(W, \hat{\theta}, \tilde{\theta}, \hat{\beta}) &= (Y_i - s_i - (D_i - m_i) \hat{\theta}') (D_i - m_i)\end{aligned}$$

And we can compute:

$$\widehat{\text{Var}}(\hat{\theta}) = \frac{1}{n} J_0^{-1} \Psi J_0^{-1}$$

2.2 The Poisson Setting - LASSO implementation

We have only solved the orthogonal score for the case when D is a univariate binary variable. Consider again for the regression case, the function l is

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

Taking FOC with respect to β , we get:

$$\begin{aligned}\beta_\theta(X) &= E[\exp(D\theta + \beta(X))|X] \\ &= \exp(\beta(X)) E[\exp(D\theta)|X] \\ &= \exp(\beta(x)) (Pr(D = 1|X) \exp(\theta) + (1 - Pr(D = 1|X))) \\ &= \exp(\beta(x)) (E[D|X] \exp(D\theta) + (1 - E[D|X]))\end{aligned}$$

Let $s(X) = E[Y|X]$ and $m(X) = E[D|X]$ and rearrange to get:

$$\exp(\beta_\theta(X)) = \frac{s(X)}{\exp(\theta)m(X) + 1 - m(X)}$$

From this, we have the following:

$$\begin{aligned}\beta_\theta(X) &= \log \left(\frac{s(X)}{\exp(\theta)m(X) + 1 - m(X)} \right) \\ \partial_\theta \beta_\theta &= - \frac{m(X) \exp(\theta)}{m(X) \exp(\theta) + (1 - m(x))}\end{aligned}$$

Putting this altogether and plugging into the score function, we get:

$$\psi(W; \theta, s(X), m(X)) = \left(Y - \frac{\exp(D\theta)s(X)}{\exp(\theta)m(X) + 1 - m(X)} \right) \left(D - \frac{\exp(\theta)m(X)}{\exp(\theta)m(X) + 1 - m(X)} \right)$$

Implementation:

Steps 1 - 5 are the same as the previous

6. Using the DML2 algorithm, for each k , construct the average of the moment:

$$\frac{1}{n} \sum_{i=1}^n \left(Y_i - \frac{\exp(D_i\theta)s_i}{\exp(\theta)m_i + 1 - m_i} \right) \left(D_i - \frac{\exp(\theta)m_i}{\exp(\theta)m_i + 1 - m_i} \right)$$

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

$$J_0 = \frac{\partial}{\partial \theta} E_W \left[\psi(W, \hat{\theta}, \hat{\beta}) \right]$$

To break down this calculation, let

$$\begin{aligned} P_i &= \exp(\theta)m_i + 1 - m_i \\ A_i &= \frac{\exp(D_i\theta)s_i}{P} \\ B_i &= \frac{\exp(\theta)m_i}{P} \end{aligned}$$

So we have

$$J_0 = E_W \left[A \times \left(\frac{\exp(\theta)m}{P} + \frac{\exp(\theta)^2 s^2}{P^2} \right) + B \times \left(\frac{D \exp(D\theta)s}{P} + \frac{\exp(D\theta)s}{P^2} \right) \right]$$

As always, we have

$$\Psi = E_W \left[\psi(W, \theta, \tilde{\theta}, \hat{\beta}) \psi(W, \theta, \tilde{\theta}, \hat{\beta})' \right]$$

And we can compute

$$\widehat{\text{Var}}(\hat{\theta}) = \frac{1}{n} J_0^{-1} \Psi J_0^{-1}$$

2.3 Regression Forest Setting

Regression forest can be used instead of LASSO for both the linear and Poisson concentrating-out approach.

Implementation:

1. Make k splits of the data into training and estimation sets. Default is $k = 5$ folds in our implementation

For each $k = 1, \dots, K$, implement the following steps:

2. In **training** set k , use a regression forest to generate a model of Y on and X . Use this trained regression forest and apply to data in the **estimation** set to calculate $s_k = E[Y|X]$.
3. In **training** set k , use a regression forest to generate a model of D_j on X . Use this trained regression forest and apply to data in the **estimation** set to calculate $m_{k,j} = E[D_j|X]$. Collect to form $m_k = (m'_{k,1}, m'_{k,2}, \dots, m'_{k,J})$
4. The rest of the implementation should follow steps 6 and 7 of either the linear or Poisson setting, depending on your model.