Private Regression using Block Coordinate Descent

Erik-Jan van Kesteren

Utrecht University

2019-03-15

Outline

Regression models and vertical partitioning

Coordinate descent

The privreg package

Regression models and vertical partitioning

Basic regression model

Basic setup:

$$\boldsymbol{y} = \boldsymbol{x} \cdot \beta + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} = \boldsymbol{y} - \boldsymbol{x} \cdot \beta$$
$$\widehat{\boldsymbol{y}} = \boldsymbol{x} \cdot \hat{\beta}, \qquad \boldsymbol{y}_{res} = \boldsymbol{y} - \boldsymbol{x} \cdot \hat{\beta}$$

ML / LS estimate

$$\hat{\beta} = \frac{COV(\boldsymbol{x}, \boldsymbol{y})}{VAR(\boldsymbol{x})}$$

If $\boldsymbol{X}$ and $\boldsymbol{y}$ are centered:

$$\hat{\beta} = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$$

Basic regression model with P predictors

Basic setup:

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$$

ML / LS estimates:

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$$

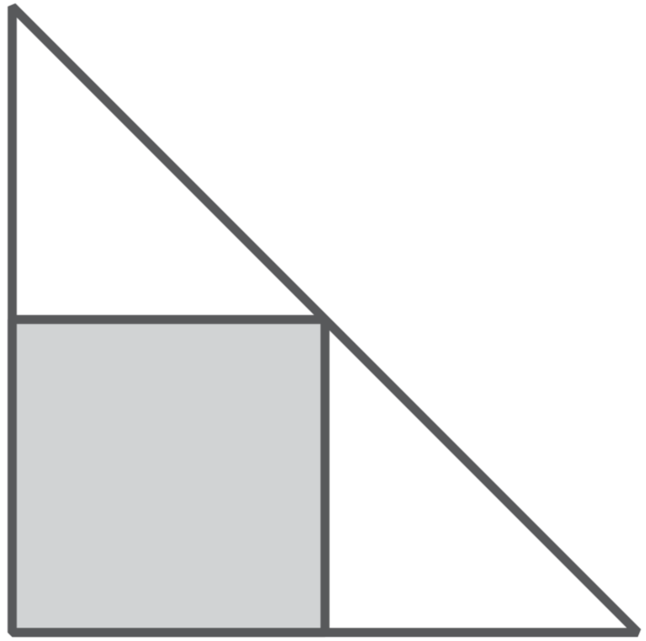
If $\boldsymbol{X}$ and $\boldsymbol{y}$ are centered:

$$\boldsymbol{X}'\boldsymbol{X} = COV(\boldsymbol{X}), \qquad \boldsymbol{X}'\boldsymbol{y} = COV(\boldsymbol{X}, \boldsymbol{y})$$

Regression with vertically partitioned data

- Alice and Bob are two institutions
- Alice and Bob cannot share their predictors

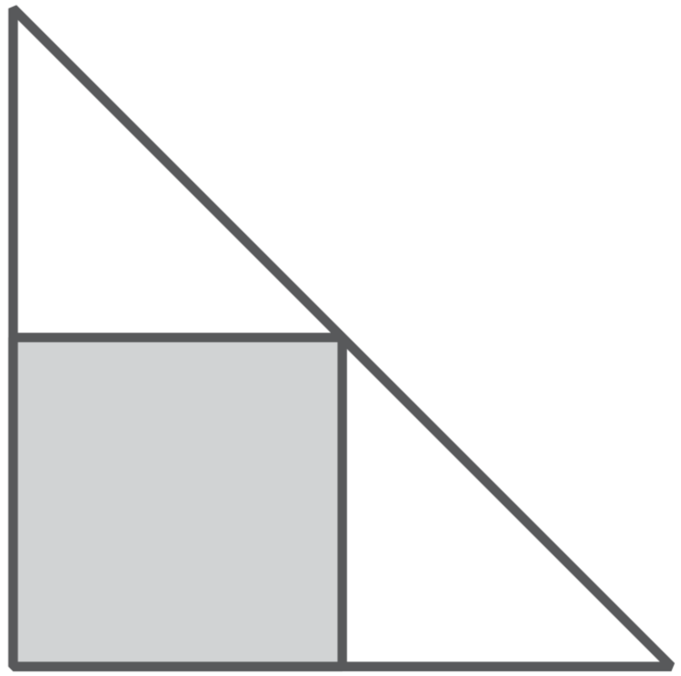
- Two sets of non-overlapping predictors $\boldsymbol{X}_a$ and $\boldsymbol{X}_b$
- $P_a \times P_b$ missing covariance matrix elements



