Cross-validation and the estimation of σ^2 and R^2

Patrick Breheny

February 20

Introduction

- Today we will discuss the selection of λ and the estimation of σ^2 (which, in turn, allows us to quanify the signal-to-noise ratio present in the data)
- For lasso models, both of these involve tend to revolve around cross-validation, although we will discuss a few different approaches

Degrees of freedom

- In our discussion of ridge regression, we used information criteria to select λ
- All of the criteria we discussed required an estimate of the degrees of freedom of the model
- For linear fitting methods, we saw that $\mathrm{df} = \mathrm{tr}(\mathbf{S})$
- The lasso, however, is not a linear fitting method; there is no exact, closed form solution to $\mathrm{Cov}(\mathbf{y}, \hat{\mathbf{y}})$

Degrees of freedom for the lasso

- A natural proposal would be to use $\mathrm{df}(\lambda) = \|\widehat{\boldsymbol{\beta}}(\lambda)\|_0$, the number of nonzero coefficients
- From one perspective, this might seem to underestimate the true degrees of freedom, as the variables were not prespecified
- For example, in our forward selection example from our first class (Jan. 14), we selected 5 features but the true df was ≈ 19
- On the other hand, shrinkage reduces the degrees of freedom in an estimator, as we have seen in ridge regression; from this perspective, $\|\widehat{\boldsymbol{\beta}}(\lambda)\|_0$ might seem to overestimate the true degrees of freedom

Degrees of freedom for the lasso (cont'd)

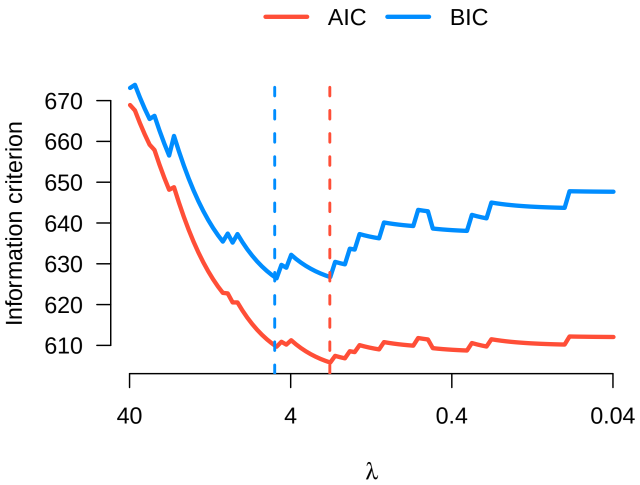
- Surprisingly, it turns out that these two factors exactly cancel and $\text{df}(\lambda) = \|\widehat{\boldsymbol{\beta}}(\lambda)\|_0$ can be shown to be an unbiased estimate of the lasso degrees of freedom
- Given this estimate, we can then use information criteria such as BIC for the purposes of selecting λ

ncvreg

- To illustrate, we will use the `ncvreg` package to fit the lasso path
- The primary purpose of `ncvreg` is to provide penalties other than the lasso, which we will discuss in our next topic
- However, it provides a `logLik` method, unlike `glmnet`, so it can be used with `R`'s `AIC` and `BIC` functions:

```
fit <- ncvreg(X, y, penalty="lasso")
AIC(fit)
BIC(fit)
```

AIC, BIC for pollution data



Remarks

- As we would expect, BIC applies a stronger penalty for overfitting and chooses a smaller, more parsimonious model than does AIC
- The main advantage of AIC and BIC is that they are computationally convenient: they can be calculated using the fit of lasso model at very little computational cost
- The primary disadvantage is that both AIC and BIC rely on a number of asymptotic approximations that can be quite inaccurate for high-dimensional data

Cross-validation: Introduction

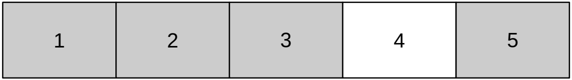
- As we have discussed, a reasonable approach to selecting λ in an objective manner is to choose the value of λ that yields the greatest predictive power
- An alternative to the approximations of AIC and BIC is to assess predictive power more directly and empirically through a technique called cross-validation
- Cross-validation is more reliable in general, although it comes at an added computation cost

Sample splitting

- As we have discussed, using the observed agreement between fitted values and the data is too optimistic; we require independent data to test predictive accuracy
- One solution, known as *sample splitting*, is to split the data set into two fractions, a training set and test set, using one portion to estimate $\widehat{\beta}$ (i.e., "train" the model) and the other to evaluate how well $\mathbf{X}\widehat{\beta}$ predicts the observations in the second portion (i.e., "test" the model)
- The problem with this solution is that we rarely have so much data that we can freely part with half of it solely for the purpose of choosing λ

Cross-validation

To finesse this problem, *cross-validation* splits the data into K folds, fits the data on $K-1$ of the folds, and evaluates prediction error on the fold that was left out



