Risk estimation for high-dimensional lasso regression

Daniel J. McDonald
Indiana University, Bloomington
mypage.iu.edu/~dajmcdon
Joint with Darren Homrighausen

2 August 2016

Motivation


LAGAVULIN
DISTILLERY
PRODUCT OF SCOTLAND
LAGAVULIN
SINGLE ISLAY MALT WHISKY
AGED 16 YEARS
SCOTCH WHISKY
IMPORTED BY DIAGEO NORTH AMERICA, INC. NORWALK, CT

- Observe Y_i, $i = 1, \ldots, n$ real-valued response variables.
- Let X_i be a p-vector of predictors, $p \gg n$.
- Suppose
$$Y_i \sim \mathcal{N}(X_i^\top \beta_*,\ \sigma^2)$$
for some $\sigma > 0$, $\beta_* \in \mathbb{R}^p$.
- Concatenate predictors into the design matrix $\mathbb{X} \in \mathbb{R}^{n \times p}$
- Observations in the n-vector Y.

Usual goal

- Choose some procedure that does one or more of the following:
  1. Predicts new values Y from the same distribution
  2. Estimates β_* with small error
  3. Finds the support of β_*
- Since p is big, we regularize by minimizing a sum of the training error plus the lasso penalty:

$$\widehat{\beta}(\lambda) = \operatorname*{argmin}_{\beta} \frac{1}{n} \left|\left|Y - \mathbb{X}\beta\right|\right|_2^2 + \lambda \left|\left|\beta\right|\right|_1 .$$

- The problem is that we need to choose λ in some principled manner.

Selecting tuning parameters

1. Cross-validation
2. Information criteria
3. Stein's unbiased risk estimation
4. Computational tricks

$$\text{info}(C_n,\ g) := \log\left(\widehat{\text{train}}\right) + C_n g(\text{df})$$

$$\widehat{\text{train}} := \frac{1}{n}\left|\left|Y - \mathbb{X}\beta\right|\right|_2^2.$$

$$\text{df} := \frac{1}{\sigma^2}\sum_{i=1}^{n}\text{Cov}(\widehat{Y}_i,\ Y_i)$$

- AIC: $C_n = 2/n,\ g(z) = z$
- BIC: $C_n = \log n,\ g(z) = z$
- log GCV: $C_n = -2/n,\ g(z) = 1 - z/n$

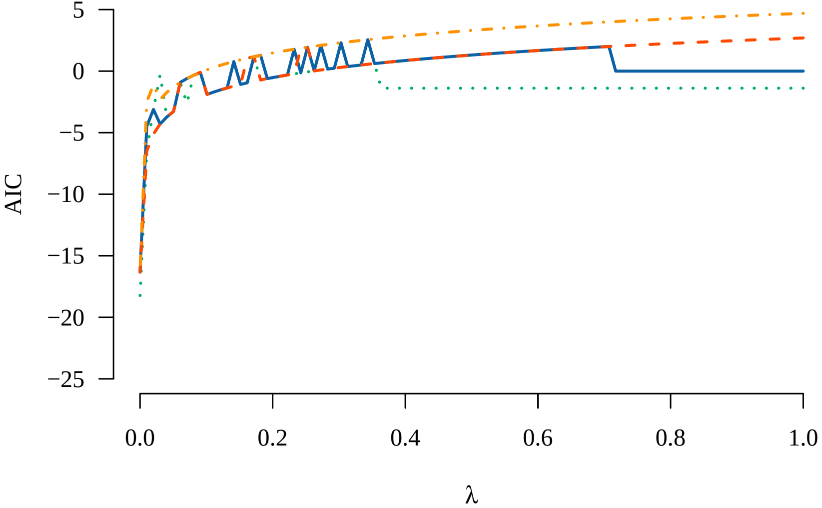
Choose λ to minimize $\text{info}(C_n,\ g)$

Problem!

$$\text{info}(C_n,\ g) = \log\left(\widehat{\text{train}}\right) + C_n g(\text{df})$$

- If $p \gg n$, then as $\lambda \to 0$, $\log\left(\widehat{\text{train}}\right) \to -\infty$
- Unless $C_n g(\text{df}) \to \infty$ increases at a faster rate, we'll always select $\lambda = 0$.