Privacy note

- Alice is not allowed to know the values of $\boldsymbol{X}_b$ and vice versa
- We can securely create $COV(\boldsymbol{X})$ (Karr et al., 2009; Hall et al., 2011)
- But sharing $COV(\boldsymbol{X})$ means Alice can predict $\boldsymbol{x}_b \in \boldsymbol{X}_b$ with some certainty:

$$\widehat{\boldsymbol{x}}_b = \boldsymbol{X}_a(\boldsymbol{X}_a'\boldsymbol{X}_a)^{-1}COV(\boldsymbol{X}_a, \boldsymbol{x}_b)$$



Coordinate descent

Coordinate descent for multiple regression

Marginal parameter:

$$\hat{\beta}_1^m = \langle \boldsymbol{x}_1, \boldsymbol{y} \rangle / \langle \boldsymbol{x}_1, \boldsymbol{x}_1 \rangle$$

Conditional parameter:

$$\hat{\beta}_1 = \langle \boldsymbol{x}_1, \boldsymbol{y}_{res}^1 \rangle / \langle \boldsymbol{x}_1, \boldsymbol{x}_1 \rangle, \qquad \boldsymbol{y}_{res}^1 = \boldsymbol{y} - \underbrace{\boldsymbol{X}^1 \widehat{\boldsymbol{\beta}}^1}_{excl\ \boldsymbol{x}_1}$$

Because by definition: $\boldsymbol{y}_{res} = \boldsymbol{\epsilon} \perp \boldsymbol{X}$

Coordinate descent for multiple regression

```r
X        <- matrix(rnorm(100), 20)
y        <- X %*% runif(5, -1, 1) + rnorm(20)
beta_hat <- solve(crossprod(X), crossprod(X, y))

y_res_1  <- y - X[,-1] %*% beta_hat[-1]
beta_1   <- crossprod(X[,1], y_res_1) / crossprod(X[,1])

beta_1                          # Same
beta_hat[1]                     # Same
cov(X[,1], y) / var(X[,1]) # Different


```

Coordinate descent for multiple regression

1. Initialise $\widehat{\boldsymbol{\beta}} \leftarrow \mathbf{0}_P$
2. For $p \in P$:
    1. $\boldsymbol{y}_{res}^{p} = \boldsymbol{y} - \boldsymbol{X}^{p}\widehat{\boldsymbol{\beta}}^{p}$
    2. $\widehat{\boldsymbol{\beta}}_p \leftarrow \langle \boldsymbol{x}_p, \boldsymbol{y}_{res}^{p} \rangle / \langle \boldsymbol{x}_p, \boldsymbol{x}_p \rangle$
3. Repeat until convergence

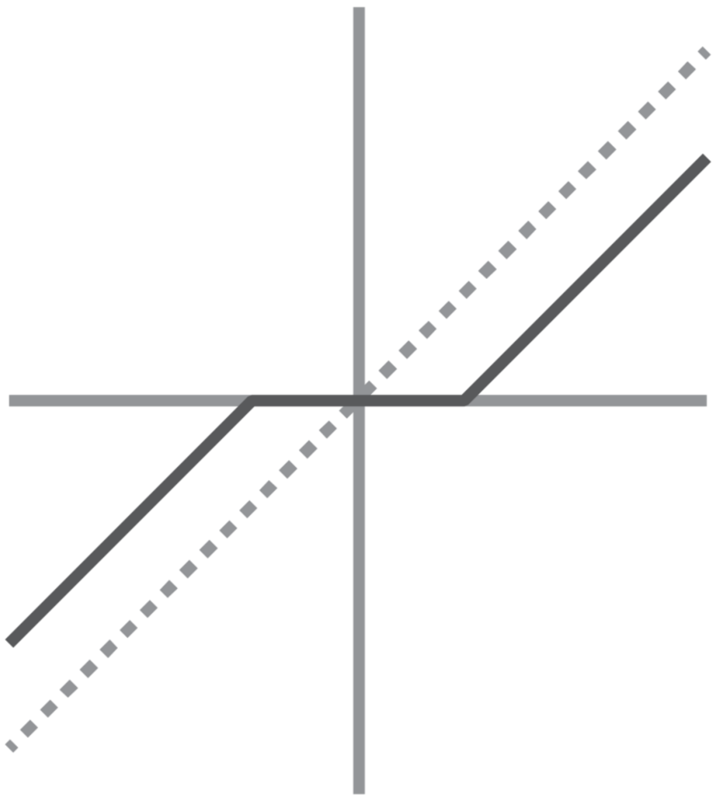
Example implementation in R:

https://gist.github.com/vankesteren/88621f8a0d532083691e6f2a505dd5c7

Coordinate descent for multiple regression

Soft-thresholding $\widehat{\boldsymbol{\beta}}_i$ in each step of the algorithm leads to the LASSO estimate (Hastie et al., 2015):