Common choices for K are 5, 10, or n (also known as leave-one-out cross-validation)

Cross-validation: Details

(1) Specify a grid of regularization parameter values $\Lambda = \{\lambda_1, \ldots, \lambda_K\}$

(2) Divide the data into V roughly equal parts $D_1, \ldots, D_V$

(3) For each $v = 1, \ldots, V$, compute the lasso solution path using the observations in $\{D_u, u \neq v\}$

(4) For each $\lambda \in \Lambda$, compute the mean squared prediction error

$$\text{MSPE}_v(\lambda) = \frac{1}{n_v} \sum_{i \in D_v} \{y_i - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}_{-v}(\lambda)\}^2,$$

where n_v is the number of observations in D_v, as well as

$$\text{CV}(\lambda) = \frac{1}{V} \sum_{v=1}^{V} \text{MSPE}_v(\lambda).$$

Cross-validation: Details (cont'd)

- Then $\hat{\lambda}$ is taken to be the value that minimizes $\mathrm{CV}(\lambda)$ and $\widehat{\boldsymbol{\beta}} \equiv \widehat{\boldsymbol{\beta}}(\hat{\lambda})$ the estimator of the regression coefficients
- Note that
  - $\mathrm{MSPE}_v(\lambda)$ is the mean squared prediction error for the model based on the training data $\{D_u, u \neq v\}$ in predicting the response variables in D_v
  - $\mathrm{CV}(\lambda)$ is an estimate of the expected mean squared prediction error, $\mathbb{E}\mathrm{PE}(\lambda)$, defined in the Feb. 11 lecture

Variability of CV estimates

- Regardless of the number of cross-validation folds, each observation in the data appears exactly once in a test set
- Letting $\hat{\mu}_i(\lambda) = \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}_{u(i)}(\lambda)$, the mean of $\{y_i - \hat{\mu}_i(\lambda)\}_{i=1}^n$ is equal to $\text{CV}(\lambda)$
- Its variability, however, is useful for estimating the accuracy with which $\mathbb{E}(\text{MSPE}(\lambda))$ is estimated

CV standard errors

- Letting $\text{SD}_{\text{CV}}(\lambda)$ denote the sample standard deviation of the $\{y_i - \hat{\mu}_i(\lambda)\}_{i=1}^n$ values, the standard error of $\text{CV}(\lambda)$ is

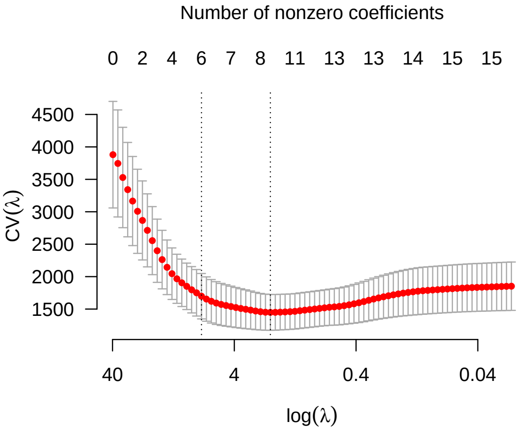
$$\text{SE}_{\text{CV}}(\lambda) = \frac{\text{SD}_{\text{CV}}(\lambda)}{\sqrt{n}},$$

  which, in turn, can be used to construct confidence intervals
- The cross-validation procedure described in this section, along with the estimates of $\text{CV}(\lambda)$ and its standard error, are implemented in `glmnet` and can be carried out using

```
cvfit <- cv.glmnet(X, y)
plot(cvfit)
```

  By default, `cv.glmnet` uses $V = 10$ folds, but this can be changed through the `nfolds` option.

CV plot for lasso: Pollution data



Intervals are $\pm 1\mathrm{SE}$

Remarks

- The value $\lambda = 1.84$ minimizes the cross-validation error, at which point 9 variables are selected
- However, as the confidence intervals show, there is substantial uncertainty about this minimum value
- A fairly wide range of λ values ($\lambda \in [0.12, 9.83]$) yield $\mathrm{CV}(\lambda)$ estimates falling within $\pm 1\mathrm{SE}_{\mathrm{CV}}$ of the minimum
- This is almost always the case in model selection: a large number of models could reasonably be considered the "best" model, subject to random variability

Repeated cross-validation

- Note that $\mathrm{CV}(\lambda)$, and hence $\widehat{\boldsymbol{\beta}}$, will change somewhat depending on the random folds
- To avoid this, some people carry out *repeated cross-validation*, and select λ according to the average CV error
- Another option is to carry out n-fold cross-validation, in which there is only one way to select the fold assignments
- It is important to realize, however, that neither of these approaches does anything to eliminate actual uncertainty with respect to the selection of λ