Really dumb example



Don’t use the usual AIC, do something else

Stein's unbiased risk estimation

Under our model, the prediction risk of β can be written

$$\begin{aligned} \frac{1}{n}\mathbb{E}\,||\mathbb{X}\beta - \mathbb{X}\beta_*||_2^2 &= \frac{1}{n}\mathbb{E}\,||\mathbb{X}\beta - Y||_2^2 - \sigma^2 + \frac{2}{n}\sum_{i=1}^{n}\mathrm{Cov}(\widehat{Y}_i, Y_i) \\ &= \frac{1}{n}\mathbb{E}\,||\mathbb{X}\beta - Y||_2^2 - \sigma^2 + \frac{2}{n}\sigma^2\mathrm{df} \end{aligned}$$

Estimate the risk with

$$\frac{1}{n}\,||\mathbb{X}\beta - Y||_2^2 - \widehat{\sigma}^2 + C_n\widehat{\sigma}^2\widehat{\mathrm{df}},$$

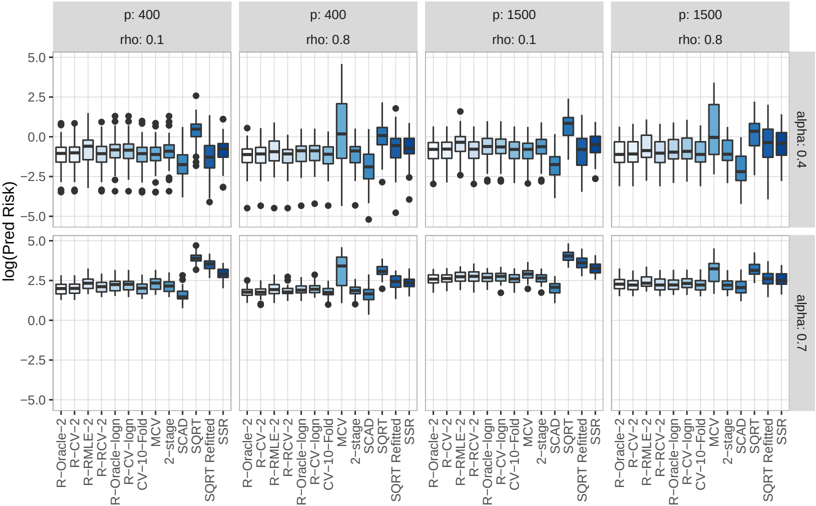
where $\widehat{\sigma}^2$ is an estimator of σ^2, C_n is a constant that is allowed to depend on n, and $\widehat{\mathrm{df}}$ is an estimator of the degress of freedom.

Appropriate estimators

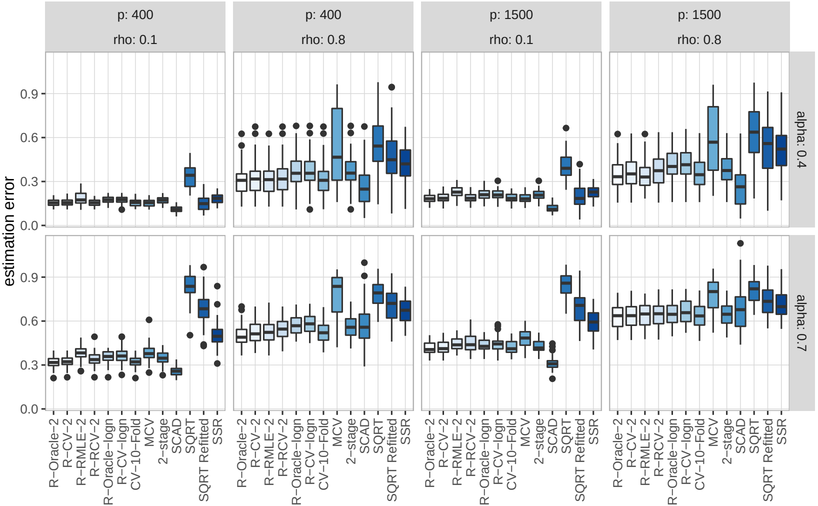
- For lasso, use
$$\widehat{\mathrm{df}} := \#\{\widehat{\beta} \neq 0\}.$$
- For σ^2, we can't use $\widehat{\text{train}}$.
- We tried 3 different high dimensional variance estimators (see Reid, Tibshirani, Friedman 2016):
  1. Choose $\widehat{\lambda}$ by cross validation. Produce,
  $$\widehat{\sigma}^2_{CV} := \frac{1}{n - \widehat{\mathrm{df}}} \left|\left| Y - \mathbb{X}\widehat{\beta} \right|\right|_2^2 .$$
  2. Choose $\widehat{\lambda}$ by cross validation. Produce,
  $$\widehat{\sigma}^2_{RMLE} := \frac{1}{n - \widehat{\mathrm{df}}} \left|\left| H^{\perp} Y \right|\right|_2^2 .$$
  3. Split the data in half, do (1) on each half and average. This is $\widehat{\sigma}^2_{RCV}$ (refitted cross validation, see Fan, Guo, Hao 2012).

