Fast approximate leave-one-out cross-validation for large sample sizes

Rosa Meijer Jelle Goeman

Department of Medical Statistics
Leiden University Medical Center

Validation in Statistics and Machine Learning
6 October 2010

Outline

Penalized regression

Ridge regression

$$\hat{\beta}_{\text{ridge}} = \text{argmax}\{l(\boldsymbol{\beta}) - \lambda \sum_i \beta_i^2\}$$

Shrinks

Lasso regression

$$\hat{\beta}_{\text{ridge}} = \text{argmax}\{l(\boldsymbol{\beta}) - \lambda \sum_i |\beta_i|\}$$

Shrinks and selects

The penalized package

On CRAN: R package penalized

- Ridge
- Lasso
- Elastic net

Regression models

- Linear regression
- Logistic regression (GLM)
- Cox Proportional Hazards model

Choosing the value of λ

Between

λ too large: over-shrinkage

λ too small: overfit

Choosing the value of λ

Between

λ too large: over-shrinkage

λ too small: overfit

How to optimize λ?

- Leave-one-out cross-validation
- K-fold cross-validation
- Akaike's information criterion
- Generalized cross-validation
- (.632+) bootstrap cross-validation
- ...

Leave-one-out

Ingredients

- Response $y_1, \ldots, y_n$
- Predictor variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- Fitted models $\hat{\beta}^{\lambda}_{(-i)}$ not using x_i and y_i
- A loss function L. Assume continuity.

Leave-one-out loss

$$\sum_{i=1}^{n} L(y_i, x_i, \hat{\beta}^{\lambda}_{(-i)})$$

Approximate leave-one-out

Leave-one-out loss

Requires calculation of $\hat{\beta}^{\lambda}_{(-1)}, \ldots, \hat{\beta}^{\lambda}_{(-n)}$

Approximate leave-one-out

Leave-one-out loss

Requires calculation of $\hat{\beta}^{\lambda}_{(-1)}, \ldots, \hat{\beta}^{\lambda}_{(-n)}$

Time consuming

- when n is large
- when each $\hat{\beta}^{\lambda}_{(-i)}$ takes much time
- double cross-validation

Approximate leave-one-out

Leave-one-out loss

Requires calculation of $\hat{\beta}^{\lambda}_{(-1)}, \dots, \hat{\beta}^{\lambda}_{(-n)}$

Time consuming

- when n is large
- when each $\hat{\beta}^{\lambda}_{(-i)}$ takes much time
- double cross-validation

Solution

approximate $\hat{\beta}^{\lambda}_{(-i)}$ based on $\hat{\beta}^{\lambda}$

Models

Assumption

$$-\frac{\partial^2 l}{\partial \boldsymbol{\eta} \partial \boldsymbol{\eta}'} = \mathbf{D} \qquad \text{(diagonal)}$$

with $\boldsymbol{\eta} = \mathbf{X}\beta$ the linear predictor

Generalized linear models

- Linear regression
- Logistic regression
- Cox proportional hazards (full likelihood)

General idea

Taylor approximation of $l'_{(-i)}(\beta)$ at $\beta = \hat{\beta}^{\lambda}$

$$l'_{(-i)}(\beta) = l'_{(-i)}(\hat{\beta}^{\lambda}) + (\beta - \hat{\beta}^{\lambda}) l''_{(-i)}(\hat{\beta}^{\lambda}) + O\left((\beta - \hat{\beta}^{\lambda})^2\right).$$

solving $l'_{(-i)}(\beta) = 0$ at $\beta = \hat{\beta}^{\lambda}_{(-i)}$ gives:

$$\hat{\beta}^{\lambda}_{(-i)} = \hat{\beta}^{\lambda} - \left(l''_{(-i)}(\hat{\beta}^{\lambda})\right)^{-1} l'_{(-i)}(\hat{\beta}^{\lambda}) + O\left((\hat{\beta}^{\lambda}_{(-i)} - \hat{\beta}^{\lambda})^2\right)$$