1. Initialise $\widehat{\boldsymbol{\beta}} \leftarrow \mathbf{0}_P$
2. For $p \in P$:
   1. $\boldsymbol{y}_{res}^p = \boldsymbol{y} - \boldsymbol{X}^p \widehat{\boldsymbol{\beta}}^p$
   2. $\widehat{\boldsymbol{\beta}}_p \leftarrow S_\lambda\left(\langle \boldsymbol{x}_p, \boldsymbol{y}_{res}^p \rangle / \langle \boldsymbol{x}_p, \boldsymbol{x}_p \rangle\right)$
3. Repeat until convergence



Block coordinate descent for regression

Assume we have K blocks

1. Initialise $\widehat{\boldsymbol{\beta}} \leftarrow \mathbf{0}_P$
2. For $k \in K$:
   1. $\boldsymbol{y}_{res}^k = \boldsymbol{y} - \boldsymbol{X}^k \widehat{\boldsymbol{\beta}}^k$
   2. $\widehat{\boldsymbol{\beta}}_k \leftarrow (\boldsymbol{X}_k' \boldsymbol{X}_k)^{-1} \boldsymbol{X}_k \boldsymbol{y}_{res}^k$
3. Repeat until convergence

privreg

Set $\boldsymbol{\epsilon}^{b} = \boldsymbol{y}$

Alice

$$\widehat{\boldsymbol{\beta}}_a = (\boldsymbol{X}_a'\boldsymbol{X}_a)^{-1}\boldsymbol{X}_a\boldsymbol{y}_{res}^{b}$$

$$\boldsymbol{y}_{res}^{a} = \boldsymbol{y} - \boldsymbol{X}_a\widehat{\boldsymbol{\beta}}_a$$

Send $\boldsymbol{y}_{res}^{a}$ to Bob

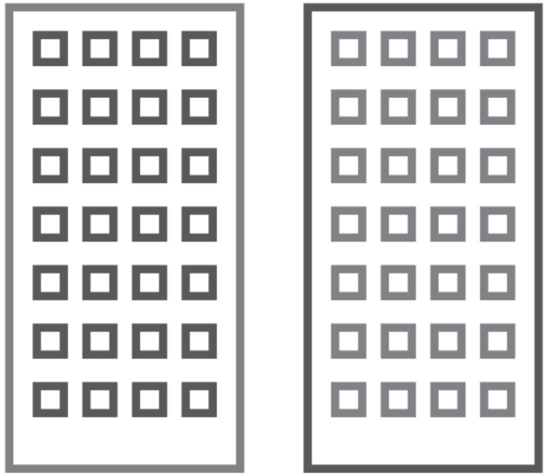
Bob

$$\widehat{\boldsymbol{\beta}}_b = (\boldsymbol{X}_b'\boldsymbol{X}_b)^{-1}\boldsymbol{X}_b\boldsymbol{y}_{res}^{a}$$

$$\boldsymbol{y}_{res}^{b} = \boldsymbol{y} - \boldsymbol{X}_b\widehat{\boldsymbol{\beta}}_b$$

Send $\boldsymbol{y}_{res}^{b}$ to Alice

privreg



Result

Coefficients:

$$(\hat{\boldsymbol{\beta}}_a, \hat{\boldsymbol{\beta}}_b) = \hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$$

Prediction:

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}_a + \hat{\boldsymbol{y}}_b = \boldsymbol{X}_a\hat{\boldsymbol{\beta}}_a + \boldsymbol{X}_b\hat{\boldsymbol{\beta}}_b$$

vankesteren / privreg

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Privacy-preserving regression

Edit

Manage topics

15 commits 1 branch 0 releases 1 contributor View license

Branch: master New pull request Create new file Upload files Find File Clone or download

vankesteren Update README.md Latest commit 917053f an hour ago

R	added modeling with formulas	16 days ago
man	Skip automated test, set up travis	18 days ago
tests	added modeling with formulas	16 days ago
.Rbuildignore	update travis, add .yml to build ignore	18 days ago
.gitattributes	Initial commit	29 days ago
.gitignore	Initial commit	29 days ago
.travis.yml	update travis, add .yml to build ignore	18 days ago
DESCRIPTION	Add message encryption via openssl, clean up code	18 days ago
LICENSE	change to port 443	21 days ago
LICENSE.md	Initial commit	29 days ago
NAMESPACE	Add message encryption via openssl, clean up code	18 days ago
README.md	Update README.md	an hour ago
privreg.Rproj	Initial commit	29 days ago

README.md

`privreg`: Private regression using block coordinate descent

build passing

install

Time for a tryout

github.com/vankesteren/privreg