σ^2: Plug-in estimator

- We have discussed estimation of $\boldsymbol{\beta}$; let us now turn our attention to estimation of the residual variance, σ^2
- In ordinary least squares regression,

$$\hat{\sigma}^2_{\text{OLS}} = \frac{\text{RSS}}{n - \text{df}}$$

- For the lasso, an obvious plug-in alternative is

$$\hat{\sigma}^2_P = \frac{\text{RSS}(\lambda)}{n - \text{df}(\lambda)}$$

σ^2: CV estimator

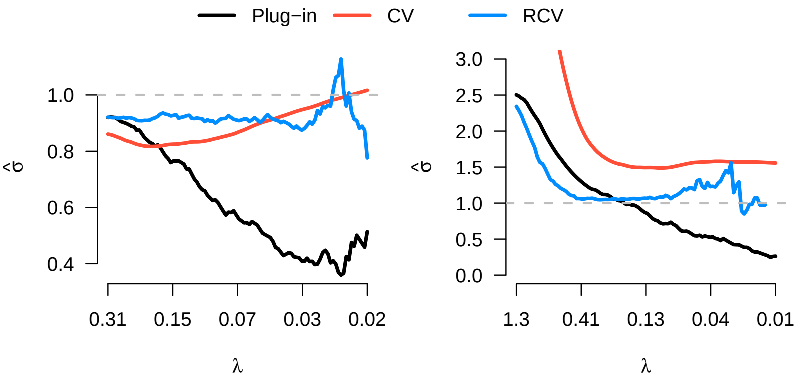
- The plug-in estimator works reasonably well in many cases, but as it is based on the observed fit of the model, tends to underestimate σ^2 for small values of λ
- An alternative approach is to use an estimate of the out-of-sample prediction error in place of the observed $\mathrm{RSS}(\lambda)$
- This is the exact quantity estimated by cross-validation:

$$\hat{\sigma}^2_{\mathrm{CV}} = \mathrm{CV}(\lambda)$$

Refitted CV

- Other, more computationally intensive methods have also been proposed based on sample splitting
- The basic idea is to randomly partitioning the dataset into two sets D_1 and D_2, use the lasso on D_1 for the purposes of variable selection, then fit an OLS model to D_2 (using the predictors selected by D_1) for the purposes of estimating σ^2
- This can be repeated several times, as well as applied in the reverse direction (switching the roles of D_1 and D_2) to obtain a more stable estimate

Comparison of estimators



$n = 100$, $p = 1,000$, $\sigma = 1$. Left: $\boldsymbol{\beta} = \mathbf{0}$; Right: $\beta_j = 1$ for $j = 1, 2, \ldots, 5$; $\beta_j = 0$ for $j = 6, 7, \ldots, 1000$

Coefficient of determination

- One reason that estimating σ^2 is of considerable practical interest is that it enables us to estimate the proportion of variance in the outcome that can be explained by the model
- This quantity, familiar from classical regression, is known as the *coefficient of determination* and denoted R^2
- The coefficient of determination is given by

$$R^2 = 1 - \frac{\mathrm{Var}(Y|\mathbf{X})}{\mathrm{Var}(Y)};$$

we have just discussed the estimation of $\sigma^2 = \mathrm{Var}(Y|\mathbf{X})$; estimation of $\mathrm{Var}(Y)$ is straightforward

R^2: Calculation in R

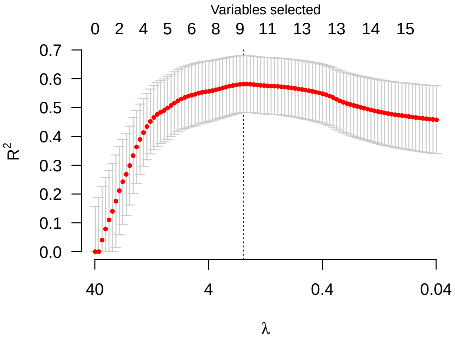
- Once cross-validation has been carried out, calculation of R^2 is straightforward
- With `glmnet`:

```
cvfit <- cv.glmnet(X, y)
rsq <- 1-cvfit$cvm/var(y)
```

- Also, the coefficient of determination is available as a plot type in `ncvreg`:

```
cvfit <- cv.ncvreg(X, y, penalty="lasso")
plot(cvfit, type="rsq")
```

R^2 plot: Pollution data



It is worth noting that only a small amount of the explained variability comes from the pollution variables: $\max R^2 = 0.58$ with the pollution variables; $\max R^2 = 0.56$ without the pollution variables

summary.cv.ncvreg

ncvreg also provides a summary() method for its cross-validation object that reports all of this information:

```
> summary(cvfit)
lasso-penalized linear regression with n=60, p=15
At minimum cross-validation error (lambda=1.9762):
-------------------------------------------------
  Nonzero coefficients: 9
  Cross-validation error (deviance): 1591.57
  R-squared: 0.58
  Signal-to-noise ratio: 1.39
  Scale estimate (sigma): 39.895
```