General idea

Taylor approximation of $l'_{(-i)}(\beta)$ at $\beta = \hat{\beta}^{\lambda}$

$$l'_{(-i)}(\beta) = l'_{(-i)}(\hat{\beta}^{\lambda}) + (\beta - \hat{\beta}^{\lambda}) l''_{(-i)}(\hat{\beta}^{\lambda}) + O\left((\beta - \hat{\beta}^{\lambda})^2\right).$$

solving $l'_{(-i)}(\beta) = 0$ at $\beta = \hat{\beta}^{\lambda}_{(-i)}$ gives:

$$\hat{\beta}^{\lambda}_{(-i)} = \hat{\beta}^{\lambda} - \left(l''_{(-i)}(\hat{\beta}^{\lambda})\right)^{-1} l'_{(-i)}(\hat{\beta}^{\lambda}) + O\left((\hat{\beta}^{\lambda}_{(-i)} - \hat{\beta}^{\lambda})^2\right)$$

still n inverses to be calculated

Sherman-Morrison-Woodbury theorem

$$\left(\mathbf{B} + \mathbf{u}\mathbf{v}^T\right)^{-1} = \mathbf{B}^{-1} - \frac{\mathbf{B}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{B}^{-1}}{1 + \mathbf{v}^T\mathbf{B}^{-1}\mathbf{u}},$$

$\mathbf{B}$ nonsingular $p \times p$ matrix, $\mathbf{u}$, $\mathbf{v}$ p-dimensional column vectors

Sherman-Morrison-Woodbury theorem

$$\left(\mathbf{B} + \mathbf{u}\mathbf{v}^T\right)^{-1} = \mathbf{B}^{-1} - \frac{\mathbf{B}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{B}^{-1}}{1 + \mathbf{v}^T\mathbf{B}^{-1}\mathbf{u}},$$

$\mathbf{B}$ nonsingular $p \times p$ matrix, $\mathbf{u}$, $\mathbf{v}$ p-dimensional column vectors

Apply to $(l''_{(-i)}(\hat{\beta}^{\lambda}))^{-1}$ (in the ridge model)

$$\left(\mathbf{X}_{(-i)}^T\mathbf{D}_{(-i)}\mathbf{X}_{(-i)} + \lambda\mathbf{I}_p\right)^{-1} = \left(\mathbf{X}^T\mathbf{D}\mathbf{X} + \lambda\mathbf{I}_p - d_{ii}\mathbf{x}_i\mathbf{x}_i^T\right)^{-1}$$

Final formula ridge

$$\hat{\beta}^{\lambda}_{(-i)} = \hat{\beta}^{\lambda} - \frac{\left(\mathbf{X}^T\mathbf{D}\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\mathbf{x}_i\Delta_i}{1 - v_{ii}},$$

with

$$\mathbf{V} = \mathbf{D}^{\frac{1}{2}}\mathbf{X}\left(\mathbf{X}^T\mathbf{D}\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\mathbf{X}^T\mathbf{D}^{\frac{1}{2}}$$

$\mathbf{D}$ and Δ (residuals) based on value $\hat{\beta}^{\lambda}$

all approximate $\hat{\beta}^{\lambda}_{(-i)}$'s with just 1 inverse calculation and some matrix multiplications!

Final formula ridge

$$\hat{\beta}^{\lambda}_{(-i)} = \hat{\beta}^{\lambda} - \frac{\left(\mathbf{X}^T\mathbf{D}\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\mathbf{x}_i\Delta_i}{1 - v_{ii}},$$

with

$$\mathbf{V} = \mathbf{D}^{\frac{1}{2}}\mathbf{X}\left(\mathbf{X}^T\mathbf{D}\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\mathbf{X}^T\mathbf{D}^{\frac{1}{2}}$$

$\mathbf{D}$ and Δ (residuals) based on value $\hat{\beta}^{\lambda}$

all approximate $\hat{\beta}^{\lambda}_{(-i)}$'s with just 1 inverse calculation and some matrix multiplications!

Reparamaterization

Dimension covariate space can be reduced from p to n

Models

Linear model

Approximation = exact

Models

Linear model

Approximation = exact

Cox proportional hazards

- Use full likelihood, not partial likelihood
- Baseline hazard not cross-validated
- Trick possible: add intercept term

Final formula lasso

$$\hat{\beta}^{\lambda}_{(-i)} = \hat{\beta}^{\lambda} - \frac{\left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-1}\mathbf{x}_i\Delta_i}{1 - v_{ii}},$$

with

$$\mathbf{V} = \mathbf{D}^{\frac{1}{2}}\mathbf{X}\left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{D}^{\frac{1}{2}}$$

Final formula lasso

$$\hat{\beta}^{\lambda}_{(-i)} = \hat{\beta}^{\lambda} - \frac{\left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-1}\mathbf{x}_i\Delta_i}{1 - v_{ii}},$$

with

$$\mathbf{V} = \mathbf{D}^{\frac{1}{2}}\mathbf{X}\left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{D}^{\frac{1}{2}}$$

locally, if $\hat{\beta}^{\lambda}_k \approx \hat{\beta}^{\lambda}_{(-i)_k}$ we know:

if $\hat{\beta}^{\lambda}_k = 0 \quad \Rightarrow \quad \hat{\beta}^{\lambda}_{(-i)_k} = 0$

Refinements possible

To what extent is this approximation useful?

