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

1. Introduction
2. The approximation method
3. Results
4. Summary
Penalized regression

Ridge regression

\[ \hat{\beta}_{\text{ridge}} = \arg\max \{ l(\beta) - \lambda \sum_i \beta_i^2 \} \]

Shrinks

Lasso regression

\[ \hat{\beta}_{\text{ridge}} = \arg\max \{ l(\beta) - \lambda \sum_i |\beta_i| \} \]

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 $\lambda$

Between

$\lambda$ too large: over-shrinkage
$\lambda$ too small: overfit
Choosing the value of $\lambda$

**Between**

- $\lambda$ too large: over-shrinkage
- $\lambda$ too small: overfit

**How to optimize $\lambda$?**

- 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 $x_1, \ldots, x_n$
- 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 \( \hat{\beta}_{(-1)}^\lambda, \ldots, \hat{\beta}_{(-n)}^\lambda \)
Approximate leave-one-out

Leave-one-out loss

Requires calculation of $\hat{\beta}_λ(−1), \ldots, \hat{\beta}_λ(−n)$

Time consuming

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

**Solution**

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

Assumption

\[- \frac{\partial^2 l}{\partial \eta \partial \eta'} = D \quad \text{(diagonal)}\]

with $\eta = X\beta$ 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**

\[
(B + uv^T)^{-1} = B^{-1} - \frac{B^{-1}uv^TB^{-1}}{1 + v^TB^{-1}u},
\]

$B$ nonsingular $p \times p$ matrix, $u$, $v$ $p$-dimensional column vectors
**Sherman-Morrison-Woodbury theorem**

\[
(B + uv^T)^{-1} = B^{-1} - \frac{B^{-1}uv^TB^{-1}}{1 + v^TB^{-1}u},
\]

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

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

\[
(X^T_{(-i)}D_{(-i)}X_{(-i)} + \lambda I_p)^{-1} = (X^TDX + \lambda I_p - d_{ii}x_ix_i^T)^{-1}
\]
Final formula ridge

\[ \hat{\beta}^\lambda_{(-i)} = \hat{\beta}^\lambda - \frac{(X^TDX + \lambda I_p)^{-1} x_i \Delta_i}{1 - \nu_{ii}}, \]

with

\[ V = D^{\frac{1}{2}}X \left( X^TDX + \lambda I_p \right)^{-1} X^TD^{\frac{1}{2}} \]

D and \( \Delta \) (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

\[ \hat{\beta}_\lambda^{(\cdot)} = \hat{\beta}_\lambda - \frac{(X^TDX + \lambda I_p)^{-1} x_i \Delta_i}{1 - v_{ii}}, \]

with

\[ V = D^{\frac{1}{2}} X (X^TDX + \lambda I_p)^{-1} X^T D^{\frac{1}{2}} \]

D and Δ (residuals) based on value \( \hat{\beta}_\lambda \)

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

Reparamaterization

Dimension covariate space can be reduced from \( p \) to \( n \)

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

Rosa Meijer, Jelle Goeman
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

\[ \hat{\beta}^\lambda_{(-i)} = \hat{\beta}^\lambda - \frac{(X^TDX)^{-1} x_i \Delta_i}{1 - \nu_{ii}}, \]

with

\[ V = D^{\frac{1}{2}} X (X^TDX)^{-1} X^T D^{\frac{1}{2}}. \]
Final formula lasso

\[ \hat{\beta}^\lambda_{(-i)} = \hat{\beta}^\lambda - \frac{(X^TDX)^{-1} x_i \Delta_i}{1 - v_{ii}}, \]

with

\[ V = D^{1/2} X \left( X^TDX \right)^{-1} X^T D^{1/2} \]

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

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

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 $\lambda$?
- 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**
- 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

Fast approximate leave-one-out cross-validation for large sample sizes
Rosa Meijer, Jelle Goeman
Ridge Regression: in more detail

![Graph showing lambda vs cvpl and appr cvpl]

**Results**

- **appr cvpl**: \(\text{lambda} = 438.2634, \text{cvpl} = -475.8422\)
- **real cvpl**: \(\text{lambda} = 458.5212, \text{cvpl} = -476.2204\)
Lasso Regression

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

Rosa Meijer, Jelle Goeman
Lasso Regression: in more detail

![Graph showing cvpl and approx cvpl](image)

**appr cvpl:** \( \lambda = 7.60564, \quad \text{cvpl} = -477.3704 \)

**real cvpl:** \( \lambda = 7.70299, \quad \text{cvpl} = -479.4855 \)
Wang breast cancer data: ridge
Wang breast cancer data: ridge

![Graph showing the approximation method for ridge regression with cvpl, appr cvpl, and appr int cvpl metrics against lambda.](image)
Wang breast cancer data: lasso
Wang breast cancer data: lasso zoomed

Fast approximate leave-one-out cross-validation for large sample sizes
Rosa Meijer, Jelle Goeman
Efficiency

Time needed to calculate $cv_l$ for specific value of $\lambda$, lasso

$\lambda = 7.70$
real $cvpl$: 49.00 seconds
appr $cvpl$: 6.09 seconds
approximately 8 times as fast

Time needed to calculate $cv_l$ for specific value of $\lambda$, 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
- \( \lambda \) gets larger
In short...

Approximate LOOCV

- gives reasonable approximate of $\lambda$ 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 $\lambda$

Best for large values of $n$

- best possible approximations
- most time saved

Double LOOCV

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

Rosa Meijer, Jelle Goeman
Questions?