Did some simulations

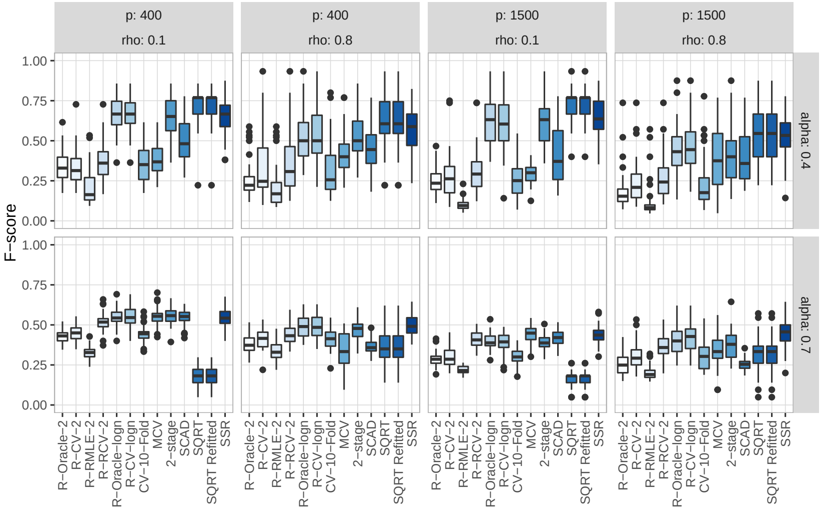
Predicting new responses



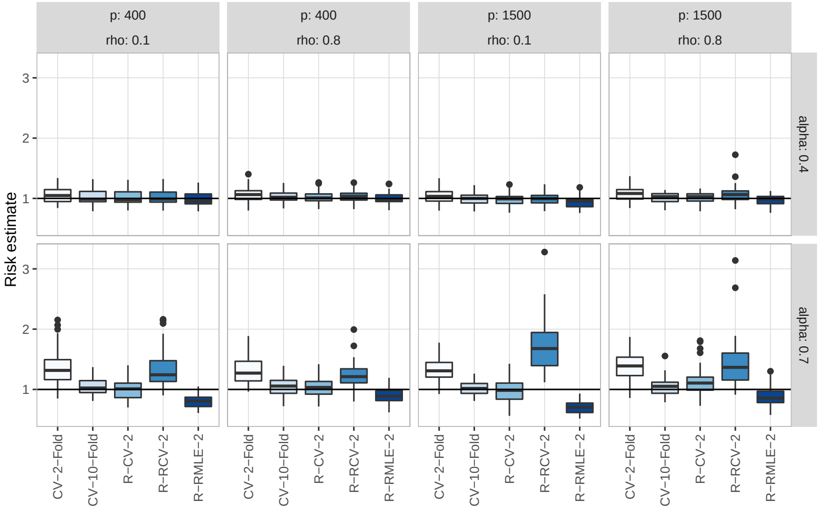
ESTIMATING β_*



FINDING THE RIGHT SUPPORT OF β_*



Estimating the prediction risk



Conclusions

- Don't use regular AIC/BIC in high dimensions
- You need a high-dimensional variance estimator
- Generally (across many simulations not shown) $\widehat{\sigma}^2_{CV}$ works well
- Can still do AIC/BIC like things
- Thanks to NSF and INET for support.