Are the approximated values comparable to the real values?

- $cvl(\text{real } \hat{\beta}^{\lambda}_{(-i)}) \approx cvl(\text{approximated } \hat{\beta}^{\lambda}_{(-i)})$?

Would we find approximately the same values of λ?

- do we find approximately the same maximum of the *cvl* when using the approximated $\hat{\beta}^{\lambda}_{(-i)}$'s?

How much worse are the models?

- do we find approximately the same *cvl* at the maximum found?

The dataset used

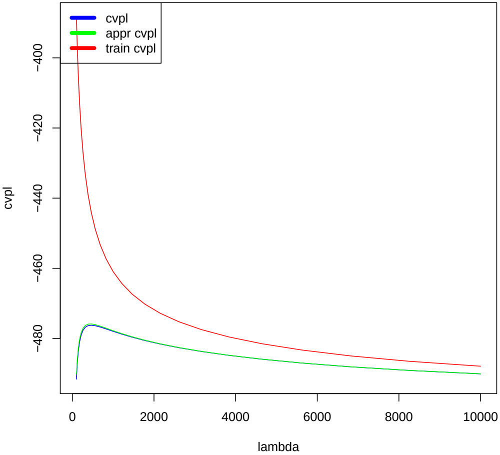
Breast cancer data of the Netherlands Cancer Institute

- Paper by Van 't Veer *et al.* (*Nature*, 2002)
- Followed up by Van de Vijver *et al.* (*NEJM*, 2002)
- 295 breast cancer patients
- effective dimension 79, due to censoring
- Microarray (Agilent): 4,919 genes preselected (Rosetta technology)

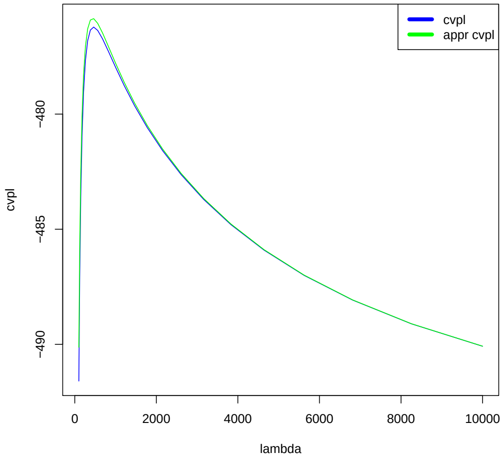
Response of interest

survival time (up to 18 years follow-up)

