$$\cdot\ AIC_m = -2\,\underbrace{\hat{l}_{n,m}}_{\text{maximum log-likelihood of } M_m \text{ given } n \text{ observations}} + 2\overset{\text{dimension of Model } M_m}{d_m}$$

Autoregressive order selection.

$$AIC_K = 2\log \underbrace{\hat{e}_K}_{\text{Average in-sample prediction error}} + 2K$$

Others:

- Takeuchi's Information Criterion
- AIC based on quasi-likelihood
- Replace "2" with a different positive number

$$\cdot\ BIC_m = -2\,\hat{l}_{n,m} + 2d_m \log n$$

- Bayesian Posterior Probability

$$P(M_m) > 0 \text{ each } m, \quad \sum_{m \in M} P(M_m) = 1$$

likelihood of data

$$p(\mathbf{z} \mid M_m) = \int_{H_m} p_m(\mathbf{z} \mid \theta_m)\, p_m(\theta_m)\, d\theta_m \qquad (1)$$

parameter space for M_m prior density for θ_m

$$p(M_m \mid \mathbf{z}) = \frac{p(\mathbf{z} \mid M_m)\, p(M_m)}{\sum_{m' \in M} p(\mathbf{z} \mid M_{m'})\, p(M_{m'})} \qquad (2)$$

Select the model with the largest posterior probability.

$$B_{m,m'} = \frac{P(\mathbf{z} \mid M_m)}{P(\mathbf{z} \mid M_{m'})} \Big/ \frac{p(M_{m'})}{p(M_m)} \qquad \text{select } M_m \text{ if } B_{m,m'} > 1$$

Bayes Factor

Use sequential MC or MCMC to compute (1) and (2).

Methods that do not require parametric assumptions

Cross Validation

For cross validation with n_t training data and n_v validation data

$\frac{n_v}{n_t} \to 0$ asymptotically AIC ; $\frac{n_v}{n_t} \to \infty$, asymptotically BIC

Bridge Criterion (BC):

$$BC_m = -2\hat{L}_{n,m} + C_n\left(1 + \frac{1}{2} + \cdots + \frac{1}{d_m}\right)$$

(suggested C_n: $n^{\frac{2}{3}}$)

Consistent in a parametric framework and efficient in both frameworks

- Penalized Regression for variable selection

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \|Y - X\beta\|^2 + \sum_{j=1}^{d_n} p(|\beta_j| ; \lambda, \gamma) \right\}$$

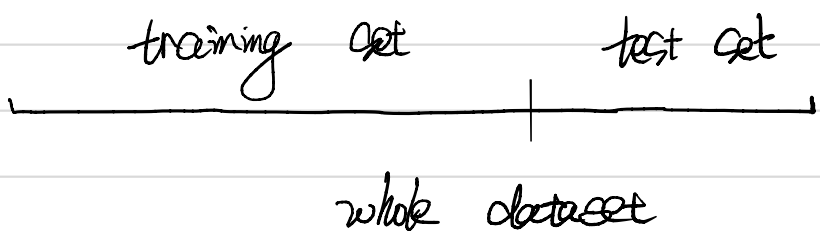
Lasso: $\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \|Y - X\beta\|^2 + \lambda \|\beta\|_1 \right\}$

λ: tuning parameter

Step 1: Using the whole data, generate a data-driven penalty parameter $\{\lambda_1, \lambda_2, \cdots, \lambda_R\}$

Step 2: fit $\hat{\beta}_r = \hat{\beta}(\lambda_r)$ Obtain R models $M_1, M_2, \cdots, M_R$

Step 3: Use leave-one-out cross validation to select model from $\{M_1, M_2, \dots, M_R\}$



Step 1: devide the whole dataset into 2 parts "training set" & "test set";

Step 2:
On training set we apply leave-one-out cross validation to select a "best model";

Step 3: Test the performance of "best model" on "test set", i.e. calculate $\sum_{\{i:\, Y_i \in \text{test set}\}} (Y_{i\alpha} - X_{i\alpha}^T \hat{\beta}_\alpha)^2$

$\hat{\beta}_\alpha$ represents the best model that is fitted & selected in the "training set";

Step 4: Repeat Step 1 – 3 b times with different splits