Ridge Regression



Ridge Regression: in more detail



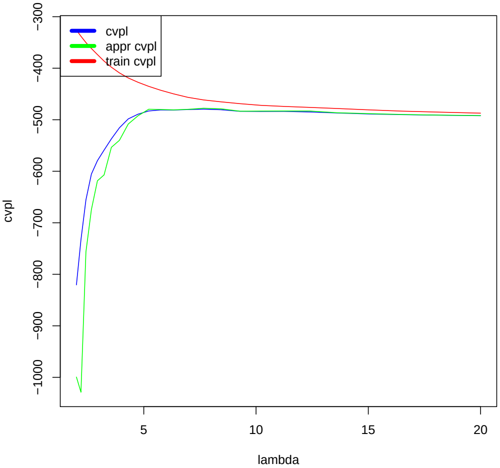
appr cvpl: lambda= 438.2634,

cvpl= -475.8422

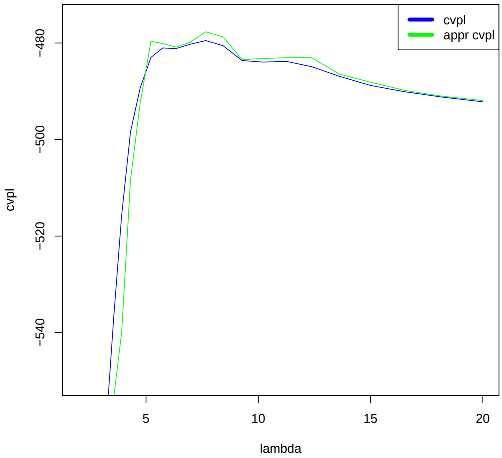
real cvpl: lambda= 458.5212,

cvpl= -476.2204

Lasso Regression



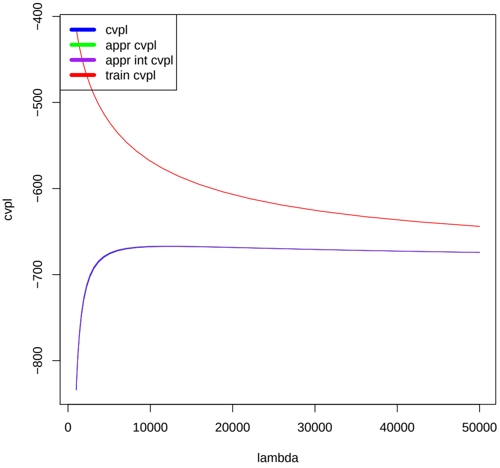
Lasso Regression: in more detail



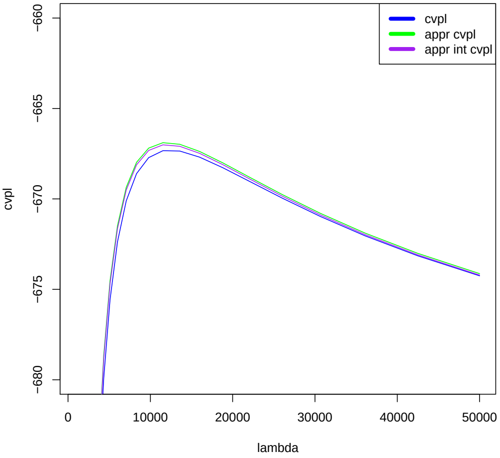
appr cvpl: lambda= 7.60564, cvpl= -477.3704

real cvpl: lambda= 7.70299, cvpl= -479.4855

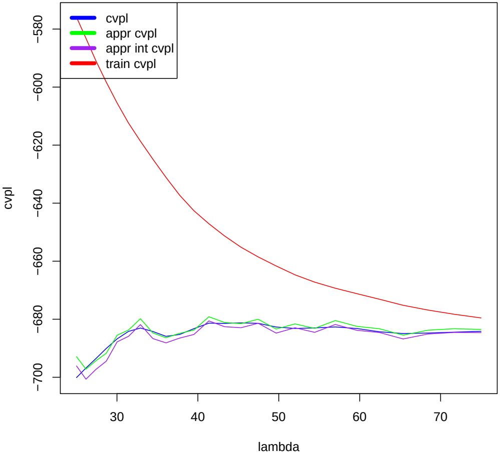
Wang breast cancer data: ridge



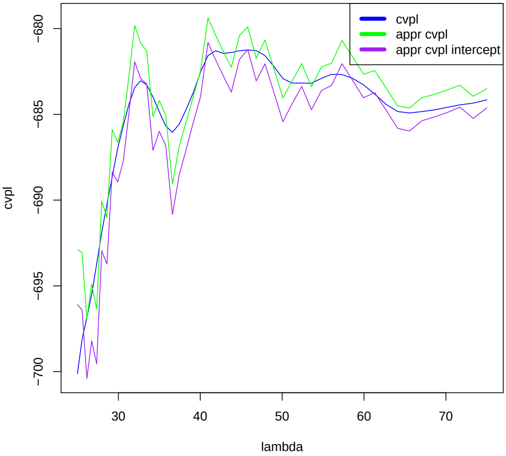
Wang breast cancer data: ridge



Wang breast cancer data: lasso



Wang breast cancer data: lasso zoomed



Efficiency

Time needed to calculate *cvl* for specific value of λ, lasso

$\lambda = 7.70$

real *cvpl*: 49.00 seconds

appr *cvpl*: 6.09 seconds

approximately 8 times as fast

Time needed to calculate *cvl* for specific value of λ, ridge

$\lambda = 458.5$

real *cvpl*: 389.27 seconds

appr *cvpl*: 17.40 seconds

more than 20 times as fast!

Some additional comments

Are these results representative of different datasets?

What aspects of a dataset determine the performance of the approximation method?

Back to the theory:

$$O\left((\hat{\beta}^{\lambda}_{(-i)} - \hat{\beta}^{\lambda})^2\right)$$

Error diminishes when:

- n gets larger
- λ gets larger

In short...

Approximate LOOCV

- gives reasonable approximate of λ in penalization methods
- reasonable outcomes of approximated *cvl*: comparisons between models possible
- works great for ridge; less stable for lasso

Can be used to find "neighborhood" of optimal λ

Best for large values of n

- best possible approximations
- most time saved

double LOOCV

Questions?